Improving the Consistency of Nonlinear Estimators: Analysis, Algorithms, and Applications

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Guoquan Huang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

STERGIOS ROUMELIOTIS (advisor)

February, 2013

Acknowledgements

This thesis would never have been possible without the support from a number of people, to whom I am deeply grateful. Foremost is my advisor, Professor Stergios Roumeliotis, for supporting me throughout the years. I would like to thank him for his constant encouragement and guidance as well as patience, for the long hours of passing along his knowledge and experience, for pushing me beyond my own limitations, and for his seemingly endless supply of interesting research problems. I am also thankful for the time and invaluable advice from my committee members, Professor Georgios Giannakis, Professor Tryphon Georgiou, Professor Yousef Saad, and Professor Volkan Isler. At this point, I would like to thank my former supervisors at the Hong Kong Polytechnic University, Professor Ahmad Rad (now at the Simon Fraser University) and Professor Yiu-Kwong Wong, for introducing me into robotics.

I would like to thank all my amazing friends at the MARS lab, Tassos, Nikolas, Sam, Faraz, Ke, Joel, Esha, Igor, Dimitrios, Gian-Luca, Paolo, Chao, and Kejian, with all of whom I have shared hours of discussion, work, laughter, beer, food, and play. Thanks to all my Minnesota friends, Qiang, John, Jun, Zhifeng, Hao, Hanhuai, Paula, Bob, Jim, and many others, for making my stay at Minnesota so memorable. Thanks also go to my best friends back home, Guohua, Zhihua, Yuehao, and Guojing, for their help for everything, literally.

Most importantly, none of this would have been possible without the support and love of my family. Even though my studies sent me far away from home, my parents and brother as well as my parents-in-law have always done everything so that I could follow my dreams, and have never failed to believe in me. Finally, I am so fortunate to have my wife, Ning, who has shared with me all the ups and downs of life in graduate school, and still loves me. In particular, I am so grateful for her tireless monthly commutes between Nashville and Minneapolis in the past two years; she absolutely has provided support beyond words in helping me achieve this.

Abstract

Autonomous robots are emerging as candidates for performing increasingly complex tasks, such as surveillance and environment monitoring, search and rescue, and planetary exploration. Nonlinear estimation (i.e., estimating the state of a nonlinear system from noisy measurements) arises in all these applications. For instance, robot localization – which is considered as one of the fundamental problems in robotics – seeks to determine the robot's pose (position and orientation) using measurements from onboard sensors (e.g., an odometer and a camera). Another closely-related and important example is target tracking, where the objective is to estimate the target's state using remote sensor observations. Even though many different algorithms, such as the extended Kalman filter (EKF) and the batch maximum a posteriori (MAP) estimator, have been developed for solving these problems, substantial empirical evidence shows that most existing nonlinear estimators tend to become *inconsistent* (i.e., the state estimates are biased and the error covariance estimates are smaller than the true ones). Moreover, a significant limitation is that the causes of inconsistency have not been sufficiently studied in the literature; if an estimator is inconsistent, the accuracy of its estimates is unknown, which makes the estimator unreliable. The objective of this dissertation is to investigate the main causes of inconsistency of nonlinear estimation and develop new algorithms for improving consistency.

As one of the main research thrusts, we study in depth the inconsistency problem in robot localization, including simultaneous localization and mapping (SLAM) and multi-robot cooperative localization (CL). In particular, we show *for the first time ever* that one fundamental cause of inconsistency is the mismatch between the observability properties of the underlying nonlinear system and the linearized system used by the estimator. By performing observability analysis, we prove that the linearized error-state system used by standard filtering/smoothing algorithms – the EKF, the unscented Kalman filter (UKF), and the sliding-window filter (SWF) – has an observable subspace of *higher* dimension than that of the underlying nonlinear system. This implies that these estimators gain *spurious* information (more specifically, about the global orientation) from the measurements, which unjustifiably reduces the uncertainty of the state estimates and causes inconsistency. Based on this key insight, for unobservable nonlinear systems, we propose a novel methodology for designing consistent linearized estimators. Specifically, we develop a family of Observability-Constrained (OC)-estimators – including the

OC-EKF, the OC-UKF, and the OC-SWF – whose Jacobians are computed in a way to ensure that the estimator's linearized system model has an observable subspace of the *same* dimension as that of the underlying nonlinear system.

Furthermore, we investigate the inconsistency of estimators for observable nonlinear systems, such as target tracking using distance or bearing measurements, whose cost functions are non-convex and often have *multiple* local minima. In such cases, we discover that the inconsistency of a standard linearized estimator, such as the EKF, is primarily due to the fact that the estimator is able to find and track only *one* local minimum. To address this issue, we convert the estimator's nonlinear cost function into polynomial form and employ algebraic geometry techniques to *analytically* compute all its local minima. These local minima are used as initial estimates by a bank of MAP estimators to efficiently track the most probable hypotheses for the entire state trajectory. Moreover, we adapt this idea to particle filters (PFs) and develop an Analytically-Guided-Sampling (AGS)-PF. Specifically, the AGS-PF employs an *analytically-determined* Gaussian mixture as proposal distribution which not only takes into account the most recent measurement but also matches all the modes of the posterior (optimal proposal) distribution. As a result, the AGS-PF samples the most probable regions of the state space and hence significantly reduces the number of particles required.

As precise long-term localization and tracking are essential for a variety of robotic applications, by introducing a solid theoretical framework for improving the consistency of nonlinear estimators, this work offers significant benefits for robots employed in these tasks. Moreover, the proposed solutions constitute novel paradigms for engineers to follow when designing consistent estimators for other nonlinear systems, and hence have the potential to benefit applications beyond robotics.

Contents

List of Tables

List of Figures

Nomenclature and Abbreviations

$\mathcal{N}(\boldsymbol{\mu}, \mathbf{P}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{P}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{P}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$ multivariate Gaussian pdf with mean $\boldsymbol{\mu}_{n\times 1}$ and covariance matrix $\mathbf{P}_{n\times n}$

$\mathbf{C}(\phi)$ 2×2 rotation matrix with rotation angle ϕ

${}^W\mathbf{r}$ vector $\mathbf{r}$ expressed with respect to frame $\{W\}$

${}^W\mathbf{p}_Z$ origin of frame $\{Z\}$ expressed with respect to frame $\{W\}$

BA Bundle Adjustment

CL Cooperative localization

d.o.f. degrees of freedom

EKF Extended Kalman Filter

FLS Fixed-Lag Smoother

GPS Global Positioning System

GSF Gaussian Sum Filter

GSPF Gaussian Sum Particle Filter

IEKF Iterated Extended Kalman Filter

IMU Inertial Measurement Unit

iSAM incremental Smoothing and Mapping

LRKF Linear Regression Kalman Filter

MAP Maximum A Posteriori

MHEKF Multi-Hypothesis Extended Kalman Filter

MLE Maximum Likelihood Estimator

NEES Normalized Estimation Error Squared

OC Observability Constrained

pdf probability density function

PF Particle Filter

RMSE Root Mean Squared Error

SAM Smoothing and Mapping

SLAM Simultaneous Localization and Mapping

SRUKF Square-Root Unscented Kalman Filter

SWF Sliding Window Filter

UKF Unscented Kalman Filter

UPF Unscented Particle Filter

V-INS Vision-aided Inertial Navigation System

Chapter 1

Introduction

1.1 Nonlinear estimation in robotics

Autonomous mobile robots have the potential to assist people (e.g., during search and rescue operations [25]) and to augment human capabilities (e.g., working in dangerous or inaccessible environments [46]). To achieve this, however, a fundamental estimation problem that must be first solved is *localization* – that is, determining the position and orientation (pose) of a robot using measurements from its onboard sensors. Another important nonlinear estimation problem arising in robotics is *target tracking*, in which one or more, possibly mobile, sensors observe and track a target in order to estimate its position, velocity, etc. Localization and target tracking appear in many practical applications, such as transportation [19, 156], construction [128, 141], planetary exploration [46], guidance for the visually impaired [61, 164], surveillance and environment monitoring [24, 129], as well as search and rescue operations in disaster zones [25]. In this work, we primarily focus on these two key robotic problems and use them to illustrate our methodologies of improving the consistency of nonlinear estimation algorithms. In what follows, we first provide a brief overview of robot localization and target tracking in order to better understand the ensuing discussion on estimator consistency.

1.1.1 Robot localization

The objective of robot localization is to fuse proprioceptive (e.g., from an odometer) and exteroceptive (e.g., from a laser) sensor measurements in order to compute an estimate of the robot's

pose. There exist many variants of this problem, depending on the number of robots involved and the prior knowledge they may have about the environments within which they operate. In this thesis, we particularly focus on the following two important cases:

- Simultaneous Localization and Mapping (SLAM): When exploring an unknown environment, a robot seeks to estimate both its own pose and the positions of the landmarks it observes (e.g., corner points detected from images, or line segments extracted from the laser scanner data) [11, 43, 121, 145, 157]. This process offers several benefits: By observing static landmarks over multiple time instants, or when the robot revisits an area (i.e., loop closing), the estimation errors over long time periods remain bounded [116]. Thus, a SLAM solution permits accurate, long-term localization in unknown environments, and is considered an enabling technology for robot autonomy [155].
- Cooperative Localization (CL): A team of robots can localize by sharing robot-to-robot measurements and jointly estimating their poses [102]. A CL solution provides the means for implicit sensor sharing, as localization information is dissipated over a (wireless) network to all the members of the group, and results in considerable gains in terms of localization accuracy for all robots [117]. Moreover, when at least one of the robots is capable of obtaining measurements to static landmarks, then the poses of all robots and the positions of all landmarks can be simultaneously estimated, through multi-robot cooperative SLAM (C-SLAM) [118]. C-SLAM enjoys the advantages of both SLAM and CL and attains bounded localization errors for all robots within the team [118].

1.1.2 Target tracking

Target tracking is the problem of estimating the kinematic state of a moving target using range and/or bearing measurements provided by a (mobile) sensor whose pose is often assumed to be known. This is a classical nonlinear estimation problem that has attracted significant interest over the past decades [14, 17]. Examples of recent research on target tracking include designing new estimation algorithms and adaptively controlling the sensor's motion [7, 31, 99, 170–172]. Depending on the type of measurements used, it includes range-only tracking [28, 135] and bearing-only tracking [47, 122], which are the two cases also studied in this work.

1.2 Consistency of nonlinear estimation

Nonlinear estimation problems, such as mobile robot localization and target tracking, are challenging for a number of reasons. In contrast to the linear-model Gaussian-noise case, a nonlinear estimator is generally intractable without imposing simplifying assumptions. Specifically, it is not feasible to propagate and update either an entire probability density function (pdf) for the state conditioned on the available measurements or an infinite number of parameters describing that density. One particular difficulty arising in the design of nonlinear estimators is the *consistency* issue, since no provably consistent estimator can be constructed for a nonlinear system. As defined in [14], a state estimator is *consistent* if the estimation errors are zero-mean and have covariance matrix smaller than or equal to the one calculated by the estimator. Consistency is one of the primary criteria for evaluating the performance of any estimator; if an estimator is inconsistent, then the accuracy of the produced state estimates is unknown, which in turn makes the estimator unreliable.

Despite these challenges, the problems of robot localization and target tracking have been studied for decades [17, 155], and various estimators have been employed for solving them, such as the extended Kalman filter (EKF) [14, 43], the maximum likelihood estimator (MLE) [64], the maximum a posteriori (MAP) estimator [124], and the particle filter (PF) [8, 48]. Among these algorithms, the EKF remains a popular choice primarily due to its relatively low processing requirements and its ease of implementation. However, its performance depends on the magnitude of the linearization errors. To reduce the linearization errors, the iterated EKF (IEKF) [82] is often used, which iteratively relinearizes the nonlinear measurement model till convergence. Alternatively, the unscented Kalman filter (UKF) [84] deterministically samples the nonlinear model around the current state estimate and employs linear regression to improve the accuracy of the linear approximation. Nevertheless, the EKF, as well as any (explicit or implicit) linearization-based filtering approach, marginalizes all but the current state and hence is unable to correct linearization errors involving previous states. For this reason, smoothing algorithms, either in batch or incremental fashion, have become popular, especially for SLAM [36, 38, 53–55, 67, 87, 88, 95, 100, 142, 150]. In particular, a sliding-window filter (SWF) [142] (or fixed-lag smoother (FLS) [38, 111, 132]) estimates the states over a sliding time window, by concurrently processing all the measurements involving these states, hence

reducing the effect of the linearization errors. Moreover, a batch-MAP estimator [91] computes the estimates for the states at all time steps using all available measurements. This allows continuous relinearization around all the states, which can greatly reduce the linearization errors. However, most filters and smoothers typically can track only one, of the potentially many, modes of the posterior pdf, which can degrade the performance. Only a few estimators, such as the multi-hypothesis EKF (MHEKF) [99] (i.e., the Gaussian sum filter (GSF) [3]) and the PF [8, 41, 58], are specifically designed to handle multimodal distributions by simultaneously tracking a set of different hypotheses for the state estimates. Most of the time, however, these hypotheses are generated randomly, thus wasting a considerable portion of the computational resources.

Due to the aforementioned reasons (i.e., large linearization errors and multiple local minima), most estimators (either filters or smoothers) tend to become *inconsistent* when applied to robot localization and target tracking. The lack of understanding of the fundamental causes of estimator inconsistency in these applications clearly is a significant limitation, which we seek to address in this dissertation. In particular, we prove that the observability of the system model based on which an estimator is built, profoundly affects the estimator's performance, and plays a significant role in determining consistency. Furthermore, we show that the inability of an estimator to track all modes of the multimodal posterior pdf can cause inconsistency or even divergence. Once the root causes of estimator inconsistency are identified, we design new estimation algorithms that explicitly address these causes and improve consistency. In what follows, we discuss the research objectives of this dissertation in more detail.

1.3 Research objectives

The primary objectives of this research effort are to determine the fundamental causes of estimator inconsistency and design new estimation algorithms that improve consistency. While the proposed analysis and algorithms can be generalized to a broad class of nonlinear systems, we hereafter focus on the problems of mobile robot localization and target tracking. Specifically, the goals of this research are the following:

1.3.1 Investigate the fundamental causes of estimator inconsistency

We first study the problem of estimation inconsistency from the perspective of system observability. Observability examines the feasibility of estimating the state given all available measurements. A dynamical system is observable if its state at a certain time instant can be uniquely determined based on a finite sequence of its outputs (measurements) [18]. By performing observability analysis, we are able to determine the observable directions in the state space along which an estimator should acquire information from the available measurements. We conjecture that *if the observable directions are erroneous, the estimator may become inconsistent*. For instance, for the robot localization systems [i.e., SLAM (see Section 2.4) and CL (see Section 3.4)], we analytically prove that the standard EKF employs a linearized error-state system model that has an unobservable subspace of *lower* dimension than that of the underlying nonlinear system. As a result, the estimator gains spurious information from the measurements and hence incorrectly reduces the estimated covariance. This is shown to be the main cause of inconsistency in this case.

Although system observability has been identified as an important cause of estimator inconsistency, there certainly exist other issues affecting the consistency of nonlinear estimators. For example, the observability properties of systems such as range-only and bearing-only target tracking, have less impact since both systems are observable [146, 147], and the corresponding linearized system models are also observable. However, as it will be shown, most estimators, such as the EKF, can still become inconsistent. In this case, we discover that the fundamental cause of inconsistency is *the inability of the estimator to track a multimodal posterior pdf* (see Section 7.4). In particular, if the estimator erroneously tracks a mode different from the global optimum, its estimates may become inconsistent.

The PF can ideally track multiple modes of the pdf and does not require linearization. However, it can also become inconsistent when applied to robot localization and target tracking [13, 165]. We conjecture that this is primarily due to *particle depletion* [8], where the very few surviving particles (i.e., particles with significant weights) cannot sufficiently represent the underlying posterior pdf (see Chapter 8).

1.3.2 Improve the consistency of nonlinear estimators

Our objective is to design new nonlinear estimation algorithms that improve consistency, by explicitly addressing the primary causes of inconsistency. Firstly, we propose a novel observability-based methodology for improving estimator consistency in robot localization including SLAM and CL. The key idea behind our approach is to compute the filter Jacobians in such a way that the resulting linearized system model used by the estimator has an unobservable subspace of appropriate dimensions. Within this framework, we develop a family of Observability-Constrained (OC)-estimators, including both filters and smoothers [i.e., the OC-EKFs (see Chapters 2 and 3), the OC-UKF (see Chapter 4), and the OC-SWF (see Chapter 5)]. Moreover, we study filter consistency for a certain class of *observable* nonlinear systems, and employ the same observability-based methodology to develop new estimation algorithms for improving their consistency (see Chapter 6).

Secondly, we introduce a general framework for finding and tracking the modes of multi-modal posterior pdfs for a broad class of nonlinear estimation problems in robotics and computer vision that can be expressed in (or converted into) polynomial form. The key idea of this approach is to convert the estimator's nonlinear cost function into polynomial form, and then employ algebraic-geometry techniques [33] to analytically compute all the stationary points, and thus the local minima, which correspond to the multiple modes of the posterior pdf. Furthermore, we employ a bank of MAP estimators, which allow relinearization of the entire state history as well as multi-hypothesis tracking, and introduce an efficient hypothesis generation scheme (see Chapter 7). Moreover, we adapt this idea of analytically selecting hypotheses to PFs, and develop an Analytically-Guided-Sampling (AGS)-PF (see Chapter 8). Specifically, the AGS-PF employs an *analytically-determined* Gaussian mixture as proposal distribution which not only takes into account the most recent measurement but also matches all the modes of the posterior (optimal proposal) distribution. As a result, the AGS-PF samples along the most probable regions of the state space and hence dramatically reduces the number of particles required.

Persistent long-term localization and tracking are essential for various robotic applications, ranging from planetary and underwater exploration to service robots for businesses and homes. By introducing a solid theoretical framework for designing consistency-improved nonlinear estimators, this dissertation will offer significant benefits to robots employed in a wide range of tasks such as surveillance. Moreover, the proposed solutions provide novel paradigms for other

engineers to follow when designing consistent estimators for nonlinear systems. Thus, the proposed methodologies have the potential to be extended to an even broader class of nonlinear estimation problems and hence to benefit the corresponding applications.

1.4 Organization of the manuscript

In the following chapters, the estimator inconsistency in robot localization is first addressed from the perspective of system observability. In particular, Chapters 2 and 3 focus on the EKF inconsistency in SLAM and CL, respectively, and the observability-based methodology for improving EKF consistency is presented. The UKF-SLAM inconsistency is studied in Chapter 4, where the UKF computational complexity is also reduced. In Chapter 5, the observability-based methodology is extended to the smoothing framework and the OC-SWF for SLAM is introduced. Furthermore, in Chapter 6, this methodology is generalized to a special class of observable nonlinear systems, i.e., discrete-time nonlinear systems with partial-state measurements. In Chapters 7 and 8, the estimator inconsistency is studied from a different perspective for the case of posterior pdfs with multiple modes. In particular, a bank of MAP estimators and the AGS-PF are developed for tracking the most probable hypotheses of the target's state. Finally, Chapter 9 provides concluding remarks and an outlook on future research directions.

Chapter 2

Observability-Constrained EKFs for SLAM

In this chapter, we study the EKF consistency of SLAM from the perspective of system observability. We analytically show that the linearized system employed by the EKF has different observability properties than the underlying nonlinear SLAM system, which is one fundamental cause of inconsistency. To address this problem, we develop an observability-based methodology to ensure that the EKF linearized system has the same number of unobservable directions as the nonlinear SLAM system. Parts of this chapter have been published in [66, 70, 72].

2.1 Introduction

Simultaneous localization and mapping (SLAM) is the process of building a map of an environment and concurrently generating an estimate of the robot's pose (position and orientation) using sensor readings. For autonomous vehicles exploring unknown environments, the ability to perform SLAM is essential. Since [144] first introduced a stochastic-mapping solution to the SLAM problem, rapid and exciting progress has been made, resulting in several competing solutions. Recent interest in SLAM has focused on the design of estimation algorithms (e.g., [115, 130]), data association techniques [123], and feature extraction [140]. Among the numerous algorithms developed thus far for SLAM, the EKF remains one of the most popular approaches, and has been used in several applications (e.g., [92, 125, 169]). However, in spite of its widespread adoption, the fundamental issue of *consistency* of the EKF-SLAM algorithm

has not yet been sufficiently investigated.

As discussed in the preceding chapter (see Section 1.2), a state estimator is *consistent* if the estimation errors are zero-mean and have covariance smaller than or equal to the one calculated by the estimator [14]. Consistency is one of the primary criteria for evaluating the performance of any estimator. If an estimator is inconsistent, then the accuracy of the produced state estimates is unknown, which in turn makes the estimator unreliable. Since SLAM is a nonlinear estimation problem, no provably consistent estimator can be constructed for it. The consistency of every estimator has to be evaluated experimentally. In particular for the *standard* EKF-SLAM algorithm, there exists significant empirical evidence showing that the computed state estimates tend to be *inconsistent* (see Section 2.2). Clearly, the lack of understanding the causes of the filter inconsistency is a significant limitation, which negatively affects long-term autonomous navigation.

In this chapter, we investigate in depth one fundamental cause of the inconsistency of the standard EKF-SLAM algorithm. In particular, we revisit this problem from a new perspective, i.e., by analyzing the observability properties of the filter's system model. Our key conjecture in this work is that the observability properties of the EKF linearized system model profoundly affect the performance of the filter, and are a significant factor in determining its consistency. Specifically, the major contributions of this work are the following:

- Through an observability analysis, we prove that the standard EKF-SLAM employs a linearized error-state system model that has an unobservable subspace of dimension two, even though the underlying nonlinear system model has three unobservable degrees of freedom (d.o.f.), corresponding to the position and orientation of the global reference frame. As a result, the filter gains spurious information along directions of the state space where no information is actually available. This leads to an unjustified reduction of the estimated covariance, and is a primary cause of filter inconsistency.

- Motivated by the observability analysis, we propose a new observability-based methodology for improving the EKF consistency. The key idea of this approach is to compute the EKF Jacobians in such a way that ensures that the unobservable subspace of the EKF system model is of correct dimensions. This can be achieved in three different ways, resulting in three different Observability-Constrained (OC)-EKF algorithms: i) OC-EKF1

computes the Jacobians using the *first-ever* available estimates for each of the state variables and hence is also called First-Estimates-Jacobian (FEJ)-EKF. ii) OC-EKF2 selects the linearization points that not only guarantee the desired observability properties but also minimize the expected linearization errors (i.e., the difference between the linearization point and the true state). This is formulated as a constrained minimization problem, which we solve to determine the linearization points used for computing the filter Jacobians. iii) OC-EKF3 *directly* computes the *measurement* Jacobian at each time step in such a way that ensures the information acquired from the available measurements is only along the observable directions in the state space. This is achieved by projecting the best-available measurement Jacobian (which is computed using the latest state estimates as in the standard EKF) onto the observable directions. This is in contrast to the OC-EKFs 1 and 2 which compute the Jacobians *indirectly* by first finding appropriate linearization points that ensure correct observability properties for the corresponding linearized systems.

- Through extensive Monte-Carlo simulations and real-world experiments, we verify that the proposed OC-EKFs substantially outperform the standard EKF, even though they use less accurate filter Jacobians (since the Jacobians of the OC-EKFs, in general, are different from those computed using the latest, and thus best, state estimates). This result supports our conjecture that the observability properties of the EKF system model play a fundamental role in determining consistency.

2.2 Related work

The EKF is one of the most widely used algorithms for SLAM. However, its inconsistency issue has only recently begun to attract research interest [12, 26, 27, 66, 70, 72, 78, 79, 85].

Specifically, the work of [85] first reported the issue of EKF inconsistency by observing that when a stationary robot measures the relative position of a landmark multiple times, the estimated variance of the robot's orientation error becomes smaller. Since the observation of a previously unseen feature does not provide any information about the robot state, this reduction is incorrect and leads to inconsistency. In addition, a condition was described that the filter Jacobians need to satisfy in order to permit consistent estimation. Recently, the work of [79] extended the analysis of [85] to the case in which a robot observes a landmark from two positions

(i.e., the robot observes a landmark, moves and then re-observes the landmark). A constraint was provided that the filter Jacobians need to fulfill in this case so as to allow for consistent estimation. It was also shown that this condition is generally violated, due to the fact that the filter Jacobians at different time instants are evaluated using different estimates for the same state variables. Interestingly, we will show that these conditions, i.e., for a stationary robot [85] and a one-step motion [79], are special cases of an observability-based condition derived for the general case of a moving robot (see Section 2.5.4).

The authors of [12] examined several symptoms of the inconsistency of the standard EKF-SLAM algorithm, and based on Monte-Carlo simulations, argued that the uncertainty in the robot orientation is the main cause of the EKF inconsistency. However, no theoretical results were provided. The work of [78] further confirmed the empirical findings in [12], and argued by example that the inconsistency of the standard EKF-SLAM is always in the form of over-confident estimates (i.e., the computed covariance is smaller than the actual one).

The first attempt to improve filter consistency was reported in [26, 27] where the robocentric mapping algorithm was proposed by expressing the landmarks in a robot-relative (instead of world-centric) frame of reference. In this formulation, during each propagation step, *all* landmark position estimates need to be recalculated, since they are expressed with respect to the moving robot frame. As a result, during propagation, all landmark estimates and their covariances are affected by the linearization errors of the process model, which degrades performance. Note that this issue does not exist in the world-centric formulation of SLAM. Moreover, in comparison to our proposed OC-EKFs, the computational cost of the robocentric mapping filter is significantly higher. Specifically, the OC-EKFs have computational cost identical to the standard world-centric EKF-SLAM algorithm: *linear*, in the number of landmarks, during propagation, and *quadratic* during update. In contrast, in the robocentric mapping filter, both the propagation and update steps have computational cost *quadratic* in the number of landmarks.

Most previous work has only empirically examined several symptoms of the SLAM inconsistency, except for a few special cases (e.g., a stationary robot [85], and one-step motion [79]) where analytical studies were performed. However, no theoretical analysis of the cause of filter inconsistency was conducted. In this thesis, we study this problem from the system observability perspective, and identify as a fundamental cause of inconsistency the mismatch in the dimensions of the observable subspaces between the standard EKF linearized system and

the underlying nonlinear system. Relying on this key finding, we propose an observability-constrained framework for improving filter consistency, where the filter Jacobians are computed so as to ensure that the observable subspace of the EKF linearized system has the correct dimensions. Specifically, the OC-EKF1 (i.e., FEJ-EKF) [66, 70] selects the first-ever state estimates as the linearization points used in computing the filter Jacobians. The OC-EKF2 [72] finds the optimal linearization points that not only ensure that the observable subspace of the EKF linearized system model has correct dimensions, but also minimize the linearization errors. The OC-EKF3, instead of first finding appropriate linearization points as is the case for the OC-EKFs 1 and 2, directly computes the necessary measurement Jacobian, by projecting the most accurate measurement Jacobian onto the observable directions of the system model.

2.3 Standard EKF-SLAM formulation

In this section, we present the equations of the standard EKF-SLAM formulation with *generalized* system and measurement models. To preserve the clarity of presentation, we first focus on the case where a *single* landmark is included in the state vector, while the case of multiple landmarks is addressed later on. In the standard formulation of SLAM, the state vector comprises the robot pose and the landmark position in the global frame of reference. Thus, at time-step k the state vector is given by:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{p}_{R_k}^T & \phi_{R_k} & \mathbf{p}_L^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{x}_{R_k}^T & \mathbf{p}_L^T \end{bmatrix}^T \tag{2.1}$$

where $\mathbf{x}_{R_k} = [\mathbf{p}_{R_k}^T \quad \phi_{R_k}]^T$ denotes the robot pose (position and orientation), and $\mathbf{p}_L$ is the landmark position. EKF-SLAM recursively evolves in two steps: propagation and update, based on the discrete-time process and measurement models, respectively.

2.3.1 EKF propagation

In the propagation step, the robot's odometry measurements are processed to obtain an estimate of the pose change between two consecutive time steps, and then employed in the EKF to propagate the robot state estimate. On the other hand, since the landmark is static, its state estimate does not change with the incorporation of a new odometry measurement. The EKF

propagation equations are given by:[1]

$$\hat{\mathbf{p}}_{R_{k+1|k}} = \hat{\mathbf{p}}_{R_{k|k}} + \mathbf{C}(\hat{\phi}_{R_{k|k}})^{R_k}\hat{\mathbf{p}}_{R_{k+1}} \tag{2.2}$$

$$\hat{\phi}_{R_{k+1|k}} = \hat{\phi}_{R_{k|k}} + {}^{R_k}\hat{\phi}_{R_{k+1}} \tag{2.3}$$

$$\hat{\mathbf{p}}_{L_{k+1|k}} = \hat{\mathbf{p}}_{L_{k|k}} \tag{2.4}$$

where $\mathbf{C}(\cdot)$ denotes the 2×2 rotation matrix, and ${}^{R_k}\hat{\mathbf{x}}_{R_{k+1}} = [{}^{R_k}\hat{\mathbf{p}}_{R_{k+1}}^T \quad {}^{R_k}\hat{\phi}_{R_{k+1}}]^T$ is the odometry-based estimate of the robot's motion between time-steps k and $k+1$. This estimate is corrupted by zero-mean, white Gaussian noise $\mathbf{w}_k = {}^{R_k}\mathbf{x}_{R_{k+1}} - {}^{R_k}\hat{\mathbf{x}}_{R_{k+1}}$, with covariance matrix $\mathbf{Q}_k$. This process model is nonlinear, and can be described by the following generic nonlinear function:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, {}^{R_k}\hat{\mathbf{x}}_{R_{k+1}} + \mathbf{w}_k) \tag{2.5}$$

In addition to the state propagation equations, the linearized error-state propagation equation is necessary for the EKF. This is given by:

$$\begin{aligned}\tilde{\mathbf{x}}_{k+1|k} &= \begin{bmatrix}\mathbf{\Phi}_{R_k} & \mathbf{0}_{3\times 2}\\ \mathbf{0}_{2\times 3} & \mathbf{I}_2\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}_{R_{k|k}}\\ \tilde{\mathbf{p}}_{L_{k|k}}\end{bmatrix} + \begin{bmatrix}\mathbf{G}_{R_k}\\ \mathbf{0}_{2\times 2}\end{bmatrix}\mathbf{w}_k \\ &\triangleq \mathbf{\Phi}_k\tilde{\mathbf{x}}_{k|k} + \mathbf{G}_k\mathbf{w}_k\end{aligned} \tag{2.6}$$

where $\mathbf{\Phi}_{R_k}$ and $\mathbf{G}_{R_k}$ are obtained from the state propagation equations (2.2)-(2.3):

$$\mathbf{\Phi}_{R_k} = \begin{bmatrix}\mathbf{I}_2 & \mathbf{J}\mathbf{C}(\hat{\phi}_{R_{k|k}})^{R_k}\hat{\mathbf{p}}_{R_{k+1}}\\ \mathbf{0}_{1\times 2} & 1\end{bmatrix} \tag{2.7}$$

$$\equiv \begin{bmatrix}\mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{R_{k+1|k}} - \hat{\mathbf{p}}_{R_{k|k}}\right)\\ \mathbf{0}_{1\times 2} & 1\end{bmatrix} \tag{2.8}$$

$$\mathbf{G}_{R_k} = \begin{bmatrix}\mathbf{C}(\hat{\phi}_{R_{k|k}}) & \mathbf{0}_{2\times 1}\\ \mathbf{0}_{1\times 2} & 1\end{bmatrix} \tag{2.9}$$

with $\mathbf{J} \triangleq \begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}$.

[1] Throughout this dissertation, the subscript $\ell|j$ refers to the estimate of a quantity at time-step ℓ, after all measurements up to time-step j have been processed. $\hat{x}$ is used to denote the estimate of a random variable x, while $\tilde{x} = x - \hat{x}$ is the error in this estimate. $\mathbf{0}_{m\times n}$ and $\mathbf{1}_{m\times n}$ denote $m \times n$ matrices of zeros and ones, and $\mathbf{I}_n$ is the $n \times n$ identity matrix. Finally, we use the concatenated forms $s\phi$ and $c\phi$ to denote the $\sin\phi$ and $\cos\phi$ functions.

It is important to point out that the form of the propagation equations presented above is general, and holds for any robot kinematic model (e.g., unicycle, bicycle, or Ackerman model). In Appendix A.1, we derive the expressions for (2.2)-(2.4), as well as the state and noise Jacobians, for the common case where the unicycle model is used.

2.3.2 EKF update

During SLAM, the measurement used for updates in the EKF is a function of the relative position of the landmark with respect to the robot:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k = \mathbf{h}\left({}^{R_k}\mathbf{p}_L\right) + \mathbf{v}_k \tag{2.10}$$

where ${}^{R_k}\mathbf{p}_L = \mathbf{C}^T(\phi_{R_k})(\mathbf{p}_L - \mathbf{p}_{R_k})$ is the position of the landmark with respect to the robot at time-step k, and $\mathbf{v}_k$ is zero-mean Gaussian measurement noise with covariance $\mathbf{R}_k$. In this work, we allow $\mathbf{h}$ to be *any* measurement function. For instance, $\mathbf{z}_k$ can be a direct measurement of relative position, a pair of range and bearing measurements, bearing-only measurements from monocular cameras, etc. Generally, the measurement function is nonlinear, and hence it is linearized for use in the EKF. The linearized measurement error equation is given by:

$$\begin{aligned}\tilde{\mathbf{z}}_k &\simeq \begin{bmatrix}\mathbf{H}_{R_k} & \mathbf{H}_{L_k}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}_{R_{k|k-1}} \\ \tilde{\mathbf{p}}_{L_{k|k-1}}\end{bmatrix} + \mathbf{v}_k \\ &\triangleq \mathbf{H}_k\tilde{\mathbf{x}}_{k|k-1} + \mathbf{v}_k\end{aligned} \tag{2.11}$$

where $\mathbf{H}_{R_k}$ and $\mathbf{H}_{L_k}$ are the Jacobians of $\mathbf{h}$ with respect to the robot pose and the landmark position, respectively, evaluated at the state estimate $\hat{\mathbf{x}}_{k|k-1}$. Using the chain rule of differentiation, these are computed as:

$$\mathbf{H}_{R_k} = (\nabla\mathbf{h}_k)\mathbf{C}^T(\hat{\phi}_{R_{k|k-1}})\begin{bmatrix}-\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_{k|k-1}} - \hat{\mathbf{p}}_{R_{k|k-1}})\end{bmatrix} \tag{2.12}$$

$$\mathbf{H}_{L_k} = (\nabla\mathbf{h}_k)\mathbf{C}^T(\hat{\phi}_{R_{k|k-1}}) \tag{2.13}$$

where $\nabla\mathbf{h}_k$ denotes the Jacobian of $\mathbf{h}$ with respect to the robot-relative landmark position (i.e., with respect to the vector ${}^{R_k}\mathbf{p}_L$), evaluated at the state estimate $\hat{\mathbf{x}}_{k|k-1}$.

2.4 SLAM observability analysis

In this section, we perform an observability analysis for the generalized EKF-SLAM formulation derived in the previous section, and compare its properties with those of the underlying

nonlinear system. Based on this analysis, we draw conclusions about the filter consistency.

It should be pointed out that the observability properties of SLAM have been studied in only a few cases in the literature. In particular, the work of [4, 5] investigated the observability of a simple linear time-invariant (LTI) SLAM system, and showed that it is unobservable. The work of [167] approximated the SLAM system by a piecewise constant linear (PWCL) system, applied the technique of [51] to study the observability properties of bearing-only SLAM, and showed that it is also unobservable. On the other hand, in [70, 72, 105] the observability properties of the nonlinear SLAM system were studied using the nonlinear observability rank condition introduced by [60]. It was proved that the nonlinear SLAM system is unobservable, with *three* unobservable d.o.f., corresponding to global translation and rotation of the state vector.

All the aforementioned approaches examine the observability properties of the nonlinear SLAM system, or of linear approximations to it. However, to the best of our knowledge, an analysis of the observability properties of the EKF *linearized error-state* system model had not been carried out prior to our work [66, 70, 72]. Since this model is the one used in any actual EKF implementation, a lack of understanding of its observability properties appears to be a significant limitation. In fact, as shown in this chapter, these properties play a significant role in determining the consistency of the filter, and form the basis of our approaches for improving estimation performance.

2.4.1 Nonlinear SLAM observability analysis

We start by carrying out the observability analysis for the continuous-time nonlinear SLAM system. This analysis is based on the observability rank condition introduced in [60]: "*If a nonlinear system is locally weakly observable, the observability rank condition is satisfied generically*". We show that the SLAM system does not satisfy the observability rank condition, and thus is neither locally weakly observable nor locally observable. Note that we here conduct the analysis for a *general* measurement model, instead of only relative-position or distance-and-bearing measurement as in [70, 105].

We employ a unicycle kinematic model for the robot, while similar conclusions can be

drawn if different models are used [105]. The process model in continuous-time form is:

$$\begin{bmatrix} \dot{x}_R(t) \\ \dot{y}_R(t) \\ \dot{\phi}_R(t) \\ \dot{x}_L(t) \\ \dot{y}_L(t) \end{bmatrix} = \begin{bmatrix} c\phi_R(t) \\ s\phi_R(t) \\ 0 \\ 0 \\ 0 \end{bmatrix} v(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \omega(t)$$

$$\Rightarrow \dot{\mathbf{x}}(t) = \mathbf{f}_1 v(t) + \mathbf{f}_2 \omega(t) \tag{2.14}$$

where $\begin{bmatrix} v & \omega \end{bmatrix}^T =: \mathbf{u}$ is the control input, consisting of the linear and rotational velocities. Since any type of measurement in SLAM is a function of the relative position of the landmark with respect to the robot, we can write the measurement model in the following generic form:

$$z(t) = h(\rho, \psi) \tag{2.15}$$

$$\rho = ||\mathbf{p}_L - \mathbf{p}_R|| \tag{2.16}$$

$$\psi = \text{atan2}(y_L - y_R, x_L - x_R) - \phi_R \tag{2.17}$$

where ρ and ψ are the robot-to-landmark distance and bearing angle, respectively. Note that parameterizing the measurement with respect to ρ and ψ is equivalent to parameterizing it with respect to the landmark position expressed in the robot frame, ${}^R\mathbf{p}_L$. The relation between these quantities is ${}^R\mathbf{p}_L = \rho \begin{bmatrix} c\psi \\ s\psi \end{bmatrix}$. To facilitate the ensuing nonlinear observability analysis, we first prove that:

Lemma 2.4.1. *All the Lie derivatives of the nonlinear SLAM system [see* (2.14) *and* (2.15)*] are functions of ρ and ψ only.*

Proof. See Appendix A.2 □

We will now employ this result for the nonlinear observability analysis. In particular, assume that a number of different measurements are available, $z_i = h_i(\rho, \psi)$, $i = 1, 2, ..., n$. Then, since all the Lie derivatives for all measurements are functions of ρ and ψ only, we can prove that:

Lemma 2.4.2. *The space spanned by all the k-th order Lie derivatives $L^k_{\mathbf{f}_j} h_i$ $(\forall k \in \mathbb{N}, j = 1, 2, i = 1, 2, ..., n)$ is denoted by $\mathcal{G}$, and the space $d\mathcal{G}$ spanned by the gradients of the elements*

of $\mathcal{G}$ is given by:

$$d\mathcal{G} = \underset{\text{row}}{\text{span}} \begin{bmatrix} s\phi_R & -c\phi_R & -c\phi_R\delta x - s\phi_R\delta y & -s\phi_R & c\phi_R \\ c\phi_R & s\phi_R & s\phi_R\delta x - c\phi_R\delta y & -c\phi_R & -s\phi_R \end{bmatrix} \tag{2.18}$$

where $\delta x \triangleq x_L - x_R$ and $\delta y \triangleq y_L - y_R$.

Proof. See Appendix A.3 □

The matrix shown above is the "observability matrix" for the nonlinear SLAM system under consideration. Clearly, this is not a full-rank matrix, and hence the system is unobservable. Intuitively, this is a consequence of the fact that we cannot gain *absolute*, but rather only *relative* state information from the available measurements. Even though the notion of an "unobservable subspace" cannot be strictly defined for this system, the physical interpretation of the basis of $d\mathcal{G}^\perp$ will give us useful insight for our following analysis in Section 2.4.2. By inspection, we see that one possible basis for the space $d\mathcal{G}^\perp$ is given by:

$$d\mathcal{G}^\perp = \underset{\text{col.}}{\text{span}} \begin{bmatrix} 1 & 0 & -y_R \\ 0 & 1 & x_R \\ 0 & 0 & 1 \\ 1 & 0 & -y_L \\ 0 & 1 & x_L \end{bmatrix} \triangleq \text{span} \begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \mathbf{n}_3 \end{bmatrix} \tag{2.19}$$

From the structure of the vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ we see that a change in the state by $\Delta\mathbf{x} = \alpha\mathbf{n}_1 + \beta\mathbf{n}_2$, $\alpha, \beta \in \mathbb{R}$, corresponds to a "shifting" of the $x-y$ plane by α units along x, and by β units along y. Thus, if the robot and landmark positions are shifted equally, the states $\mathbf{x}$ and $\mathbf{x} + \Delta\mathbf{x}$ will be indistinguishable given the measurements. To understand the physical meaning of $\mathbf{n}_3$, we consider the case where the $x-y$ plane is rotated by a small angle $\delta\phi$. Rotating the coordinate system transforms any point $\mathbf{p} = [x \;\; y]^T$ to a point $\mathbf{p}' = [x' \;\; y']^T$, i.e.,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{C}(\delta\phi) \begin{bmatrix} x \\ y \end{bmatrix} \simeq \begin{bmatrix} 1 & -\delta\phi \\ \delta\phi & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \delta\phi \begin{bmatrix} -y \\ x \end{bmatrix}$$

where we have employed the small angle approximations $c(\delta\phi) \simeq 1$ and $s(\delta\phi) \simeq \delta\phi$. Using this result, we see that if the plane containing the robot and landmarks is rotated by $\delta\phi$, the

SLAM state vector will change to:

$$\mathbf{x}' = \begin{bmatrix} x'_R \\ y'_R \\ \phi'_R \\ x'_L \\ y'_L \end{bmatrix} \simeq \begin{bmatrix} x_R \\ y_R \\ \phi_R \\ x_L \\ y_L \end{bmatrix} + \delta\phi \begin{bmatrix} -y_R \\ x_R \\ 1 \\ -y_L \\ x_L \end{bmatrix} = \mathbf{x} + \delta\phi\mathbf{n}_3 \tag{2.20}$$

which indicates that the vector $\mathbf{n}_3$ corresponds to a rotation of the $x-y$ plane. Since $\mathbf{n}_3 \in d\mathcal{G}^\perp$, this result shows that any such rotation is unobservable, and will cause no change to the measurements. The preceding analysis for the meaning of the basis vectors of $d\mathcal{G}^\perp$ agrees with intuition, which dictates that the *global coordinates* of the state vector in SLAM (rotation and translation) are unobservable.

2.4.2 Linearized SLAM observability analysis

Since the standard EKF employs the linearized system model defined by (2.6) and (2.11) for propagating and updating the state and covariance estimates, the observability properties of this model significantly affect the performance of the estimator as shown below. It is important to note that, in general, the Jacobian matrices $\mathbf{\Phi}_k$, $\mathbf{G}_k$, and $\mathbf{H}_k$ used in the EKF linearized error-state model [see (2.6) and (2.11)] are defined as:

$$\mathbf{\Phi}_k = \nabla_{\mathbf{x}_k}\mathbf{f}\Big|_{\{\mathbf{x}^\star_{k|k},\mathbf{x}^\star_{k+1|k},\mathbf{0}\}}, \quad \mathbf{G}_k = \nabla_{\mathbf{w}_k}\mathbf{f}\Big|_{\{\mathbf{x}^\star_{k|k},\mathbf{0}\}}, \quad \mathbf{H}_k = \nabla_{\mathbf{x}_k}\mathbf{h}\Big|_{\{\mathbf{x}^\star_{k|k-1}\}} \tag{2.21}$$

where $\mathbf{x}^\star_{k|k-1}$ and $\mathbf{x}^\star_{k|k}$ denote the *linearization points* for the state $\mathbf{x}_k$, used for evaluating the Jacobians before and after the EKF update at time-step k, respectively, while a linearization point equal to the zero vector is chosen for the zero-mean noise. Since the linearized error-state model is time-varying, we employ the *local observability matrix* [30, 112] to perform the observability analysis. Specifically, the local observability matrix for the time interval between

time-steps k_o and k_o+m is defined by:

$$\mathbf{M} \triangleq \begin{bmatrix} \mathbf{H}_{k_o} \\ \mathbf{H}_{k_o+1}\mathbf{\Phi}_{k_o} \\ \vdots \\ \mathbf{H}_{k_o+m}\mathbf{\Phi}_{k_o+m-1}\cdots\mathbf{\Phi}_{k_o} \end{bmatrix} \tag{2.22}$$

$$= \begin{bmatrix} \mathbf{H}_{R_{k_o}} & \mathbf{H}_{L_{k_o}} \\ \mathbf{H}_{R_{k_o+1}}\mathbf{\Phi}_{R_{k_o}} & \mathbf{H}_{L_{k_o+1}} \\ \vdots & \vdots \\ \mathbf{H}_{R_{k_o+m}}\mathbf{\Phi}_{R_{k_o+m-1}}\cdots\mathbf{\Phi}_{R_{k_o}} & \mathbf{H}_{L_{k_o+m}} \end{bmatrix} \tag{2.23}$$

$$= \mathbf{M}(\mathbf{x}^\star_{k_o|k_o-1}, \mathbf{x}^\star_{k_o|k_o}, \ldots, \mathbf{x}^\star_{k_o+m|k_o+m-1}) \tag{2.24}$$

The last expression (2.24) makes explicit the fact that the observability matrix is a function of the linearization points used for computing all the Jacobians within the time interval $[k_o, k_o+m]$. In turn, this implies that *the choice of linearization points affects the observability properties* of the linearized error-state system of the EKF. This key fact truly is the basis of our ensuing analysis, where we discuss different possible choices of linearization points, and the observability properties of the corresponding linearized systems.

Ideal EKF-SLAM

Before considering the rank of the matrix $\mathbf{M}$, which is constructed using the *estimated* values of the state in the filter Jacobians, it is interesting to study the observability properties of the "oracle", or "ideal" EKF (i.e., the filter whose Jacobians are evaluated using the *true* values of the state variables, in other words, $\mathbf{x}^\star_{k|k-1} = \mathbf{x}^\star_{k|k} = \mathbf{x}_k$, for all k). In the following, all matrices evaluated using the true state values are denoted by the symbol "$\breve{}$".

We start by noting that [see (2.8)]:

$$\breve{\mathbf{\Phi}}_{R_{k_o+1}}\breve{\mathbf{\Phi}}_{R_{k_o}} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\mathbf{p}_{R_{k_o+2}} - \mathbf{p}_{R_{k_o}}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{2.25}$$

Based on this property, it is easy to show by induction that:

$$\breve{\mathbf{\Phi}}_{R_{k_o+\ell-1}}\breve{\mathbf{\Phi}}_{R_{k_o+\ell-2}}\cdots\breve{\mathbf{\Phi}}_{R_{k_o}} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\mathbf{p}_{R_{k_o+\ell}} - \mathbf{p}_{R_{k_o}}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$$

which holds for all $\ell > 0$. Using this result, and substituting for the measurement Jacobians from (2.12) and (2.13), we can prove the following useful identity:

$$\begin{aligned}\breve{\mathbf{H}}_{R_{k_o+\ell}}\breve{\mathbf{\Phi}}_{R_{k_o+\ell-1}}\cdots\breve{\mathbf{\Phi}}_{R_{k_o}} &= (\nabla\breve{\mathbf{h}}_{k_o+\ell})\mathbf{C}^T(\phi_{R_{k_o+\ell}})\begin{bmatrix}-\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_L-\mathbf{p}_{R_{k_o}})\end{bmatrix}\\ &= \breve{\mathbf{H}}_{L_{k_o+\ell}}\begin{bmatrix}-\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_L-\mathbf{p}_{R_{k_o}})\end{bmatrix}\end{aligned} \tag{2.26}$$

which holds for all $\ell > 0$. The observability matrix $\breve{\mathbf{M}}$ can now be written as:

$$\breve{\mathbf{M}} = \underbrace{\mathbf{Diag}\left(\breve{\mathbf{H}}_{L_{k_o}}, \breve{\mathbf{H}}_{L_{k_o+1}}, \cdots, \breve{\mathbf{H}}_{L_{k_o+m}}\right)}_{\breve{\mathbf{D}}} \underbrace{\begin{bmatrix}-\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_L-\mathbf{p}_{R_{k_o}}) & \mathbf{I}_2\\ -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_L-\mathbf{p}_{R_{k_o}}) & \mathbf{I}_2\\ \vdots & \vdots & \vdots\\ -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_L-\mathbf{p}_{R_{k_o}}) & \mathbf{I}_2\end{bmatrix}}_{\breve{\mathbf{U}}} \tag{2.27}$$

Lemma 2.4.3. *The rank of the observability matrix, $\breve{\mathbf{M}}$, of the ideal EKF-SLAM is 2.*

Proof. The rank of the product of the matrices $\breve{\mathbf{D}}$ and $\breve{\mathbf{U}}$ is given by (see (4.5.1) in [113]):

$$\mathrm{rank}(\breve{\mathbf{D}}\breve{\mathbf{U}}) = \mathrm{rank}(\breve{\mathbf{U}}) - \dim\left(\mathrm{null}(\breve{\mathbf{D}})\bigcap\mathrm{rng}(\breve{\mathbf{U}})\right) \tag{2.28}$$

where $\mathrm{null}(\cdot)$ denotes the right null space of a matrix, $\mathrm{rng}(\cdot)$ represents the matrix range, and $\dim(\cdot)$ the dimension of a subspace. Since $\breve{\mathbf{U}}$ comprises $m+1$ repetitions of the same 2×5 block row, it is clear that $\mathrm{rank}(\breve{\mathbf{U}}) = 2$, and the range of $\breve{\mathbf{U}}$, $\mathrm{rng}(\breve{\mathbf{U}})$, is spanned by the vectors $\mathbf{u}_1$ and $\mathbf{u}_2$, defined as follows:

$$\begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2\end{bmatrix} = \begin{bmatrix}\mathbf{I}_2\\ \vdots\\ \mathbf{I}_2\end{bmatrix} \tag{2.29}$$

We now observe that in general $\breve{\mathbf{D}}\mathbf{u}_i \neq \mathbf{0}$, for $i = 1, 2$. Moreover, note that any vector $\mathbf{y} \in \mathrm{rng}(\breve{\mathbf{U}}) \setminus \mathbf{0}$ can be written as $\mathbf{y} = \alpha_1\mathbf{u}_1 + \alpha_2\mathbf{u}_2$ for some $\alpha_1, \alpha_2 \in \mathbb{R}$, where α_1 and α_2 are not simultaneously equal to zero. Thus, we see that in general $\breve{\mathbf{D}}\mathbf{y} = \alpha_1\breve{\mathbf{D}}\mathbf{u}_1 + \alpha_2\breve{\mathbf{D}}\mathbf{u}_2 \neq \mathbf{0}$, which implies that $\mathbf{y}$ does not belong to the nullspace of $\breve{\mathbf{D}}$, $\mathrm{null}(\breve{\mathbf{D}})$. Therefore, $\dim(\mathrm{null}(\breve{\mathbf{D}})\bigcap\mathrm{rng}(\breve{\mathbf{U}})) = 0$, and, finally, $\mathrm{rank}(\breve{\mathbf{M}}) = \mathrm{rank}(\breve{\mathbf{U}}) - \dim(\mathrm{null}(\breve{\mathbf{D}})\bigcap\mathrm{rng}(\breve{\mathbf{U}})) = \mathrm{rank}(\breve{\mathbf{U}}) = 2$. □

Most importantly, it can be easily verified that a basis for the right nullspace of $\breve{\mathbf{U}}$ (and thus for the right nullspace of $\breve{\mathbf{M}}$) is given by the vectors shown in (2.19). Thus, the unobservable subspace of the ideal EKF system model is *identical* to the space $d\mathcal{G}^{\perp}$, which contains the unobservable directions of the nonlinear SLAM system. We therefore see that if it was possible to evaluate the Jacobians using the true state values, the linearized error-state model employed in the EKF would have observability properties similar to those of the nonlinear SLAM system.

The preceding analysis was carried out for the case where a single landmark is included in the state vector. We now examine the more general case where $M > 1$ landmarks are included in the state. Suppose the M landmarks are observed at time-step $k_o + \ell$ ($\ell > 0$), then the measurement matrix $\breve{\mathbf{H}}_{k_o+\ell}$ is given by:[2]

$$\breve{\mathbf{H}}_{k_o+\ell} = \begin{bmatrix} \breve{\mathbf{H}}^{(1)}_{R_{k_o+\ell}} & \breve{\mathbf{H}}^{(1)}_{L_{k_o+\ell}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \breve{\mathbf{H}}^{(M)}_{R_{k_o+\ell}} & \mathbf{0} & \cdots & \breve{\mathbf{H}}^{(M)}_{L_{k_o+\ell}} \end{bmatrix} \tag{2.30}$$

where $\breve{\mathbf{H}}^{(i)}_{R_{k_o+\ell}}$ and $\breve{\mathbf{H}}^{(i)}_{L_{k_o+\ell}}$ ($i = 1, 2, ..., M$), are obtained by (2.12) and (2.13) using the true values of the states, respectively. The observability matrix $\breve{\mathbf{M}}$ now becomes:

$$\breve{\mathbf{M}} = \begin{bmatrix} \breve{\mathbf{H}}^{(1)}_{R_{k_o}} & \breve{\mathbf{H}}^{(1)}_{L_{k_o}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \breve{\mathbf{H}}^{(M)}_{R_{k_o}} & \mathbf{0} & \cdots & \breve{\mathbf{H}}^{(M)}_{L_{k_o}} \\ \breve{\mathbf{H}}^{(1)}_{R_{k_o+1}}\breve{\mathbf{\Phi}}_{R_{k_o}} & \breve{\mathbf{H}}^{(1)}_{L_{k_o+1}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \breve{\mathbf{H}}^{(M)}_{R_{k_o+1}}\breve{\mathbf{\Phi}}_{R_{k_o}} & \mathbf{0} & \cdots & \breve{\mathbf{H}}^{(M)}_{L_{k_o+1}} \\ \vdots & \vdots & \vdots & \vdots \\ \breve{\mathbf{H}}^{(1)}_{R_{k_o+m}}\breve{\mathbf{\Phi}}_{R_{k_o+m-1}}\cdots\breve{\mathbf{\Phi}}_{R_{k_o}} & \breve{\mathbf{H}}^{(1)}_{L_{k_o+m}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \breve{\mathbf{H}}^{(M)}_{R_{k_o+m}}\breve{\mathbf{\Phi}}_{R_{k_o+m-1}}\cdots\breve{\mathbf{\Phi}}_{R_{k_o}} & \mathbf{0} & \cdots & \breve{\mathbf{H}}^{(M)}_{L_{k_o+m}} \end{bmatrix} \tag{2.31}$$

[2] We here assume that all M landmarks are observed at every time step in the time interval $[k_o,\ k_o + m]$. This is done only to simplify the notation, and is not a necessary assumption in the analysis.

Using the identity (2.26), substitution of the Jacobian matrices in (2.31) yields:

$$\breve{\mathbf{M}} = \underbrace{\mathbf{Diag}\left(\breve{\mathbf{H}}^{(1)}_{L_{k_o}}, \ \cdots, \breve{\mathbf{H}}^{(M)}_{L_{k_o+m}}\right)}_{\breve{\mathbf{D}}} \underbrace{\begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_{L_1} - \mathbf{p}_{R_{k_o}}) & \mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_{L_M} - \mathbf{p}_{R_{k_o}}) & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 \\ \\ -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_{L_1} - \mathbf{p}_{R_{k_o}}) & \mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_{L_M} - \mathbf{p}_{R_{k_o}}) & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 \\ \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \\ -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_{L_1} - \mathbf{p}_{R_{k_o}}) & \mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_{L_M} - \mathbf{p}_{R_{k_o}}) & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 \end{bmatrix}}_{\breve{\mathbf{U}}} \tag{2.32}$$

Clearly, the matrix $\breve{\mathbf{U}}$ now consists of $m+1$ repetitions of the M block rows:

$$\begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_{L_i} - \mathbf{p}_{R_{k_o}}) & \mathbf{0}_{2\times 2} & \cdots & \underbrace{\mathbf{I}_2}_{i\text{th landmark}} & \cdots & \mathbf{0}_{2\times 2} \end{bmatrix}$$

for $i = 1, 2, ..., M$. Therefore, $\mathrm{rank}(\breve{\mathbf{M}}) = 2M$. Furthermore, by inspection, a possible basis for the right nullspace of $\breve{\mathbf{M}}$ is given by:

$$\mathrm{null}(\breve{\mathbf{M}}) = \underset{\text{col.}}{\mathrm{span}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\mathbf{p}_{R_{k_o}} \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\mathbf{p}_{L_1} \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}\mathbf{p}_{L_M} \end{bmatrix} \tag{2.33}$$

By noting the similarity of this result with that of (2.19), the physical interpretation of this result is analogous to that of the single-landmark case: the global translation and orientation of the state vector are unobservable.

Standard EKF-SLAM

We now study the observability properties of the standard EKF-SLAM, in which the Jacobians are evaluated at the latest state estimates (i.e., $\mathbf{x}^{\star}_{k|k-1} = \hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{x}^{\star}_{k|k} = \hat{\mathbf{x}}_{k|k}$, for all k). Similarly, we begin by examining the single-landmark case. By deriving an expression analogous to that of (2.25), we obtain (see Section 2.4.2):

$$\mathbf{\Phi}_{R_{k_o+1}}\mathbf{\Phi}_{R_{k_o}} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{R_{k_o+2|k_o+1}} - \hat{\mathbf{p}}_{R_{k_o|k_o}} - \Delta\mathbf{p}_{R_{k_o+1}}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$$

where $\Delta\mathbf{p}_{R_{k_o+1}} \triangleq \hat{\mathbf{p}}_{R_{k_o+1|k_o+1}} - \hat{\mathbf{p}}_{R_{k_o+1|k_o}}$ is the correction in the robot position due to the EKF update at time-step k_o+1. Using induction, we can show that:

$$\mathbf{\Phi}_{R_{k_o+\ell-1}}\mathbf{\Phi}_{R_{k_o+\ell-2}}\cdots\mathbf{\Phi}_{R_{k_o}} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{R_{k_o+\ell|k_o+\ell-1}} - \hat{\mathbf{p}}_{R_{k_o|k_o}} - \sum_{j=k_o+1}^{k_o+\ell-1}\Delta\mathbf{p}_{R_j}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{2.34}$$

where $\ell > 0$. Therefore [see (2.11), (2.12), and (2.13)]

$$\mathbf{H}_{R_{k_o+\ell}}\mathbf{\Phi}_{R_{k_o+\ell-1}}\cdots\mathbf{\Phi}_{R_{k_o}} = \mathbf{H}_{L_{k_o+\ell}}\begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o+\ell|k_o+\ell-1}} - \hat{\mathbf{p}}_{R_{k_o|k_o}} - \sum_{j=k_o+1}^{k_o+\ell-1}\Delta\mathbf{p}_{R_j}\right)\end{bmatrix} \tag{2.35}$$

Using this result, we can write $\mathbf{M}$ as [see (2.22)]:

$$\mathbf{M} = \underbrace{\mathbf{Diag}\left(\mathbf{H}_{L_{k_o}}, \mathbf{H}_{L_{k_o+1}}, \cdots, \mathbf{H}_{L_{k_o+m}}\right)}_{\mathbf{D}} \underbrace{\begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o|k_o-1}} - \hat{\mathbf{p}}_{R_{k_o|k_o-1}}\right) & \mathbf{I}_2 \\ -\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o+1|k_o}} - \hat{\mathbf{p}}_{R_{k_o|k_o}}\right) & \mathbf{I}_2 \\ -\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o+2|k_o+1}} - \hat{\mathbf{p}}_{R_{k_o|k_o}} - \Delta\mathbf{p}_{R_{k_o+1}}\right) & \mathbf{I}_2 \\ \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o+m|k_o+m-1}} - \hat{\mathbf{p}}_{R_{k_o|k_o}} - \sum_{j=k_o+1}^{k_o+m-1}\Delta\mathbf{p}_{R_j}\right) & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{U}} \tag{2.36}$$

Lemma 2.4.4. *The rank of the observability matrix,* $\mathbf{M}$*, of the standard EKF-SLAM is 3.*

Proof. First, we note that the estimates of any given state variable at different time instants are generally different. Hence, in contrast to the case of the ideal EKF-SLAM, the following inequalities generally hold: $\hat{\mathbf{p}}_{R_{k_o+i|k_o+i-1}} \neq \hat{\mathbf{p}}_{R_{k_o+i|k_o+i}}$ and $\hat{\mathbf{p}}_{L_{k_o+i|k_o+i-1}} \neq \hat{\mathbf{p}}_{L_{k_o+\ell|k_o+\ell-1}}$, for $i \neq \ell$. Therefore, the third column of $\mathbf{U}$ will be, in general, a vector with unequal elements, and thus $\mathrm{rank}(\mathbf{U}) = 3$. Proceeding similarly to the proof of Lemma 2.4.3, we first find one

possible basis for the range space of $\mathbf{U}$, $\mathrm{rng}(\mathbf{U})$. By inspection, we see that such a basis is given simply by the first 3 columns of $\mathbf{U}$, which we denote by $\mathbf{u}_i$ $(i = 1, 2, 3)$. Moreover, it can be verified that generally $\mathbf{D}\mathbf{u}_i \neq \mathbf{0}$. Therefore, $\dim(\mathrm{null}(\mathbf{D}) \bigcap \mathrm{rng}(\mathbf{U})) = 0$, and finally $\mathrm{rank}(\mathbf{M}) = \mathrm{rank}(\mathbf{U}) - \dim(\mathrm{null}(\mathbf{D}) \bigcap \mathrm{rng}(\mathbf{U})) = \mathrm{rank}(\mathbf{U}) = 3$. □

We thus see that the linearized error-state model employed in the standard EKF-SLAM has different observability properties than that of the ideal EKF-SLAM (see Lemma 2.4.3) and that of the underlying nonlinear system (see Lemma 2.4.2). In particular, by processing the measurements collected in the interval $[k_o, k_o + m]$, the filter acquires information in 3 dimensions of the state space (along the directions corresponding to the observable subspace of the EKF). However, the measurements actually provide information in only 2 directions of the state space (i.e., the robot-to-landmark relative position). As a result, the EKF gains "spurious information" along the unobservable directions of the underlying nonlinear SLAM system, which leads to inconsistency.

To probe further, we note that the basis of the right nullspace of $\mathbf{M}$ is given by:

$$\mathrm{null}(\mathbf{M}) = \underset{\text{col.}}{\mathrm{span}} \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{0}_{1\times 2} \\ \mathbf{I}_2 \end{bmatrix} = \mathrm{span} \begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 \end{bmatrix} \tag{2.37}$$

Note that these two vectors correspond to a shifting of the $x-y$ plane, which implies that such a shifting is unobservable. On the other hand, the direction corresponding to the global orientation is "missing" from the unobservable subspace of the EKF system model [see (2.19) and (2.20)]. Therefore, we see that the filter will gain "nonexistent" information about the robot's global orientation. This will lead to an unjustified reduction in the orientation uncertainty, which will, in turn, further reduce the uncertainty in all the state variables. This agrees in some respects with [12, 79], where it was argued that the orientation uncertainty is the main cause of the filter's inconsistency in SLAM. However, we point out that the *root cause* of the problem is that the linearization points used for computing the Jacobians in the standard EKF-SLAM (i.e., the latest state estimates) change the dimension of the observable subspace, and thus fundamentally alter the properties of the estimation process.

Similar conclusions can be drawn when $M > 1$ landmarks are included in the state vector.

In particular, in this case the observability matrix can be found as follows:

$$\mathbf{M} = \underbrace{\mathbf{Diag}\left((\nabla \mathbf{h}_{k_o}^{(1)})\mathbf{C}^T(\hat{\phi}_{R_{k_o|k_o-1}}), \cdots, (\nabla \mathbf{h}_{k_o+m}^{(M)})\mathbf{C}^T(\hat{\phi}_{R_{k_o+m|k_o+m-1}})\right)}_{\mathbf{D}} \times$$

$$\underbrace{\begin{bmatrix}
-\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o|k_o-1}}^{(1)} - \hat{\mathbf{p}}_{R_{k_o|k_o-1}}\right) & \mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o|k_o-1}}^{(M)} - \hat{\mathbf{p}}_{R_{k_o|k_o-1}}\right) & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 \\
-\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o+1|k_o}}^{(1)} - \hat{\mathbf{p}}_{R_{k_o|k_o}}\right) & \mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o+1|k_o}}^{(M)} - \hat{\mathbf{p}}_{R_{k_o|k_o}}\right) & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 \\
-\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o+2|k_o+1}}^{(1)} - \hat{\mathbf{p}}_{R_{k_o|k_o}} - \Delta\mathbf{p}_{R_{k_o+1}}\right) & \mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o+2|k_o+1}}^{(M)} - \hat{\mathbf{p}}_{R_{k_o|k_o}} - \Delta\mathbf{p}_{R_{k_o+1}}\right) & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 \\
\\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\\
-\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o+m|k_o+m-1}}^{(1)} - \hat{\mathbf{p}}_{R_{k_o|k_o}} - \sum_{j=k_o+1}^{k_o+m-1}\Delta\mathbf{p}_{R_j}\right) & \mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k_o+m|k_o+m-1}}^{(M)} - \hat{\mathbf{p}}_{R_{k_o|k_o}} - \sum_{j=k_o+1}^{k_o+m-1}\Delta\mathbf{p}_{R_j}\right) & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2
\end{bmatrix}}_{\mathbf{U}} \tag{2.38}$$

The nullspace of the observability matrix (2.38) can be shown to be equal to:

$$\text{null}(\mathbf{M}) = \underset{\text{col.}}{\text{span}} \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{0}_{1\times 2} \\ \mathbf{I}_2 \\ \vdots \\ \mathbf{I}_2 \end{bmatrix} \tag{2.39}$$

We thus see that the global orientation is erroneously observable in this case as well, which leads to inconsistent estimates.

An interesting remark is that the covariance matrices of the system and measurement noise do not appear in the observability analysis of the filter's system model. Therefore, even if these covariance matrices are artificially inflated, the filter will retain the same observability properties (i.e., the same observable and unobservable subspaces). This shows that no amount of covariance inflation can result in correct observability properties. Similarly, even if the IEKF [14]

is employed for state estimation, the same, erroneous, observability properties will arise, since the landmark position estimates will generally differ at different time steps.

2.5 Observability-Constrained (OC)-EKF SLAM algorithms

We have seen from the preceding section that when the EKF Jacobians are evaluated using the latest state estimates, the EKF error-state model has an observable subspace of higher dimension than the actual nonlinear SLAM system. This will always lead to an unjustified reduction of the estimated covariance, and thus inconsistency. We now propose a general framework for addressing this problem.

Our key conjecture is that, by ensuring an unobservable subspace of appropriate dimension, we can avoid the influx of spurious information in the erroneously observable direction of the state space, and thus improve the consistency of the estimates. To do so, we propose computing the EKF Jacobians in such a way that guarantees the linearized error-state system model has an unobservable subspace of dimension three. This corresponds to satisfying conditions (2.40)-(2.41) of the following lemma:

Lemma 2.5.1. *If the EKF Jacobians $\mathbf{\Phi}_k$ and $\mathbf{H}_{k+1}$ at every time step, are computed so as to fulfill the following conditions:*

$$\mathbf{H}_{k_o}\mathbf{N} = \mathbf{0}\ ,\ \ \text{for}\ \ell = 0 \tag{2.40}$$

$$\mathbf{H}_{k_o+\ell}\mathbf{\Phi}_{k_o+\ell-1}\cdots\mathbf{\Phi}_{k_o}\mathbf{N} = \mathbf{0}\ ,\ \ \forall\,\ell > 0 \tag{2.41}$$

where $\mathbf{N}$ is a full-rank matrix whose 3 column vectors define the desired unobservable subspace, then the corresponding observability matrix is of correct rank $\dim(\mathbf{x}) - 3$.

Proof. When (2.40)-(2.41) hold, then all the block rows of the observability matrix [see (2.22)] will have the same nullspace, spanned by the columns of $\mathbf{N}$. □

Essentially, the selection of $\mathbf{N}$ is a design choice, which allows us to control the unobservable subspace of the resulting EKF system model. Ideally, we would like the column vectors of $\mathbf{N}$ to be identical to those in (2.19), which define the unobservable directions of the actual nonlinear systems. However, this cannot be achieved in practice, since these directions depend on the *true* values of the states, which are unavailable during any real-world implementation. A

natural selection, which is realizable in practice, is to define the unobservable subspace using the *first-available* state estimates, i.e., for the single-landmark case to choose[3]

$$\mathbf{N} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{R_{k_o|k_o-1}} \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{L_{k_o|k_o}} \end{bmatrix} \tag{2.42}$$

Note that the matrix $\mathbf{N}$ in (2.42) satisfies condition (2.40), since $\mathbf{H}_{k_o}$ is the first block row of the observability matrix in (2.22). It is also important to note that advanced initialization techniques, such as delayed-state initialization [107], can be used to improve the first estimates' accuracy and thus yield an unobservable subspace as close as possible to the true one. Once $\mathbf{N}$ is selected, the next step is to appropriately compute the filter Jacobians so as to ensure that (2.41) holds. Clearly, several options exist, each of which leads to a different algorithm within the general framework described here. In what follows, we present three different Observability-Constrained (OC)-EKF algorithms to achieve this goal.

2.5.1 OC-EKF1

We start by describing the first version of our OC-EKF algorithms, the OC-EKF1, which was originally proposed in [70]. The key idea of this approach is to choose the *first-ever-available* state estimates as the linearization points [and hence it is also termed as First-Estimates-Jacobian (FEJ)-EKF] so as to guarantee the appropriate observability properties of the EKF linearized system. The procedure of the OC-EKF1 SLAM is explained in detail by the following lemma:

Lemma 2.5.2. *If the linearization points, at which the filter Jacobians* $\mathbf{\Phi}_{R_k} = \mathbf{\Phi}_{R_k}(\mathbf{x}^\star_{R_{k+1|k}}, \mathbf{x}^\star_{R_{k|k}})$ *and* $\mathbf{H}_k = \mathbf{H}_k(\mathbf{x}^\star_{R_{k|k-1}}, \mathbf{p}^\star_{L_{k|k-1}})$ *are evaluated, are selected as:*

$$\mathbf{x}^\star_{R_{k+1|k}} = \hat{\mathbf{x}}_{R_{k+1|k}}\ ,\ \ \mathbf{x}^\star_{R_{k|k}} = \hat{\mathbf{x}}_{R_{k|k-1}}\ ,\quad \mathbf{x}^\star_{R_{k|k-1}} = \hat{\mathbf{x}}_{R_{k|k-1}}\ ,\ \ \mathbf{p}^\star_{L_{k|k-1}} = \hat{\mathbf{p}}_{L_{k_o|k_o}} \tag{2.43}$$

then it is guaranteed that the unobservable subspace of the resulting EKF linearized SLAM error-state model is of dimension 3.

Proof. Using the linearization points (2.43), as compared to the standard EKF, we have the following two changes in computing the filter Jacobians:

[3] In the case where multiple ($M > 1$) landmarks are included in the state vector, $\mathbf{N}$ can be chosen analogously, augmented by a new block row, $\begin{bmatrix}\mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{L_i,k_o|k_o}\end{bmatrix}$, corresponding to each landmark, L_i ($i = 1, 2, \ldots, M$) [72].

1. Instead of computing the state-propagation Jacobian matrix $\mathbf{\Phi}_{R_k}$ as in (2.8), we employ the expression:

$$\mathbf{\Phi}'_{R_k} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{R_{k+1|k}} - \hat{\mathbf{p}}_{R_{k|k-1}}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{2.44}$$

   The difference compared to (2.8) is that the prior robot position estimate, $\hat{\mathbf{p}}_{R_{k|k-1}}$, is used in place of the posterior estimate, $\hat{\mathbf{p}}_{R_{k|k}}$.

2. In the evaluation of the measurement Jacobian matrix $\mathbf{H}_{k+1}$ [see (2.11), (2.12), and (2.13)], we always utilize the landmark estimate *from the first time* the landmark was detected and initialized. Thus, if a landmark was first seen at time-step k_o, we compute the measurement Jacobian as:

$$\begin{aligned} \mathbf{H}'_{k+1} &= \begin{bmatrix} \mathbf{H}_{R_{k+1}} & \mathbf{H}_{L_{k+1}} \end{bmatrix} \\ &= (\nabla \mathbf{h}_{k+1}) \mathbf{C}^T(\hat{\phi}_{R_{k+1|k}}) \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_{k_o|k_o}} - \hat{\mathbf{p}}_{R_{k+1|k}}) & \mathbf{I}_2 \end{bmatrix} \end{aligned} \tag{2.45}$$

As a result of the above modifications, only the *first* estimates of all landmark positions and all robot poses appear in the filter Jacobians. It is easy to verify that the above Jacobians satisfy (2.40) and (2.41) for the choice of $\mathbf{N}$ in (2.42). Thus, the OC-EKF1 SLAM is based on an error-state system model whose unobservable subspace is of dimension 3. □

2.5.2 OC-EKF2

We now describe the second version of our OC-EKF algorithms, the OC-EKF2. We first note that, when linearizing a generic scalar nonlinear function $f(x)$ around a point $x^\star$, the linearization error depends on the accuracy of the linearization point $x^\star$; if $x^\star$ is inaccurate, the linearization error will be large. To see this, using Taylor's theorem, we have:

$$f(x) = f(x^\star) + f'(x^\star)(x - x^\star) + \frac{f''(\xi)}{2}(x - x^\star)^2 \tag{2.46}$$

where the last quadratic term, $\frac{f''(\xi)}{2}(x - x^\star)^2$, defines the linearization error, and ξ is a point that lies between x and $x^\star$. It is clear that the linearization error is approximately proportional to the error square of the linearization point, $(x - x^\star)^2$. This result will be useful for the ensuing derivations.

Even though the OC-EKF1 typically performs substantially better than the standard EKF (see Sections 2.6 and 2.7), it relies heavily on the initial state estimates; if these estimates are inaccurate, the linearization errors become large and thus the performance of the estimator may degrade. This could be the case when the first estimates of the landmark positions are of poor quality (e.g., in bearing-only SLAM). We note that the selection of the linearization points employed in the OC-EKF1 is not the only one that ensures that (2.41) is satisfied. However, all linearization points chosen in a way that only fulfills (2.41), can result in large linearization errors, and thus improved performance cannot always be guaranteed. On the other hand, we also know that, in the standard EKF, the latest, and thus best, state estimates are used as the linearization points, which, in general, have the smallest linearization errors. However, as shown in Section 2.4.2, since the linearization points (i.e., the best state estimates) do not satisfy the observability condition (2.41), filter inconsistency occurs, which degrades performance.

Therefore, we propose selecting the linearization points of the EKF so as to minimize the linearization errors while satisfying the observability conditions (2.40)-(2.41). This can be formulated as a constrained minimization problem where the constraints express the observability requirements. Specifically, at time-step $k+1$, we aim at minimizing the linearization error of the points $\mathbf{x}^\star_{R_{k|k}}$ and $\mathbf{x}^\star_{k+1|k}$, which appear in the filter Jacobians $\mathbf{\Phi}_k$ and $\mathbf{H}_{k+1}$, subject to the observability constraint (2.41). Mathematically, this is expressed as:

$$\min_{\mathbf{x}^\star_{R_{k|k}}, \mathbf{x}^\star_{k+1|k}} \int \left\| \mathbf{x}_{R_k} - \mathbf{x}^\star_{R_{k|k}} \right\|^2 p(\mathbf{x}_{R_k}|\mathbf{z}_{0:k}) d\mathbf{x}_{R_k} + \int \left\| \mathbf{x}_{k+1} - \mathbf{x}^\star_{k+1|k} \right\|^2 p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k}) d\mathbf{x}_{k+1} \quad (2.47)$$

$$\text{subject to } \mathbf{H}_{k+1}\mathbf{\Phi}_k \cdots \mathbf{\Phi}_{k_o}\mathbf{N} = \mathbf{0}\,, \;\; \forall k \geq k_o \quad (2.48)$$

where $\mathbf{z}_{0:k}$ denotes all the measurements available during the time interval $[0, k]$. Note that during EKF propagation, since the landmarks are static, only the robot pose participates in the linearization process [see (2.6)], while during EKF update, both the robot pose and the landmark positions are involved in the linearization of measurement equation [see (2.11)]. This justifies the choice of above optimization variables. In general, the constrained minimization problem (2.47)-(2.48) is intractable. However, when the two pdfs, $p(\mathbf{x}_{R_k}|\mathbf{z}_{0:k})$ and $p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k})$, are Gaussian distributions (which is the assumption employed in the EKF), we can solve the problem *analytically*.

We now show how the closed-form solution can be computed for the case where only one landmark is included in the state vector. We note that the following lemma will be helpful for the ensuing derivations:

Lemma 2.5.3. *The constrained optimization problem* (2.47)-(2.48) *is equivalent to:*

$$\min_{\mathbf{x}^\star_{R_{k|k}},\, \mathbf{x}^\star_{k+1|k}} \left\|\hat{\mathbf{x}}_{R_{k|k}} - \mathbf{x}^\star_{R_{k|k}}\right\|^2 + \left\|\hat{\mathbf{x}}_{k+1|k} - \mathbf{x}^\star_{k+1|k}\right\|^2 \tag{2.49}$$

$$\text{subject to } \mathbf{p}^\star_{L_{k+1|k}} - \mathbf{p}^\star_{R_{k|k}} = \hat{\mathbf{p}}_{L_{k_o|k_o}} - \mathbf{p}^\star_{R_{k|k-1}} + \sum_{j=k_o}^{k-1} \Delta\mathbf{p}^\star_{R_j} \tag{2.50}$$

where $\Delta\mathbf{p}^\star_{R_j} \triangleq \mathbf{p}^\star_{R_{j|j}} - \mathbf{p}^\star_{R_{j|j-1}}$.

Proof. See Appendix A.4. □

Using the technique of Lagrangian multipliers, the optimal solution to the problem (2.49)-(2.50) can be obtained as:

$$\begin{aligned} \mathbf{p}^\star_{R_{k|k}} &= \hat{\mathbf{p}}_{R_{k|k}} + \frac{\boldsymbol{\lambda}_k}{2}\,, \quad \phi^\star_{R_{k|k}} = \hat{\phi}_{R_{k|k}}\,, \\ \mathbf{x}^\star_{R_{k+1|k}} &= \hat{\mathbf{x}}_{R_{k+1|k}}\,, \qquad \mathbf{p}^\star_{L_{k+1|k}} = \hat{\mathbf{p}}_{L_{k+1|k}} - \frac{\boldsymbol{\lambda}_k}{2} \end{aligned} \tag{2.51}$$

with

$$\boldsymbol{\lambda}_k = \left(\hat{\mathbf{p}}_{L_{k+1|k}} - \hat{\mathbf{p}}_{L_{k_o|k_o}}\right) - \left(\hat{\mathbf{p}}_{R_{k|k}} - \mathbf{p}^\star_{R_{k|k-1}} + \sum_{j=k_o}^{k-1} \Delta\mathbf{p}^\star_{R_j}\right)$$

Note that in the case where multiple landmarks are included in the state vector, each landmark imposes a constraint analogous to (2.50), and thus the analytic solution of the optimal linearization points can be obtained similarly [72]. Using the linearization points in (2.51), the filter Jacobians in the OC-EKF2 SLAM are now computed as follows:

1. The state-propagation Jacobian matrix is calculated as:

$$\boldsymbol{\Phi}''_{R_k} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{R_{k+1|k}} - \hat{\mathbf{p}}_{R_{k|k}} - \frac{\boldsymbol{\lambda}_k}{2}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{2.52}$$

2. The measurement Jacobian matrix is calculated as:

$$\begin{aligned} \mathbf{H}''_{k+1} &= \begin{bmatrix} \mathbf{H}_{R_{k+1}} & \mathbf{H}_{L_{k+1}} \end{bmatrix} \\ &= (\nabla\mathbf{h}_{k+1})\mathbf{C}^T(\hat{\phi}_{R_{k+1|k}}) \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}\left(\hat{\mathbf{p}}_{L_{k+1|k}} - \hat{\mathbf{p}}_{R_{k+1|k}} - \frac{\boldsymbol{\lambda}_k}{2}\right) & \mathbf{I}_2 \end{bmatrix} \end{aligned} \tag{2.53}$$

2.5.3 OC-EKF3

We now present our OC-EKF3 algorithm, an alternative to the OC-EKF2. In particular, in contrast to the OC-EKF2 which *indirectly* computes both the propagation and measurement Jacobians by first finding the optimal linearization points, the OC-EKF3 *directly* computes the measurement Jacobian by projecting the most accurate measurement Jacobian onto the observable subspace, while the propagation Jacobian is calculated identically as in the standard EKF [see (2.8)]. As a result, the observable subspace of the linearized EKF system model is guaranteed to have the correct dimensions.

Specifically, we aim to find the measurement Jacobian closest to the ideal one that has the best accuracy, while satisfying the observability constraint (2.41), i.e.,

$$\min_{\mathbf{H}_{k+1}} \ ||\mathbf{H}_o - \mathbf{H}_{k+1}||_F^2 \tag{2.54}$$

$$\text{subject to } \ \mathbf{H}_{k+1}\mathbf{\Phi}_k \cdots \mathbf{\Phi}_{k_o}\mathbf{N} = \mathbf{0} \tag{2.55}$$

where $||\mathbf{\Xi}||_F$ denotes the Frobenius norm of matrix $\mathbf{\Xi}$, and $\mathbf{H}_o$ is the ideal measurement Jacobian evaluated at the true states. However, since in real-world applications the true states are generally unavailable, we instead evaluate it at the current best state estimates as in the standard EKF, i.e., $\mathbf{H}_o = \mathbf{H}_o(\hat{\mathbf{x}}_{k+1|k})$ [see (2.11)]. Therefore, the optimal closed-form solution is obtained by application of the following lemma:

Lemma 2.5.4. *The optimal solution to the constrained minimization problem* (2.54)-(2.55) *is:*

$$\mathbf{H}_{k+1}''' = \mathbf{H}_o \left(\mathbf{I}_{\dim(\mathbf{x})} - \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\right) \tag{2.56}$$

where $\mathbf{V} \triangleq \mathbf{\Phi}_k \cdots \mathbf{\Phi}_{k_o}\mathbf{N}$.

Proof. See Appendix A.5. □

Note that $\mathbf{V}$ in the above equation is the propagated unobservable subspace at time-step $k+1$, and $\left(\mathbf{I}_{\dim(\mathbf{x})} - \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\right)$ is the subspace orthogonal to $\mathbf{V}$, i.e., the observable subspace. Hence, as seen from (2.56), the measurement Jacobian of the OC-EKF3, $\mathbf{H}_{k+1}'''$, is the projection of the best-available measurement Jacobian onto the observable subspace. It is also important to observe that in the case of multiple landmarks, each measurement only depends on the robot pose and the measured landmark, and hence the corresponding measurement Jacobian has sparse structure [see (2.11)]. Based on this observation and by exploiting the sparse structure

of the Jacobian matrix, we only need to consider the nonzero submatrices of the measurement Jacobian and solve a reduced-size problem similar to (2.54)-(2.55). Once the optimal solution is attained, we can easily construct the full measurement Jacobian by padding it with zeros [see (2.11)].

Remarks It is important to note that, as compared to the standard EKF, the *only* change in the three OC-EKF algorithms is the way in which the Jacobians are computed. The state estimates and covariance in the OC-EKFs are propagated and updated in the same way as in the standard EKF. Moreover, we stress that the OC-EKFs are causal and realizable "in the real world," since they do not utilize any knowledge of the future or true states. In summary, the main steps of the proposed OC-EKFs for SLAM are outlined in Algorithm 1.

2.5.4 Relation to prior work

At this point, it is interesting to examine the relation of our analysis, which addresses the general case of a moving robot, to the previous work that has focused on special cases [79, 85]. We first note that the "correct" observability properties of the OC-EKFs are attributed to the fact that conditions (2.40)-(2.41) hold, which is not the case for the standard EKF. Thus, (2.40)-(2.41) can be seen as sufficient conditions that, when satisfied by the filter Jacobians, ensure that the observability matrix has a nullspace of appropriate dimensions. Note also that, due to the identity (2.26), the conditions (2.40)-(2.41) are trivially satisfied by the ideal EKF with null space $\mathbf{N} = \begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \mathbf{n}_3 \end{bmatrix}$ [see (2.19)]. In what follows, we show that the conditions (2.40)-(2.41) encompass the ones derived in [85] and [79] as special cases.

Stationary robot

We first examine the special case studied in [85], where the robot remains stationary, while observing the relative position of a single landmark. In [85], the following Jacobian constraint

Algorithm 1 Observability-Constrained (OC)-EKFs for SLAM

Require: Initial state estimates and covariance

1: **loop**

2: `Propagation:` If proprioceptive (e.g., odometry) measurements are available,

3: propagate the state estimates [see (2.2), (2.3) and (2.4)]

4: compute the propagation Jacobian [see (2.44) for OC-EKF1, (2.52) for OC-EKF2, and (2.8) for OC-EKF3]

5: propagate the state covariance:

$$\mathbf{P}_{k+1|k} = \mathbf{\Phi}_k \mathbf{P}_{k|k} \mathbf{\Phi}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \tag{2.57}$$

6: `Update:` If exteroceptive measurements are available,

7: compute the measurement residual:

$$\mathbf{r}_{k+1} = \mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}) \tag{2.58}$$

8: compute the measurement Jacobian [see (2.45) for OC-EKF1, (2.53) for OC-EKF2, and (2.56) for OC-EKF3]

9: compute the residual covariance and Kalman gain:

$$\mathbf{S}_{k+1} = \mathbf{H}_{k+1} \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1} \tag{2.59}$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T \mathbf{S}_{k+1}^{-1} \tag{2.60}$$

10: update the state estimate and covariance:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} \mathbf{r}_{k+1} \tag{2.61}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1} \mathbf{S}_{k+1} \mathbf{K}_{k+1}^T \tag{2.62}$$

11: **end loop**

for consistent estimation was derived (see Theorem 1 therein):

$$\begin{aligned} \nabla \mathbf{h}^{\mathbf{x}} - \nabla \mathbf{h}^{\mathbf{p}} \nabla \mathbf{g}^{\mathbf{x}} &= \mathbf{0} \\ \Leftrightarrow \mathbf{H}_{R_k} + \mathbf{H}_{L_k} \nabla \mathbf{g}^{\mathbf{x}} &= \mathbf{0} \\ \Leftrightarrow \begin{bmatrix} \mathbf{H}_{R_k} & \mathbf{H}_{L_k} \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 \\ \nabla \mathbf{g}^{\mathbf{x}} \end{bmatrix} &= \mathbf{0} \\ \Leftrightarrow \mathbf{H}_k \mathbf{N}_s &= \mathbf{0} \end{aligned} \tag{2.63}$$

where, using our notation, $\nabla \mathbf{h}^{\mathbf{x}} = \mathbf{H}_{R_k}$ and $-\nabla \mathbf{h}^{\mathbf{p}} = \mathbf{H}_{L_k}$ are the measurement Jacobian matrices with respect to the robot pose and landmark position, respectively, and $\nabla \mathbf{g}^{\mathbf{x}}$ is the landmark initialization Jacobian with respect to the robot pose at time-step k_o. Note that the condition (2.63) is identical to the one in (2.40) for the special case of a stationary robot.

Remarkably, the space spanned by the columns of the matrix $\mathbf{N}_s$, for this special case, is the same as the one spanned by the columns of $\mathbf{N}$ in (2.42). To see that, we first need to derive an expression for $\nabla \mathbf{g}^{\mathbf{x}}$. In [85], a relative-position measurement model is employed (by combining a distance and a bearing measurement), and thus the initialization function $\mathbf{g}(\cdot)$ is given by:

$$\mathbf{p}_{L_{k_o}} = \mathbf{g}(\mathbf{x}_{R_{k_o}}, \mathbf{z}_{k_o}, \mathbf{v}_{k_o}) = \mathbf{C}(\phi_{R_{k_o}}) \left(\mathbf{z}_{k_o} - \mathbf{v}_{k_o} \right) + \mathbf{p}_{R_{k_o}} \tag{2.64}$$

where $\mathbf{z}_{k_o}$ is the first measurement of the landmark's relative position and $\mathbf{v}_{k_o}$ denotes the noise in this measurement. Evaluating the derivative of this function with respect to the robot pose at the current state estimate we have:

$$\begin{aligned} \nabla \mathbf{g}^{\mathbf{x}} &= \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\mathbf{C}(\hat{\phi}_{R_{k_o|k_o-1}}) \mathbf{z}_{k_o} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_2 & \mathbf{J} \left(\hat{\mathbf{p}}_{L_{k_o|k_o}} - \hat{\mathbf{p}}_{R_{k_o|k_o-1}} \right) \end{bmatrix} \end{aligned} \tag{2.65}$$

where this last equation results from taking conditional expectations on both sides of (2.64) and solving for $\mathbf{z}_{k_o}$.

Substituting (2.65) in the expression for $\mathbf{N}_s$ [see (2.63)], yields:

$$\mathbf{N}_s = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J} \left(\hat{\mathbf{p}}_{L_{k_o|k_o}} - \hat{\mathbf{p}}_{R_{k_o|k_o-1}} \right) \end{bmatrix}$$

One can easily verify that $\mathbf{N}_s$ and $\mathbf{N}$ span the same column space by noting that $\mathbf{N}_s \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{R_{k_o|k_o-1}} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} = \mathbf{N}$.

Moving robot with one-step motion

We now consider the special case studied in [79], where a robot observes a landmark, moves once and then re-observes the landmark. In [79], the key Jacobian relationship that needs to be satisfied in order for consistent estimation in this case (see Theorem 4.2 therein) is given by:

$$A_e = B_e \nabla f^A_{\phi X_r} \tag{2.66}$$

Using our notation, the above matrices are written as:

$$\begin{aligned} \nabla f^A_{\phi X_r} &= \mathbf{\Phi}_{R_{k_o}} \\ A_e &= -\mathbf{H}^{-1}_{L_{k_o}} \mathbf{H}_{R_{k_o}} \\ B_e &= -\mathbf{H}^{-1}_{L_{k_o+1}} \mathbf{H}_{R_{k_o+1}} \end{aligned}$$

Substituting in (2.66) and rearranging terms yields:

$$\begin{aligned} & \mathbf{H}^{-1}_{L_{k_o+1}} \mathbf{H}_{R_{k_o+1}} \mathbf{\Phi}_{R_{k_o}} - \mathbf{H}^{-1}_{L_{k_o}} \mathbf{H}_{R_{k_o}} = \mathbf{0} \\ \Leftrightarrow \begin{bmatrix} \mathbf{H}_{R_{k_o+1}} & \mathbf{H}_{L_{k_o+1}} \end{bmatrix} & \begin{bmatrix} \mathbf{\Phi}_{R_{k_o}} & \mathbf{0}_{3\times 2} \\ \mathbf{0}^T_{3\times 2} & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 \\ -\mathbf{H}^{-1}_{L_{k_o}} \mathbf{H}_{R_{k_o}} \end{bmatrix} = \mathbf{0} \\ & \Leftrightarrow \mathbf{H}_{k_o+1} \mathbf{\Phi}_{k_o} \mathbf{N}_1 = \mathbf{0} \end{aligned}$$

which is the same as the condition in (2.41) for the special case of $\ell = 1$ (i.e., the robot moves only once). Additionally, it is easy to verify that $\mathbf{H}_{k_o}\mathbf{N}_1 = \mathbf{0}$, which corresponds to condition (2.40). Moreover, it is fairly straightforward to show that for the case of distance and bearing measurements considered in [79], the matrix $\mathbf{N}_1$ spans the same column space as $\mathbf{N}$ in (2.42). This analysis demonstrates that the Jacobian constraints (2.40)-(2.41) derived based on the observability criterion are general, and encompass the condition of [79] as a special case.

2.6 Simulation results

A series of Monte-Carlo simulations were conducted under various conditions, in order to validate the preceding theoretical analysis and demonstrate the capability of the proposed OC-EKFs to improve consistency. The metrics used to evaluate filter performance are: (i) the root mean square error (RMSE), and (ii) the average normalized (state) estimation error squared (NEES) [14]. Specifically, for the landmarks we compute the average RMSE and average NEES

by averaging the estimation errors and the NEES, respectively, over all Monte-Carlo runs, all landmarks, and all time steps. On the other hand, for each robot pose we compute these error metrics by averaging over all Monte-Carlo runs for each time step. The RMSE provides us with a concise metric of the accuracy of a given estimator. On the other hand, the NEES is a standard criterion for evaluating filter consistency. Specifically, it is known that the NEES of an N-dimensional Gaussian random variable follows a χ^2 distribution with N d.o.f. Therefore, if a certain filter is consistent, we expect that the average NEES for the robot pose will be close to 3 for all time steps, and that the average landmark NEES will be close to 2. The larger the deviations of the NEES from these values are, the worse the inconsistency of the estimator is. By studying both the RMSE and NEES of all the filters considered here, we obtain a comprehensive picture of the estimators' performance.

In all the simulation tests, a robot with a simple differential drive model moves on a planar surface, at a constant linear velocity of $v = 0.25$ m/sec. The two drive wheels are equipped with encoders that measure revolutions and provide measurements of velocity (i.e., right and left wheel velocities, v_r and v_l, respectively) with standard deviation equal to $\sigma = 2\%v$ for each wheel. These measurements are used to obtain linear and rotational velocity measurements for the robot, which are given by $v = \frac{v_r+v_l}{2}$ and $\omega = \frac{v_r-v_l}{a}$, where $a = 0.5$ m is the distance between the drive wheels. Thus, the standard deviations of the linear and rotational velocity measurements are $\sigma_v = \frac{\sqrt{2}}{2}\sigma$ and $\sigma_\omega = \frac{\sqrt{2}}{a}\sigma$, respectively.

In this SLAM simulation, a robot moves on a circular trajectory and sequentially observes 20 landmarks in total. The robot records distance and bearing measurements to the landmarks that lie within its sensing range of 5 m. The standard deviation of the distance measurement noise is equal to 10% of the true distance, while the standard deviation of the bearing measurement noise is set to 1° deg. Note that the sensor-noise levels selected for this simulation are larger than what is typically encountered in practice. This was done purposefully, since higher noise levels lead to larger estimation errors, which make the effects of inconsistency more apparent. We performed 50 Monte-Carlo simulations and compared six filters: (1) the ideal EKF, (2) the standard EKF, (3) the OC-EKF1, (4) the OC-EKF2, (5) the OC-EKF3, and (6) the robocentric mapping filter [27], which aims at improving the consistency of SLAM by expressing the landmarks in a robot-relative frame. During each run, all filters process the same data, to ensure a fair comparison.

Fig. 2.1 shows the results for the robot orientation estimation errors, obtained from one

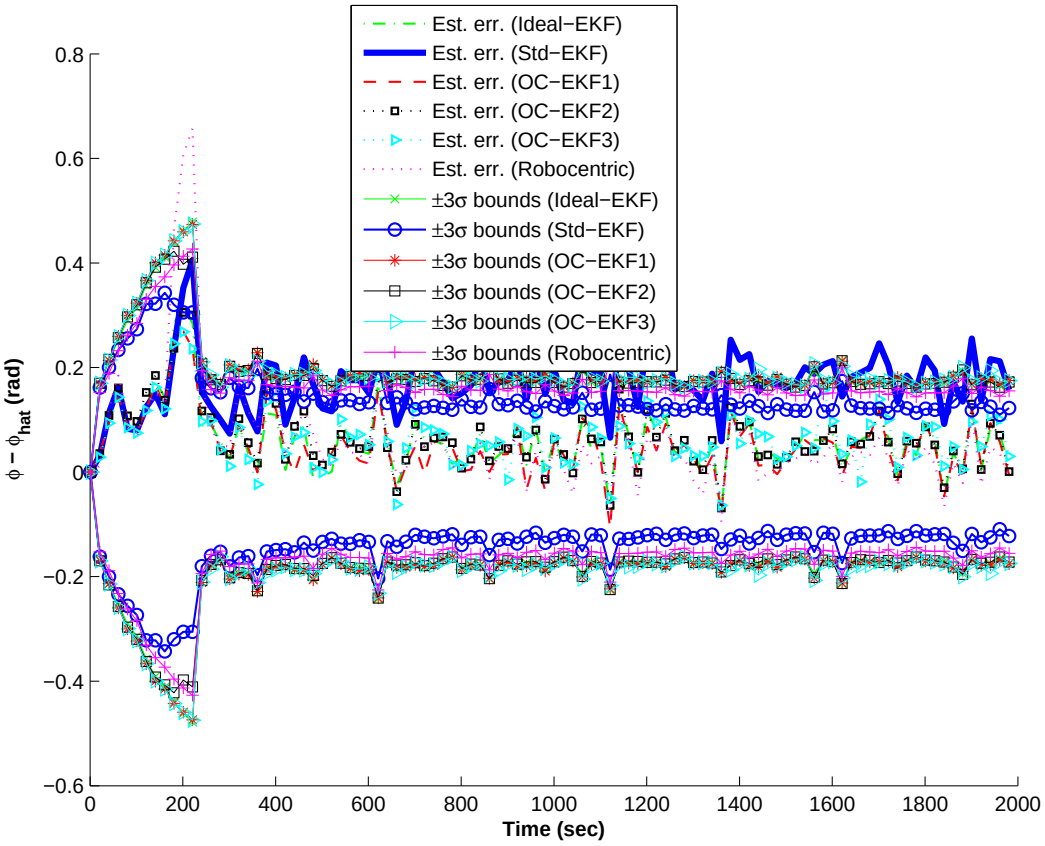
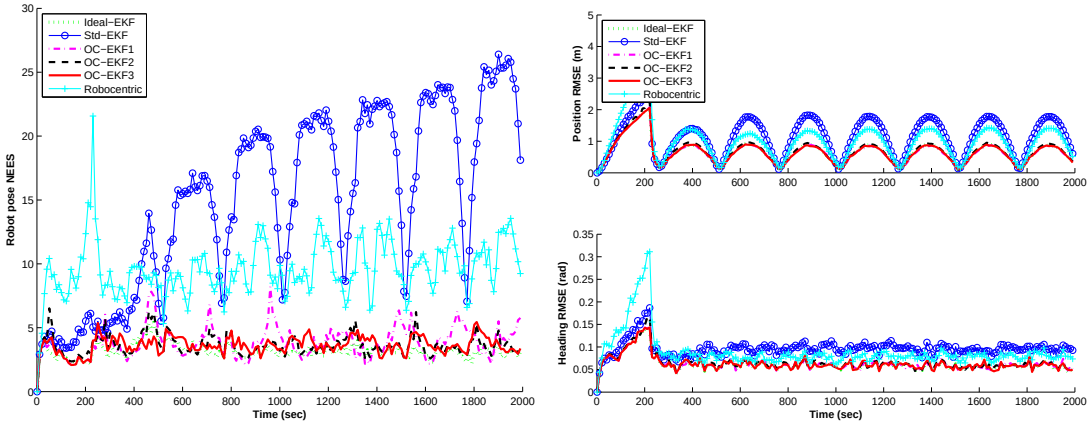


Figure 2.1: Orientation estimation errors vs. 3σ bounds obtained from one typical realization of the SLAM Monte-Carlo simulations. The σ values are computed as the square-root of the corresponding diagonal element of the estimated covariance matrix. Note that the estimation errors and the 3σ bounds of the ideal EKF, the OC-EKFs, and the robocentric mapping filter are almost identical, which makes the corresponding lines difficult to distinguish.

typical simulation of the 50 Monte-Carlo runs. As evident, the errors of the standard EKF grow significantly faster than those of all other filters, which indicates that the standard EKF tends to diverge. Note also that although the orientation errors of the ideal EKF, OC-EKFs, as well as the robocentric mapping filter remain well within their corresponding 3σ bounds (computed from the square-root of the corresponding diagonal element of the estimated covariance matrix), those of the standard EKF exceed them. Most importantly, the 3σ bounds of the standard EKF continuously *decrease* over time, as if the robot's orientation was observable. However, the robot has no access to any absolute orientation information, and thus its orientation covariance



(a) NEES of robot pose

(b) RMSE of robot pose

Figure 2.2: Monte-Carlo simulation results for a SLAM scenario with multiple loop closures. In these plots, the dotted lines correspond to the ideal EKF, the solid lines with circles to the standard EKF, the dash-dotted lines to the OC-EKF1, the dashed lines to the OC-EKF2, the solid lines to the OC-EKF3, and the solid lines with crosses to the robocentric mapping filter. Note that the RMSE of the ideal EKF, and the OC-EKFs are almost identical, which makes the corresponding lines difficult to distinguish.

should *not* continuously decrease. The results of Fig. 2.1 further strengthen our claim that the incorrect observability properties of the standard EKF cause an unjustified reduction in the orientation uncertainty.

The comparative Monte-Carlo results for all filters are presented in Fig. 2.2. Specifically, Fig. 2.2(a) and Fig. 2.2(b) show the average NEES and RMSE for the robot pose, respectively, versus time. On the other hand, Table 2.1 presents the average values of all relevant performance metrics for both the landmarks and the robot. As evident, the performance of the OC-EKFs is *very close* to that of the ideal EKF, and substantially better than that of the standard EKF, both in terms of RMSE and NEES. This occurs even though the Jacobians used in the OC-EKFs are less accurate than those used in the standard EKF, as explained in the preceding section. This fact indicates that the errors introduced by the use of inaccurate Jacobians have a less detrimental effect on consistency and accuracy than the use of an error-state system model with incorrect observability properties. Moreover, it is important to note that the performance of the OC-EKF 2 and 3 is superior to that of the OC-EKF1, by a small margin. This is attributed to the fact that the OC-EKF1 has larger linearization errors than the OC-EKF 2 and 3, since the OC-EKF 2 and 3 are optimal by construction, in terms of linearization errors and hence filter Jacobians,

Table 2.1: SLAM simulation results for robot and landmark estimation performance

Ideal-EKF	Std-EKF	OC-EKF1	OC-EKF2	OC-EKF3	Robocentric
Robot Position RMSE (m)					
0.6556	1.1416	0.6440	0.6853	0.6465	1.0315
Robot Heading RMSE (rad)					
0.0627	0.0964	0.0639	0.0657	0.0636	0.0895
Robot Pose NEES					
3.1926	14.9305	3.8802	3.6282	3.6386	9.5894
Landmark Position RMSE (m)					
0.6558	1.1895	0.7041	0.6791	0.6563	1.0532
Landmark Position NEES					
2.2420	18.8000	5.6793	3.0222	2.8011	10.6310

under the observability constraints. We also observe that the OC-EKFs perform better than the robocentric mapping filter [26, 27], both in terms of accuracy and consistency, which is explained in detail in the next.

2.6.1 Comparison to the robocentric mapping filter

From the plots of Fig. 2.2, we clearly see that the OC-EKFs also perform better than the robocentric mapping filter [26, 27], both in terms of accuracy and consistency. This result cannot be justified based on the observability properties of the filters: in [26, 27], the landmarks are represented in the robot frame, which can be shown to result in a system model with 3 unobservable degrees of freedom (see Appendix A.6). However, in the robocentric mapping filter, during each propagation step *all* landmark position estimates need to be changed, since they are expressed with respect to the moving robot frame. As a result, during each propagation step (termed *composition* in [26, 27]), all landmark estimates and their covariance are affected by the linearization errors of the process model. This problem does not exist in the world-centric formulation of SLAM, and it could offer an explanation for the observed behavior.

To test this argument, we first examine the Kullback-Leibler divergence (KLD), between the pdf estimated by each filter, and the pdf estimated by its "ideal" counterpart. Specifically, we

compute the KLD (i) between the pdf computed by the OC-EKF [4] and that of the ideal EKF, and (ii) between the pdf computed by the robocentric mapping filter and that produced by an "ideal" robocentric mapping filter, which employs the true states in computing all the Jacobian matrices. The KLD is a standard measure for the difference between probability distributions. It is nonnegative, and equals zero only if the two distributions are identical [32]. By computing the KLD between the estimated pdf and that of the "ideal" filter in each case, we can evaluate how close each filter is to its respective "golden standard". These results pertain to the same simulation setup presented in the previous section.

Since the four filters considered here (i.e., the OC-EKF, the ideal EKF, the robocentric mapping filter, and the ideal robocentric mapping filter) employ a Gaussian approximation of the pdf, we can compute the KLD in closed form. Specifically, the KLD from an approximation distribution, $p_a(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{P}_a)$, to the ideal distribution, $p_o(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_o, \mathbf{P}_o)$, is given by:

$$d_{KL} = \frac{1}{2}\left(\ln\left(\frac{\det(\mathbf{P}_o)}{\det(\mathbf{P}_a)}\right) + \mathrm{tr}(\mathbf{P}_o^{-1}\mathbf{P}_a) + (\boldsymbol{\mu}_o - \boldsymbol{\mu}_a)^T\mathbf{P}_o^{-1}(\boldsymbol{\mu}_o - \boldsymbol{\mu}_a) - \dim(\mathbf{x})\right) \quad (2.67)$$

Fig. 2.3 presents the KLD over time, between the Gaussian distributions computed by the robocentric mapping filter, the OC-EKF, and those computed by their respective ideal filters (note that the vertical axis scale is logarithmic). It is evident that the KLD in the case of the robocentric mapping filter is orders of magnitude larger than in the case of the OC-EKF. This indicates that the linearization errors in the robocentric mapping filter result in a worse approximation of the ideal pdf.

We attribute this fact to the structure of the filter Jacobians. During the update step, the structure of the Jacobians in both the robocentric and the world-centric formulations is quite similar [72]. In both cases, the terms appearing in the measurement Jacobians are either rotation matrices, or the robot-to-landmark position vector. However, the Jacobians employed during the composition step in the robocentric mapping filter are substantially more complex than those appearing in the world-centric EKF propagation [see (2.6)]. Specifically, in the robocentric mapping filter, the state vector is given by (assuming a single landmark for simplicity):

$${}^{R_k}\mathbf{x}_k = \begin{bmatrix} {}^{R_k}\mathbf{p}_{G_k}^T & {}^{R_k}\phi_{G_k} & {}^{R_k}\mathbf{p}_{L_k}^T \end{bmatrix}^T \quad (2.68)$$

[4] Due to the similar performance of the three OC-EKFs (see Fig. 2.2 and Table 2.1), we here only compare the OC-EKF2 to the robcentric mapping filter.

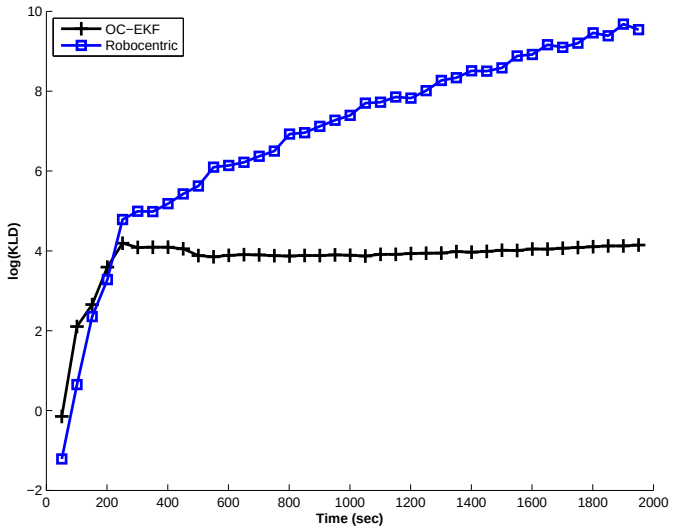


Figure 2.3: Comparison results of the KLD in the SLAM scenario with multiple loop closures. In this plot, the solid line with crosses corresponds to the OC-EKF, and the solid line with squares to the robocentric mapping filter [27]. Note that the vertical axis scale is logarithmic.

The composition step is described by the following equations:

$$
\begin{aligned}
{}^{R_k}\hat{\mathbf{p}}_{G_{k|k-1}} &= \mathbf{C}^T({}^{R_{k-1}}\hat{\phi}_{R_{k|k-1}})({}^{R_{k-1}}\hat{\mathbf{p}}_{G_{k-1|k-1}} - {}^{R_{k-1}}\hat{\mathbf{p}}_{R_{k|k-1}}) && (2.69)\\
{}^{R_k}\hat{\phi}_{G_{k|k-1}} &= {}^{R_{k-1}}\hat{\phi}_{G_{k-1|k-1}} - {}^{R_{k-1}}\hat{\phi}_{R_{k|k-1}} && (2.70)\\
{}^{R_k}\hat{\mathbf{p}}_{L_{k|k-1}} &= \mathbf{C}^T({}^{R_{k-1}}\hat{\phi}_{R_{k|k-1}})({}^{R_{k-1}}\hat{\mathbf{p}}_{L_{k-1|k-1}} - {}^{R_{k-1}}\hat{\mathbf{p}}_{R_{k|k-1}}) && (2.71)
\end{aligned}
$$

where ${}^{R_\ell}\hat{\mathbf{p}}_{L_{\ell|k-1}}$ is the estimated landmark position with respect to the robot frame at time-step ℓ ($\ell = k-1, k$), $\{{}^{R_{k-1}}\hat{\mathbf{p}}_{R_{k|k-1}}, {}^{R_{k-1}}\hat{\phi}_{R_{k|k-1}}\}$ is the estimate of the robot-pose change between time-steps $k-1$ and k, expressed with respect to the robot frame at time-step $k-1$, and $\{{}^{R_\ell}\hat{\mathbf{p}}_{G_{\ell|k-1}}, {}^{R_\ell}\hat{\phi}_{G_{\ell|k-1}}\}$ is the estimated transformation between the robot frame and the global frame at time-step ℓ. The linearized error propagation equation is given by:

$$
\begin{bmatrix} {}^{R_k}\widetilde{\mathbf{p}}_{G_{k|k-1}} \\ {}^{R_k}\widetilde{\phi}_{G_{k|k-1}} \\ {}^{R_k}\widetilde{\mathbf{p}}_{L_{k|k-1}} \end{bmatrix} = \mathbf{J}_{L_k}\,{}^{R_{k-1}}\widetilde{\mathbf{p}}_{L_{k-1|k-1}} + \mathbf{J}_{G_k}\begin{bmatrix} {}^{R_{k-1}}\widetilde{\mathbf{p}}_{G_{k-1|k-1}} \\ {}^{R_{k-1}}\widetilde{\phi}_{G_{k-1|k-1}} \end{bmatrix} + \mathbf{J}_{R_k}\begin{bmatrix} {}^{R_{k-1}}\widetilde{\mathbf{p}}_{R_{k|k-1}} \\ {}^{R_{k-1}}\widetilde{\phi}_{R_{k|k-1}} \end{bmatrix} \quad (2.72)
$$

where

$$\mathbf{J}_{L_k} = \begin{bmatrix} \mathbf{0}_{3\times 2} \\ \mathbf{C}^T\big({}^{R_{k-1}}\hat{\phi}_{R_{k|k-1}}\big) \end{bmatrix} \tag{2.73}$$

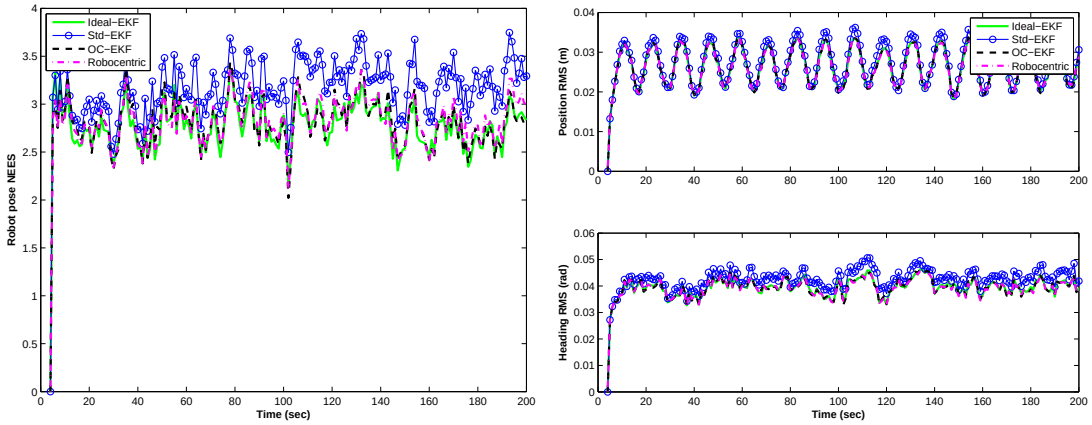
$$\mathbf{J}_{G_k} = \begin{bmatrix} \mathbf{C}^T\big({}^{R_{k-1}}\hat{\phi}_{R_{k|k-1}}\big) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \end{bmatrix} \tag{2.74}$$

$$\mathbf{J}_{R_k} = \begin{bmatrix} -\mathbf{C}^T\big({}^{R_{k-1}}\hat{\phi}_{R_{k|k-1}}\big) & -\mathbf{J}\,{}^{R_k}\hat{\mathbf{p}}_{G_{k|k-1}} \\ \mathbf{0}_{1\times 2} & -1 \\ -\mathbf{C}^T\big({}^{R_{k-1}}\hat{\phi}_{R_{k|k-1}}\big) & -\mathbf{J}\,{}^{R_k}\hat{\mathbf{p}}_{L_{k|k-1}} \end{bmatrix} \tag{2.75}$$

We note that the state estimates appear in the Jacobian matrices $\mathbf{J}_{L_k}$ and $\mathbf{J}_{G_k}$ only through the rotation matrix $\mathbf{C}({}^{R_{k-1}}\hat{\phi}_{R_{k|k-1}})$. As a result, the difference between the ideal and actual Jacobians, $\mathbf{J}_{L_k} - \breve{\mathbf{J}}_{L_k}$ and $\mathbf{J}_{G_k} - \breve{\mathbf{J}}_{G_k}$ will only contain terms of the form $c({}^{R_{k-1}}\hat{\phi}_{R_{k|k-1}}) - c({}^{R_{k-1}}\phi_{R_k})$, and $s({}^{R_{k-1}}\hat{\phi}_{R_{k|k-1}}) - s({}^{R_{k-1}}\phi_{R_k})$. The magnitude of these terms is in the same order as ${}^{R_{k-1}}\widetilde{\phi}_{R_k}$, which is typically a very small quantity. Thus, the discrepancy between the actual and ideal Jacobians is expected to be very small for $\mathbf{J}_{L_k}$ and $\mathbf{J}_{G_k}$.

On the other hand, in $\mathbf{J}_{R_k}$ the estimates for the landmark position and for the origin of the global frame with respect to the robot appear as well. As a result, the difference $\mathbf{J}_{R_k} - \breve{\mathbf{J}}_{R_k}$ will also contain the terms ${}^{R_k}\widetilde{\mathbf{p}}_{G_{k|k-1}}$ and ${}^{R_k}\widetilde{\mathbf{p}}_{L_{k|k-1}}$, whose magnitude can be significantly larger, e.g., in the order of meters (see Fig. 2.2). Thus, the Jacobian $\mathbf{J}_{R_k}$ can be very inaccurate. In contrast, the propagation Jacobians in the world-centric formulation contain terms depending on (i) the robot's displacement between consecutive time steps, and (ii) the rotation matrix of the robot's orientation [see (2.8) and (2.9)]. Since both of these quantities can be estimated with small errors, the world-centric EKF Jacobians are significantly more accurate than those of the robocentric formulation.

To further test this argument, we ran a simulation of a "mini-SLAM" scenario, where both the robot trajectory and the landmarks are confined within a small area of 1 m$\times$1 m (while all other settings are identical to the preceding simulation). In this setup, the estimation errors ${}^{R_k}\widetilde{\mathbf{p}}_{G_{k|k-1}}$ and ${}^{R_k}\widetilde{\mathbf{p}}_{L_{k|k-1}}$ remain small, and thus the Jacobians of the robocentric mapping filter become more accurate. The plots of Fig. 2.4 show the average NEES and RMSE for the robot pose in this scenario. Interestingly, we observe that in this case the performance of the OC-EKF, and the robocentric mapping filter are *almost identical*. This validates the preceding



(a) NEES of robot pose (b) RMSE of robot pose

Figure 2.4: Monte-Carlo results for a "mini-SLAM" scenario with multiple loop closures where the robot trajectory and all landmarks are confined within a very small area of 1 m $\times$ 1 m. In these plots, the solid lines correspond to the ideal EKF, the solid lines with circles to the standard EKF, the dashed lines to the OC-EKF, and the dash-dotted lines to the robocentric mapping filter of [27]. Note that in this case both the NEES and the RMSE of the ideal EKF, the OC-EKF, and the robocentric mapping filter are almost identical, which makes the corresponding lines difficult to distinguish.

discussion, and indicates that the representation used in the robocentric mapping filter results in performance loss in the case of large environments. This may justify the fact that the OC-EKF outperforms the robocentric mapping algorithm [27], even though both filters employ a system model with three unobservable d.o.f.

As a final remark, we note that, in comparison to the OC-EKF, the computational cost of the robocentric mapping filter is significantly higher. Specifically, the OC-EKF has computational cost identical to the standard world-centric SLAM algorithm: *linear* in the number of landmarks during propagation, and *quadratic* during updates. On the other hand, both the update and the composition steps in the robocentric mapping filter have computational cost *quadratic* in the number of features, which results in approximately double overall computational burden.

2.7 Experimental results

We also performed a real-world experiment to further test the proposed OC-EKF algorithms. This experiment was conducted in an indoor office environment. The robot was commanded

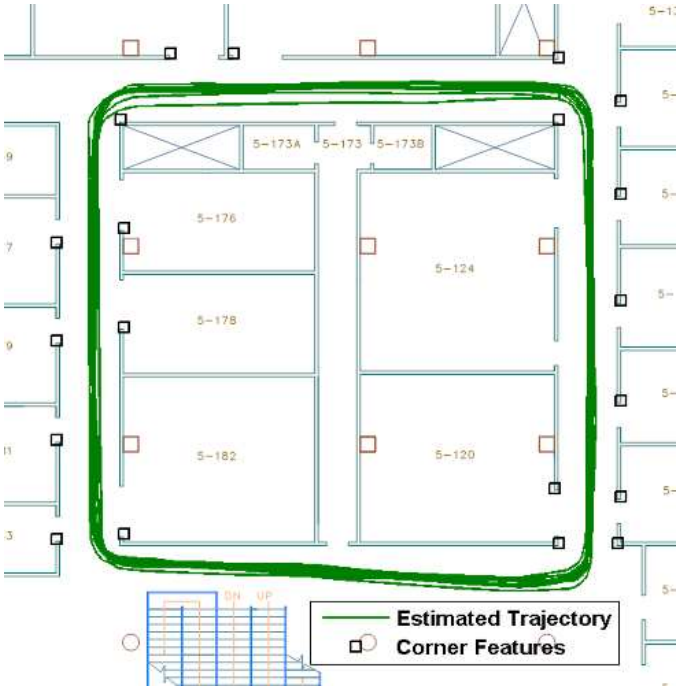


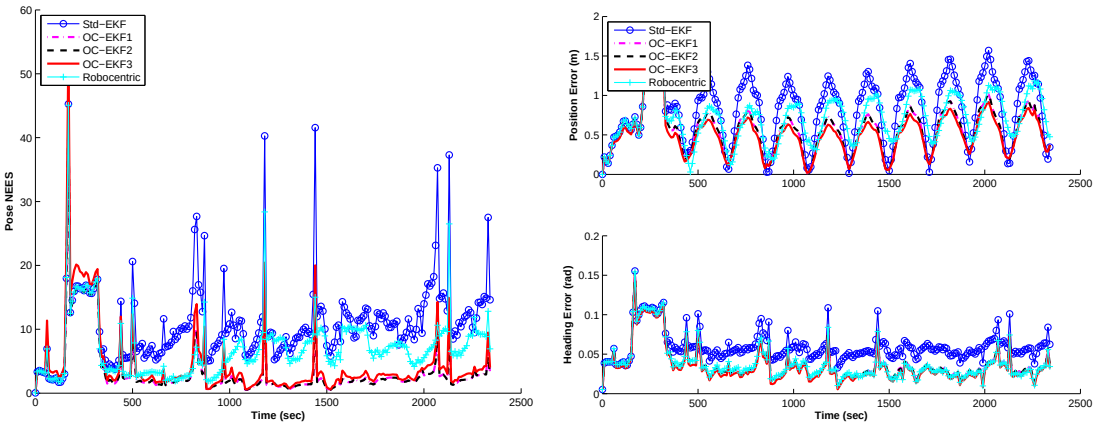
Figure 2.5: The batch maximum-a-posteriori (MAP) estimate of the robot trajectory in the indoor SLAM experiment (solid line), overlaid on the blueprint of the building. The boxes (□) denote the corners whose exact location is known from the building's blueprint. The batch-MAP estimates of the robot poses and the known corners were used as ground truth for computing the NEES and RMSE values.

to perform 11 loops around a square with sides approximately equal to 20 m (see Fig. 4.3). This special trajectory was selected since repeated re-observation of the same landmarks tends to make the effects of inconsistency more apparent, and facilitates discerning the performance of the various filters. A Pioneer robot equipped with a SICK LMS200 laser range-finder and wheel encoders was used in this experiment. From the laser range data, corner features were extracted and used as landmarks, while the wheel encoders provided the linear and rotational velocity measurements. Propagation was carried out using the kinematic model described in Appendix A.1.

Because the ground truth of the robot pose could not be obtained using external sensors (e.g., overhead cameras), in this experiment, we obtained a reference trajectory by utilizing the known map of the area where the experiment took place. Specifically, the exact location of 20 corners was known from the blueprints of the building. Measurements to these corners, as

Table 2.2: SLAM experimental results for robot and landmark estimation performance

Std-EKF	OC-EKF1	OC-EKF2	OC-EKF3	Robocentric
Robot Position RMSE (m)				
0.8209	0.5748	0.5754	0.5214	0.7160
Robot Heading RMSE (rad)				
0.0604	0.0397	0.0397	0.0356	0.0391
Robot Pose NEES				
11.0706	3.5681	3.5282	4.6127	7.2949
Landmark Position RMSE (m)				
1.1041	0.8675	0.8680	0.8474	1.0957
Landmark Position NEES				
8.5033	5.9821	5.9836	6.8402	9.6691



(a) Robot NEES (b) Robot estimation errors

Figure 2.6: SLAM experimental results. In these plots, the solid lines correspond to the standard EKF, the dash-dotted lines to the OC-EKF1, the dashed lines to the OC-EKF2, the solid lines to the OC-EKF3, and the solid lines with crosses to the robocentric mapping filter. Note that some portions of the NEES and the estimation errors of the OC-EKFs are very close, which makes the corresponding lines difficult to distinguish.

well as all other measurements obtained by the robot (including to corners whose location was not known *a priori*), were processed using a batch-MAP estimator [91], to obtain an accurate

estimate of the entire trajectory. This estimate, as well as the locations of the known corners, are shown in Fig. 4.3. This constitutes the ground truth against which the performance of the following filters was compared: (1) the standard EKF, (2) the OC-EKF1, (3) the OC-EKF2, (4) the OC-EKF3, and (4) the robocentric mapping filter [27]. Clearly, due to the way the ground truth is computed, the filter errors are expected to have some correlation to the errors in the ground truth. However, since these correlations are the same for all four filters, we can still have a fair comparison of their relative performance.

The results of NEES and estimation errors (RMSE) for all filters are presented in Fig. 2.6 and Table 2.2. We point out that during the experiment the robot detected a number of features that were not included in the set of 20 known corners (e.g., movable objects such as furniture). Since no ground truth was available for the position of these objects, we only used the 20 known corners for computing the landmarks' error statistics. From the experimental results it becomes clear that in this particular experiment the OC-EKFs outperform the standard EKF and the robocentric mapping filter, and perform almost identically to each other. This agrees with the simulation results presented in the preceding section.

2.8 Summary

In this chapter, we have presented an observability-based study of the inconsistency problem in EKF-based SLAM. By comparing the observability properties of the nonlinear SLAM system with those of the linearized error-state model employed in the EKF, we proved that the observable subspace of the standard EKF is always of higher dimension than that of the underlying nonlinear system. As a result, the covariance estimates of the EKF undergo reduction in directions of the state space where no information is available, which is a primary cause of inconsistency. Based on this analysis, we have proposed a new methodology for the design of EKF-based estimators for SLAM. Our approach dictates computing the EKF Jacobians, either indirectly or directly, so as to ensure that the resulting linearized system model has three unobservable directions.

We have introduced three OC-EKF algorithms, which adhere to the above design methodology. Specifically, the OC-EKF1 computes the Jacobians using the first-available estimate for each state variable, while in the OC-EKF2 the linearization points used for computing the Jacobians are obtained in closed form by solving an observability-constrained minimization problem

(i.e., minimizing the expected linearization errors subject to the observability constraints). Alternatively, the OC-EKF3 directly computes the measurement Jacobian by projecting the most accurate measurement Jacobian onto the observable subspace, while the propagation Jacobian is calculated in the same way as in the standard EKF. As a result, the linearized system models employed in these filters have the desirable observability properties. Extensive simulation and experimental tests verify that the OC-EKFs perform significantly better, in terms of both accuracy and consistency, than the standard EKF and the robocentric mapping filter [27]. This occurs despite the fact that the Jacobians used in the OC-EKFs are less accurate. These results indicate that ensuring the correct observability properties of the linearized system model is a crucial requirement.

Chapter 3

Observability-Constrained EKFs for CL

In this chapter, we extend the observability-based methodology for SLAM presented in the previous chapter to the case of CL. In particular, we analytically show that the error-state system model employed in the standard EKF-based CL always has an observable subspace of higher dimension than that of the actual nonlinear CL system. This results in unjustified reduction of the EKF's estimated covariance in directions of the state space where no information is available, and thus leads to inconsistency. To address this problem, we adapt the previously presented observability-constrained methodology for designing consistent estimators to compute the filter Jacobians and ensure that the linearized CL system model has observable subspace of the correct dimensions. Parts of this chapter have been published in [74, 75].

3.1 Introduction

In order for multi-robot teams to navigate autonomously and successfully perform tasks such as exploration [80], surveillance [154], and search and rescue [83], they must be able to determine their positions and orientations (poses) precisely. In GPS-denied areas and in the absence of robust landmarks, a team of robots can still localize by sharing relative robot-to-robot measurements and jointly estimating their poses [101, 133, 136]. Current approaches to solving the cooperative localization (CL) problem, in either centralized or distributed fashion, are based on the EKF [136], MLE [64], MAP estimator [124], or PF [48]. Among these algorithms, the EKF

arguably remains a popular choice primarily due to its relatively low computational cost and its ease of implementation.

While recent research efforts have primarily focused on reducing the computational complexity of EKF-based CL [90, 108, 117, 127], the fundamental issue of *consistency* has received little attention. As we know, a state estimator is consistent if the estimation errors are zero-mean, and have covariance smaller than or equal to the one calculated by the estimator [14]. Consistency is one of the primary criteria for evaluating the performance of any estimator; if an estimator is inconsistent, then the accuracy of the produced state estimates is unknown, which renders the estimator unreliable. Clearly the lack of understanding consistency in CL is a significant limitation, and hence in this chapter, we study in depth the consistency of EKF-CL. Specifically, based on the system observability analysis, we identify a major cause of the inconsistency of standard EKF-CL and introduce new OC-EKF CL algorithms that significantly improve consistency as well as accuracy. In particular, the major contributions of this work are the following:

- We investigate the observability properties of the error-state system model employed by the EKF, and show that its observable subspace has *higher dimension* than that of the underlying nonlinear CL system. As a result, the estimated covariance of the EKF undergoes reduction in directions of the state space where no information is available, hence leading to *inconsistency*. To the best of our knowledge, we are the first to identify and report this inconsistency of standard EKF-CL.

- Based on the observability analysis, we introduce three new OC-EKFs. These estimators judiciously compute the EKF Jacobians to ensure that the linearized CL system has an observable subspace of the *same dimensions* as that of the nonlinear CL system, thus improving consistency. Specifically, in the OC-EKF1, the state-propagation Jacobians are evaluated at the *prior* state estimates (i.e., *before* instead of after each update), while the measurement Jacobians are computed in the same way as for the standard EKF. In the OC-EKF2, the linearization points are selected so as not only to guarantee the desired observability properties, but also to minimize the expected linearization errors (i.e., the difference between the linearization point and the true state). This is formulated as a constrained minimization problem, whose solution provides the linearization points used for computing the filter Jacobians. In the OC-EKF3, the measurement Jacobians are

directly computed by projecting the best-available measurement Jacobians (calculated using the latest, and thus best, state estimates as in the standard EKF) onto the observable directions, while the state-propagation Jacobians are computed in the same way as for the standard EKF.

- Through extensive Monte-Carlo simulations and real-world experiments with both *homogeneous* and *heterogeneous* robot teams, we verify that the OC-EKFs substantially outperform the standard EKF in terms of consistency and accuracy, even though they use less accurate state estimates to compute the filter Jacobians (since the OC-EKF Jacobians are, in general, different from those computed using the latest, and thus best, state estimates). This result in turn indicates that the observability properties of the system model employed by the filer play a key role in determining the filter's consistency.

3.2 Related work

To date, theoretical studies on the properties of CL have focused on issues such as initialization [160–162, 173], system observability [109, 161, 173], accuracy bounds [117, 137], and the complexity of deterministic (static) robot network localization [37]. However, to the best of our knowledge, prior to our work [74, 75], no work has *analytically* examined the consistency of CL. In contrast, recent research has focused on the consistency of EKF-SLAM [12, 27, 66, 70, 72, 78, 79, 85] showing that the computed state estimates tend to be inconsistent (also see Section 2.2).

In the previous chapter, we conducted a theoretical analysis of the EKF-SLAM inconsistency, and identified as a fundamental cause the mismatch between the dimensions of the observable subspaces of the linearized system employed by the EKF, and the underlying nonlinear system. Furthermore, we introduced the OC-EKFs which significantly outperform the standard EKF and the robocentric mapping algorithm [27], in terms of both accuracy and consistency. The proposed estimators were derived by imposing the constraints inferred from the system observability analysis. In this work, we extend this observability-based methodology for designing consistent estimators for nonlinear systems to address the inconsistency of EKF-CL.

We note that the work of [9] addresses a related but different problem, namely the consistency of a distributed CL algorithm due to reuse of information. In the decentralized estimation

scheme of [9], the cross-correlations between the state estimates of different robots are not estimated. However, it is well-known that if cross-correlations between robots are not properly taken into account during filter updates, inconsistency can arise [48, 65, 136]. The algorithm in [9] avoids inconsistency by maintaining a careful record of past robot-to-robot measurement updates. In contrast to the above fully decentralized scenario, in our work the cross-correlation terms are maintained in the filter, and the EKF employed for estimation is optimal, except for the inaccuracies introduced by linearization. Our work focuses on identifying and addressing the cause of inconsistency of the EKF-CL estimator.

3.3 Standard EKF-CL formulation

In this section, we present the equations of the 2D EKF-CL formulation with *general* system and measurement models.[1] In the standard formulation of CL, the state vector comprises the N robots' poses expressed in the global frame of reference. Thus, at time-step k the state vector is given by:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{1_k}^T & \dots & \mathbf{x}_{N_k}^T \end{bmatrix}^T \tag{3.1}$$

where $\mathbf{x}_{i_k} \triangleq [\mathbf{p}_{i_k}^T \ \ \phi_{i_k}]^T \triangleq [x_{i_k} \ \ y_{i_k} \ \ \phi_{i_k}]^T$ denotes the ith robot pose (position and orientation). In general, EKF-CL recursively evolves in two steps: propagation and update, based on the discrete-time process and measurement models, respectively.

3.3.1 EKF propagation

During propagation, each robot integrates its odometry measurements to obtain an estimate of its pose change between two consecutive time steps, which is then employed in the EKF to propagate the robot state estimate. The EKF propagation equations are given by:

$$\hat{\mathbf{p}}_{i_{k+1|k}} = \hat{\mathbf{p}}_{i_{k|k}} + \mathbf{C}(\hat{\phi}_{i_{k|k}}) \, {}^k\hat{\mathbf{p}}_{i_{k+1}} \tag{3.2}$$

$$\hat{\phi}_{i_{k+1|k}} = \hat{\phi}_{i_{k|k}} + {}^k\hat{\phi}_{i_{k+1}} \tag{3.3}$$

for all $i = 1, \dots, N$. In the above expressions, $\mathbf{C}(\cdot)$ denotes the 2×2 rotation matrix, and ${}^k\hat{\mathbf{x}}_{i_{k+1}} \triangleq [{}^k\hat{\mathbf{p}}_{i_{k+1}}^T \ \ {}^k\hat{\phi}_{i_{k+1}}]^T$ is the odometry-based estimate of the ith robot's motion between

[1] For the purpose of the consistency study and in order to simplify the derivations, in this work we focus on the centralized EKF-CL. Note that a distributed implementation [136] does not alter the system properties.

time-steps k and $k+1$, expressed with respect to the robot frame of reference at time instant k. This estimate is corrupted by zero-mean white Gaussian noise $\mathbf{w}_{i_k} = {}^k\mathbf{x}_{i_{k+1}} - {}^k\hat{\mathbf{x}}_{i_{k+1}}$, with covariance matrix $\mathbf{Q}_k$. Clearly the process model is nonlinear, and can be described by the following generic nonlinear function:

$$\mathbf{x}_{i_{k+1}} = \mathbf{f}(\mathbf{x}_{i_k}, {}^k\hat{\mathbf{x}}_{i_{k+1}} + \mathbf{w}_{i_k}) \tag{3.4}$$

Linearization of (3.4) yields the error-state propagation equation:

$$\tilde{\mathbf{x}}_{i_{k+1|k}} \simeq \mathbf{\Phi}_{i_k}\tilde{\mathbf{x}}_{i_{k|k}} + \mathbf{G}_{i_k}\mathbf{w}_{i_k} \tag{3.5}$$

where $\mathbf{\Phi}_{i_k}$ and $\mathbf{G}_{i_k}$ are the system state and noise Jacobians, respectively, given by:

$$\mathbf{\Phi}_{i_k} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\mathbf{C}(\hat{\phi}_{i_{k|k}})\,{}^k\hat{\mathbf{p}}_{i_{k+1}} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{i_{k+1|k}} - \hat{\mathbf{p}}_{i_{k|k}}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.6}$$

$$\mathbf{G}_{i_k} = \begin{bmatrix} \mathbf{C}(\hat{\phi}_{i_{k|k}}) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.7}$$

By stacking all N robots' error states to create the error state vector for the entire system, we have:

$$\begin{aligned} \tilde{\mathbf{x}}_{k+1|k} &\simeq \begin{bmatrix} \mathbf{\Phi}_{1_k} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{\Phi}_{N_k} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{1_{k|k}} \\ \vdots \\ \tilde{\mathbf{x}}_{N_{k|k}} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{1_k} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{G}_{N_k} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{1_k} \\ \vdots \\ \mathbf{w}_{N_k} \end{bmatrix} \\ &\triangleq \mathbf{\Phi}_k\tilde{\mathbf{x}}_{k|k} + \mathbf{G}_k\mathbf{w}_k \end{aligned} \tag{3.8}$$

Note that the form of the propagation equations presented above is general, and holds for any robot kinematic model (e.g., unicycle, bicycle, or Ackerman model). The specialization to the common case of a unicycle model can be found in Appendix B.4 or A.1.

3.3.2 EKF update

The measurements used for updates in CL are always a function of the relative pose (i.e., relative position and orientation) of the observed robot j with respect to the observing robot i, and are given by:

$$\mathbf{z}_k^{(ij)} = \mathbf{h}(\mathbf{x}_{i_k}, \mathbf{x}_{j_k}) + \mathbf{v}_k^{(ij)} = \mathbf{h}\left({}^i\mathbf{x}_{j_k}\right) + \mathbf{v}_k^{(ij)} \tag{3.9}$$

where

$$ {}^i\mathbf{x}_{j_k} = \begin{bmatrix} {}^i\mathbf{p}_{j_k} \\ {}^i\phi_{j_k} \end{bmatrix} = \begin{bmatrix} \mathbf{C}^T(\phi_{i_k})(\mathbf{p}_{j_k} - \mathbf{p}_{i_k}) \\ \phi_{j_k} - \phi_{i_k} \end{bmatrix} \tag{3.10} $$

is the relative pose of the observed robot j with respect to the observing robot i at time-step k, and $\mathbf{v}_k^{(ij)}$ is zero-mean Gaussian noise with covariance $\mathbf{R}_k^{(ij)}$. In this work, we allow $\mathbf{h}$ to be *any* measurement function. For instance, $\mathbf{z}_k^{(ij)}$ can be a direct measurement of relative pose, a pair of distance and bearing measurements, bearing-only measurements from monocular cameras, etc. In general, the measurement function is nonlinear, and hence it is linearized for use in the EKF. The linearized measurement-error equation is given by:

$$ \begin{aligned} \tilde{\mathbf{z}}_k^{(ij)} &\simeq \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{H}_{i_k}^{(ij)} & \cdots & \mathbf{H}_{j_k}^{(ij)} & \cdots & \mathbf{0} \end{bmatrix} \tilde{\mathbf{x}}_{k|k-1} + \mathbf{v}_k^{(ij)} \\ &\triangleq \mathbf{H}_k^{(ij)} \tilde{\mathbf{x}}_{k|k-1} + \mathbf{v}_k^{(ij)} \end{aligned} \tag{3.11} $$

where $\mathbf{H}_{i_k}^{(ij)}$ and $\mathbf{H}_{j_k}^{(ij)}$ are the Jacobians of $\mathbf{h}$ with respect to the ith and jth robot poses, respectively, evaluated at the state estimate $\hat{\mathbf{x}}_{k|k-1}$. Using the chain rule of differentiation, these are computed as:

$$ \mathbf{H}_{i_k}^{(ij)} = -(\nabla \mathbf{h}_k^{(ij)}) \mathbf{A}(\hat{\phi}_{i_{k|k-1}}) \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{j_{k|k-1}} - \hat{\mathbf{p}}_{i_{k|k-1}}) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.12} $$

$$ \mathbf{H}_{j_k}^{(ij)} = (\nabla \mathbf{h}_k^{(ij)}) \mathbf{A}(\hat{\phi}_{i_{k|k-1}}) \tag{3.13} $$

where $\mathbf{A}(\hat{\phi}_{i_{k|k-1}}) \triangleq \begin{bmatrix} \mathbf{C}^T(\hat{\phi}_{i_{k|k-1}}) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$, and $\nabla \mathbf{h}_k^{(ij)}$ denotes the Jacobian of $\mathbf{h}$ with respect to the relative pose between the ith and jth robots (i.e., with respect to the vector ${}^i\mathbf{x}_{j_k}$), evaluated at the state estimate $\hat{\mathbf{x}}_{k|k-1}$. Appendix B.4 illustrates the specific form of the above expressions in the case of distance and bearing measurements.

3.4 CL observability analysis

In this section, we perform an observability analysis for the EKF-CL system derived in the previous section, and compare its observability properties with those of the underlying nonlinear system. Based on this analysis, we draw conclusions about the consistency of the filter.

By applying the *observability rank condition* for nonlinear systems [60], Martinelli and Siegwart [109] have shown that the nonlinear system of CL in general has three unobservable

d.o.f., corresponding to the global position and orientation. However, as we show in this section, the unobservable subspace of the linearized error-state model of the standard EKF is generally only of dimension two, which leads to inconsistency.[2]

Recall that, in analogy to (2.21), the Jacobian matrices $\mathbf{\Phi}_k$, $\mathbf{G}_k$, and $\mathbf{H}_k$ used in the EKF-CL linearized error-state model [see (3.8) and (3.11)], in general, are defined as:

$$\mathbf{\Phi}_k = \nabla_{\mathbf{x}_k}\mathbf{f}\Big|_{\{\mathbf{x}^\star_{k|k},\mathbf{x}^\star_{k+1|k},\mathbf{0}\}},\ \ \mathbf{G}_k = \nabla_{\mathbf{w}_k}\mathbf{f}\Big|_{\{\mathbf{x}^\star_{k|k},\mathbf{0}\}},\ \ \mathbf{H}_k = \nabla_{\mathbf{x}_k}\mathbf{h}\Big|_{\{\mathbf{x}^\star_{k|k-1}\}} \tag{3.14}$$

In these expressions, $\mathbf{x}^\star_{k|k-1}$ and $\mathbf{x}^\star_{k|k}$ denote the *linearization points* for the state $\mathbf{x}_k$, used for evaluating the Jacobians before and after the EKF update at time-step k, respectively. A linearization point equal to the zero vector is chosen for the noise. The EKF employs the linearized system model defined by (3.8), (3.11), and (3.14) for propagating and updating the state and covariance estimates, and thus the observability properties of this model affect the performance of the estimator.

Since the linearized error-state model of EKF-CL is time-varying, similarly to the case of EKF-SLAM, we employ the *local observability matrix* [30, 112] to perform the observability analysis. Specifically, the local observability matrix for the time interval between time-steps k_o and $k_o + m$ is defined by [see (2.22)]:

$$\mathbf{M} \triangleq \begin{bmatrix} \mathbf{H}_{k_o} \\ \mathbf{H}_{k_o+1}\mathbf{\Phi}_{k_o} \\ \vdots \\ \mathbf{H}_{k_o+m}\mathbf{\Phi}_{k_o+m-1}\cdots\mathbf{\Phi}_{k_o} \end{bmatrix} \tag{3.15}$$

$$= \mathbf{M}(\mathbf{x}^\star_{k_o|k_o-1}, \mathbf{x}^\star_{k_o|k_o}, \ldots, \mathbf{x}^\star_{k_o+m|k_o+m-1}) \tag{3.16}$$

The last expression (3.16), makes explicit the fact that the observability matrix is a function of the linearization points used in computing all the Jacobians within the time interval $[k_o, k_o+m]$. In turn, this implies that *the choice of linearization points affects the observability properties* of the linearized error-state system of the EKF. This key fact is the basis of our analysis. In what

[2] For simplicity, in our analysis we assume that the relative measurements guarantee observability of the relative poses between all robots. For instance, we exclude special cases where the robots' trajectories give rise to additional unobservable modes (e.g., robots moving exactly in parallel or in a straight line [173]). Another case not considered here is that of the robots measuring relative orientation, $(\phi_j - \phi_i)$, *only*. In this case the nonlinear system has $2N+1$ unobservable d.o.f. [109]. Moreover, since the relative-orientation measurement model is *linear* in the system state, the problems caused by linearization, described in Section 3.4.2, do not appear in this case.

follows, we discuss different possible choices for linearization, and the observability properties of the corresponding linearized systems.

3.4.1 Ideal EKF-CL

Before considering the rank of the matrix $\mathbf{M}$, which is constructed using the *estimated* values of the state in the filter Jacobians, it is interesting to study the observability properties of the "oracle", or "ideal" EKF (i.e., the filter whose Jacobians are evaluated using the *true* values of the state variables, so that $\mathbf{x}^\star_{k|k-1} = \mathbf{x}^\star_{k|k} = \mathbf{x}_k$, for all k).

To make the notation more compact, we define

$$\boldsymbol{\delta}\mathbf{p}_{ij}(k,\ell) \triangleq \mathbf{p}_{i_k} - \mathbf{p}_{j_\ell} \tag{3.17}$$

which is the difference between two robots' positions at time-steps k and ℓ. Using the above definition, we note that [see (3.6)]

$$\breve{\mathbf{\Phi}}_{i_{k_o+1}} \breve{\mathbf{\Phi}}_{i_{k_o}} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{ii}(k_o+2,k_o) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.18}$$

Based on this identity, it is easy to show by induction that

$$\breve{\mathbf{\Phi}}_{i_{k_o+\ell-1}} \breve{\mathbf{\Phi}}_{i_{k_o+\ell-2}} \cdots \breve{\mathbf{\Phi}}_{i_{k_o}} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{ii}(k_o+\ell,k_o) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.19}$$

which holds for all $\ell > 0$.

In the ensuing derivations, it is assumed that every robot continuously observes all other robots in the team during the time interval $[k_o,\ k_o+m]$, i.e., the relative-measurement graph (RMG) is complete. Note that this assumption is made only to simplify the notation, and is not necessary in the analysis. We hereafter first study the case where two robots comprise the team, and then extend the analysis to the general case in which the group consists of $N > 2$ robots.

Two-robot case

Based on the assumption of a complete RMG, two measurements, $\mathbf{z}^{(12)}_{k_o+\ell}$ and $\mathbf{z}^{(21)}_{k_o+\ell}$, are available at time-step $k_o+\ell$. Thus, the measurement Jacobian $\breve{\mathbf{H}}_{k_o+\ell}$ in this case can be written as

[see (3.11)-(3.13)]:

$$\breve{\mathbf{H}}_{k_o+\ell} = \begin{bmatrix} \breve{\mathbf{H}}^{(12)}_{k_o+\ell} \\ \breve{\mathbf{H}}^{(21)}_{k_o+\ell} \end{bmatrix} = \begin{bmatrix} \breve{\mathbf{H}}^{(12)}_{1_{k_o+\ell}} & \breve{\mathbf{H}}^{(12)}_{2_{k_o+\ell}} \\ \breve{\mathbf{H}}^{(21)}_{1_{k_o+\ell}} & \breve{\mathbf{H}}^{(21)}_{2_{k_o+\ell}} \end{bmatrix} = \tag{3.20}$$

$$-\mathbf{Diag}\left((\nabla\breve{\mathbf{h}}^{(12)}_{k_o+\ell})\mathbf{A}(\phi_{1_{k_o+\ell}}),\ (\nabla\breve{\mathbf{h}}^{(21)}_{k_o+\ell})\mathbf{A}(\phi_{2_{k_o+\ell}})\right)\times$$

$$\begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{21}(k_o+\ell,k_o+\ell) & -\mathbf{I}_2 & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 \\ -\mathbf{I}_2 & \mathbf{0}_{2\times 1} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{12}(k_o+\ell,k_o+\ell) \\ \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$$

where $\mathbf{Diag}(\cdot)$ denotes a block diagonal matrix. On the other hand, the following identity is immediate [see (3.8) and (3.19)]:

$$\breve{\mathbf{\Phi}}_{k_o+\ell-1}\breve{\mathbf{\Phi}}_{k_o+\ell-2}\cdots\breve{\mathbf{\Phi}}_{k_o} =$$

$$\mathbf{Diag}\left(\breve{\mathbf{\Phi}}_{1_{k_o+\ell-1}}\cdots\breve{\mathbf{\Phi}}_{1_{k_o}}\ ,\ \breve{\mathbf{\Phi}}_{2_{k_o+\ell-1}}\cdots\breve{\mathbf{\Phi}}_{2_{k_o}}\right) =$$

$$\begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{11}(k_o+\ell,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{22}(k_o+\ell,k_o) \\ \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.21}$$

From (3.20) and (3.21) we obtain

$$\breve{\mathbf{H}}_{k_o+\ell}\breve{\mathbf{\Phi}}_{k_o+\ell-1}\breve{\mathbf{\Phi}}_{k_o+\ell-2}\cdots\breve{\mathbf{\Phi}}_{k_o} = \tag{3.22}$$

$$-\mathbf{Diag}\left((\nabla\breve{\mathbf{h}}^{(12)}_{k_o+\ell})\mathbf{A}(\phi_{1_{k_o+\ell}}),\ (\nabla\breve{\mathbf{h}}^{(21)}_{k_o+\ell})\mathbf{A}(\phi_{2_{k_o+\ell}})\right)\ \times$$

$$\begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{21}(k_o+\ell,k_o) & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{22}(k_o+\ell,k_o) \\ \mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 \\ -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{11}(k_o+\ell,k_o) & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{12}(k_o+\ell,k_o) \\ \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$$

Thus, the observability matrix, $\breve{\mathbf{M}}$, can be written as [see (3.15)]:

$$\breve{\mathbf{M}} = \underbrace{-\mathbf{Diag}\left((\nabla\breve{\mathbf{h}}_{k_o}^{(12)})\mathbf{A}(\phi_{1_{k_o}}),\ \cdots,\ (\nabla\breve{\mathbf{h}}_{k_o+m}^{(21)})\mathbf{A}(\phi_{2_{k_o+m}})\right)}_{\breve{\mathbf{D}}} \times \tag{3.23}$$

$$\underbrace{\begin{bmatrix}
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{21}(k_o,k_o) & -\mathbf{I}_2 & \mathbf{0}_{2\times1} \\
\mathbf{0}_{1\times2} & 1 & \mathbf{0}_{1\times2} & -1 \\
-\mathbf{I}_2 & \mathbf{0}_{2\times1} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{12}(k_o,k_o) \\
\mathbf{0}_{1\times2} & -1 & \mathbf{0}_{1\times2} & 1 \\
 & & & \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{21}(k_o+1,k_o) & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{22}(k_o+1,k_o) \\
\mathbf{0}_{1\times2} & 1 & \mathbf{0}_{1\times2} & -1 \\
-\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{11}(k_o+1,k_o) & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{12}(k_o+1,k_o) \\
\mathbf{0}_{1\times2} & -1 & \mathbf{0}_{1\times2} & 1 \\
\vdots & \vdots & \vdots & \vdots \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{21}(k_o+m,k_o) & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{22}(k_o+m,k_o) \\
\mathbf{0}_{1\times2} & 1 & \mathbf{0}_{1\times2} & -1 \\
-\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{11}(k_o+m,k_o) & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{12}(k_o+m,k_o) \\
\mathbf{0}_{1\times2} & -1 & \mathbf{0}_{1\times2} & 1
\end{bmatrix}}_{\breve{\mathbf{U}}}$$

Lemma 3.4.1. *The rank of the observability matrix, $\breve{\mathbf{M}}$, of the ideal EKF-CL in the two-robot case, is equal to 3.*

Proof. The rank of the product of the matrices $\breve{\mathbf{D}}$ and $\breve{\mathbf{U}}$ is given by (see (4.5.1) in [113])

$$\mathrm{rank}(\breve{\mathbf{D}}\breve{\mathbf{U}}) = \mathrm{rank}(\breve{\mathbf{U}}) - \dim(\mathrm{null}(\breve{\mathbf{D}})\bigcap\mathrm{rng}(\breve{\mathbf{U}})) \tag{3.24}$$

Denoting $\breve{\mathbf{U}} \triangleq \begin{bmatrix}\breve{\mathbf{u}}_1 & \cdots & \breve{\mathbf{u}}_6\end{bmatrix}$, it is evident that $\breve{\mathbf{u}}_1 = -\breve{\mathbf{u}}_4$, $\breve{\mathbf{u}}_2 = -\breve{\mathbf{u}}_5$, while $\breve{\mathbf{u}}_3 + \breve{\mathbf{u}}_6 = \alpha_1\breve{\mathbf{u}}_4 + \alpha_2\breve{\mathbf{u}}_5$, where $\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{21}(k_o,k_o) \triangleq -\begin{bmatrix}\alpha_1 \\ \alpha_2\end{bmatrix}$. We also note that $\{\breve{\mathbf{u}}_i\}_{i=4}^{6}$ are linearly independent. Therefore, the range of the matrix $\breve{\mathbf{U}}$ is spanned by the vectors $\breve{\mathbf{u}}_4$, $\breve{\mathbf{u}}_5$, and $\breve{\mathbf{u}}_6$, i.e.,

$$\mathrm{rng}(\breve{\mathbf{U}}) = \underset{\text{col.}}{\mathrm{span}}\begin{bmatrix}\breve{\mathbf{u}}_4 & \breve{\mathbf{u}}_5 & \breve{\mathbf{u}}_6\end{bmatrix} \tag{3.25}$$

Thus, $\text{rank}(\breve{\mathbf{U}}) = 3$. We now observe that in general $\breve{\mathbf{D}}\breve{\mathbf{u}}_i \neq \mathbf{0}$, for $i = 4, 5, 6$. Moreover, note that any vector $\mathbf{x} \in \text{rng}(\breve{\mathbf{U}}) \setminus \mathbf{0}$ can be written as $\mathbf{x} = \beta_1\breve{\mathbf{u}}_4 + \beta_2\breve{\mathbf{u}}_5 + \beta_3\breve{\mathbf{u}}_6$ for some $\beta_i \in \mathbb{R}$, where β_i, $i = 1, 2, 3$, are not simultaneously equal to zero. Thus, in general, $\breve{\mathbf{D}}\mathbf{x} = \beta_1\breve{\mathbf{D}}\breve{\mathbf{u}}_4 + \beta_2\breve{\mathbf{D}}\breve{\mathbf{u}}_5 + \beta_3\breve{\mathbf{D}}\breve{\mathbf{u}}_6 \neq \mathbf{0}$, which implies that $\mathbf{x}$ does not belong to the nullspace, $\text{null}(\breve{\mathbf{D}})$, of $\breve{\mathbf{D}}$. Therefore, $\dim(\text{null}(\breve{\mathbf{D}}) \bigcap \text{rng}(\breve{\mathbf{U}})) = 0$, and, finally, $\text{rank}(\breve{\mathbf{M}}) = \text{rank}(\breve{\mathbf{U}}) - \dim(\text{null}(\breve{\mathbf{D}}) \bigcap \text{rng}(\breve{\mathbf{U}})) = \text{rank}(\breve{\mathbf{U}}) = 3$. □

The above lemma shows that three directions of the state space are unobservable. To identify these directions, we examine the nullspace of the matrix $\breve{\mathbf{M}}$. It can be easily verified that a basis for the right nullspace of $\breve{\mathbf{U}}$ (and thus of $\breve{\mathbf{M}}$) is given by:

$$\text{null}(\breve{\mathbf{M}}) = \underset{\text{col.}}{\text{span}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\mathbf{p}_{1_{k_o}} \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\mathbf{p}_{2_{k_o}} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \triangleq \underset{\text{col.}}{\text{span}} \begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \mathbf{n}_3 \end{bmatrix} \tag{3.26}$$

From the structure of the vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ we see that a change in the state by $\Delta\mathbf{x} = \alpha\mathbf{n}_1 + \beta\mathbf{n}_2$, $\alpha, \beta \in \mathbb{R}$ corresponds to a "shifting" of the $x - y$ plane by α units along x, and by β units along y. Thus, if the two robots are shifted equally, the states $\mathbf{x}$ and $\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x}$ will be indistinguishable given the odometry and relative measurements. To understand the physical meaning of $\mathbf{n}_3$, we consider the case where the $x - y$ plane is rotated by a small angle $\delta\phi$. Rotating the coordinate system transforms any point $\mathbf{p} = [x \;\; y]^T$ to a point $\mathbf{p}' = [x' \;\; y']^T$, given by:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{C}(\delta\phi) \begin{bmatrix} x \\ y \end{bmatrix} \simeq \begin{bmatrix} 1 & -\delta\phi \\ \delta\phi & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \delta\phi \begin{bmatrix} -y \\ x \end{bmatrix}$$

where we have employed the small-angle approximations $c\delta\phi \simeq 1$ and $s\delta\phi \simeq \delta\phi$. Using this result, we see that if the plane containing the two robots is rotated by $\delta\phi$, the CL state vector

will change to

$$\mathbf{x}' = \begin{bmatrix} x_1' \\ y_1' \\ \phi_1' \\ x_2' \\ y_2' \\ \phi_2' \end{bmatrix} \simeq \begin{bmatrix} x_1 \\ y_1 \\ \phi_1 \\ x_2 \\ y_2 \\ \phi_2 \end{bmatrix} + \delta\phi \begin{bmatrix} -y_1 \\ x_1 \\ 1 \\ -y_2 \\ x_2 \\ 1 \end{bmatrix} = \mathbf{x} + \delta\phi\mathbf{n}_3 \tag{3.27}$$

which indicates that the vector $\mathbf{n}_3$ corresponds to a rotation of the $x-y$ plane. This result implies that any such global rotation is unobservable, and will cause no change to the measurements. The preceding analysis for the meaning of the basis vectors of the unobservable subspace agrees with [109] as well as with intuition, which dictates that the *global coordinates* of the state vector (rotation and translation) are unobservable, since the relative measurements only depend on the relative robot configurations.

N-robot case

We now examine the general case where $N > 2$ robots are included in the group. For a complete RMG, the measurement Jacobian matrix at time-step $k_o + \ell$ can be written as:

$$\breve{\mathbf{H}}_{k_o+\ell} = \begin{bmatrix} \breve{\mathbf{H}}_{k_o+\ell}^{(12)} \\ \vdots \\ \breve{\mathbf{H}}_{k_o+\ell}^{(1N)} \\ \\ \vdots \\ \\ \breve{\mathbf{H}}_{k_o+\ell}^{(N1)} \\ \vdots \\ \breve{\mathbf{H}}_{k_o+\ell}^{(NN-1)} \end{bmatrix} = \begin{bmatrix} \breve{\mathbf{H}}_{1_{k_o+\ell}}^{(12)} & \breve{\mathbf{H}}_{2_{k_o+\ell}}^{(12)} & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \breve{\mathbf{H}}_{1_{k_o+\ell}}^{(1N)} & 0 & \cdots & 0 & \cdots & 0 & \breve{\mathbf{H}}_{N_{k_o+\ell}}^{(1N)} \\ \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \\ \breve{\mathbf{H}}_{1_{k_o+\ell}}^{(N1)} & 0 & \cdots & 0 & \cdots & 0 & \breve{\mathbf{H}}_{N_{k_o+\ell}}^{(N1)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & \breve{\mathbf{H}}_{N-1_{k_o+\ell}}^{(NN-1)} & \breve{\mathbf{H}}_{N_{k_o+\ell}}^{(NN-1)} \end{bmatrix} \tag{3.28}$$

Similarly to (3.21), the following identity holds:

$$\breve{\mathbf{\Phi}}_{k_o+\ell-1}\breve{\mathbf{\Phi}}_{k_o+\ell-2}\cdots\breve{\mathbf{\Phi}}_{k_o} = \mathbf{Diag}\left(\breve{\mathbf{\Phi}}_{1_{k_o+\ell-1}}\cdots\breve{\mathbf{\Phi}}_{1_{k_o}}\ ,\ \cdots\ ,\ \breve{\mathbf{\Phi}}_{N_{k_o+\ell-1}}\cdots\breve{\mathbf{\Phi}}_{N_{k_o}}\right) \tag{3.29}$$

Therefore, by using the results of (3.28) and (3.29), for $\ell = 1, \cdots, m$, and proceeding similarly as in the two-robot case [see (3.22)], we obtain the observability matrix, $\breve{\mathbf{M}}$, as follows:

$$\breve{\mathbf{M}} = \underbrace{-\mathbf{Diag}\left((\nabla \breve{\mathbf{h}}_{k_o}^{(12)}) \mathbf{A}(\phi_{1_{k_o}}) \, , \cdots \, , (\nabla \breve{\mathbf{h}}_{k_o+m}^{(NN-1)}) \mathbf{A}(\phi_{N_{k_o+m}}) \right)}_{\breve{\mathbf{D}}} \times \tag{3.30}$$

$$\underbrace{\begin{bmatrix}
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{21}(k_o,k_o) & -\mathbf{I}_2 & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{31}(k_o,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{N1}(k_o,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \cdots & -\mathbf{I}_2 & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & -1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-\mathbf{I}_2 & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{1N}(k_o,k_o) \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{2N}(k_o,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{N-1N}(k_o,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{21}(k_o+1,k_o) & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{22}(k_o+1,k_o) & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{31}(k_o+1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{33}(k_o+1,k_o) & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{N1}(k_o+1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{NN}(k_o+1,k_o) \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & -1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{11}(k_o+1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{1N}(k_o+1,k_o) \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{22}(k_o+1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{2N}(k_o+1,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{N-1N}(k_o+1,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{21}(k_o+m,k_o) & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{22}(k_o+m,k_o) & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{31}(k_o+m,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{33}(k_o+m,k_o) & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{N1}(k_o+m,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{NN}(k_o+m,k_o) \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & -1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{11}(k_o+m,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{1N}(k_o+m,k_o) \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{22}(k_o+m,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{2N}(k_o+m,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{N-1N}(k_o+m,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1
\end{bmatrix}}_{\breve{\mathbf{U}}}$$

Lemma 3.4.2. *The rank of the observability matrix,* $\breve{\mathbf{M}}$*, of the ideal EKF-CL in the general* N*-robot case, is* $3N-3$*.*

Proof. Proceeding similarly to the proof of Lemma 3.4.1, by denoting $\breve{\mathbf{U}} \triangleq \begin{bmatrix} \breve{\mathbf{u}}_1 & \cdots & \breve{\mathbf{u}}_{3N} \end{bmatrix}$, we first note that

$$\breve{\mathbf{u}}_1 = -\sum_{i=2}^{N} \breve{\mathbf{u}}_{3i-2}\,, \quad \breve{\mathbf{u}}_2 = -\sum_{i=2}^{N} \breve{\mathbf{u}}_{3i-1}$$

Our next goal is to show that $\breve{\mathbf{u}}_3$ can also be expressed as a linear combination of other columns of $\breve{\mathbf{U}}$. We observe that the summation of every third column of the block row of $\breve{\mathbf{U}}$ corresponding to robot j measuring robot i at time $k_o+\ell$ is given by:

$$\begin{bmatrix} \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{ij}(k_o+\ell,k_o) - \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{ii}(k_o+\ell,k_o) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{ij}(k_o,k_o) \\ 0 \end{bmatrix}$$

We can further decompose the term $\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{ij}(k_o,k_o)$ as:

$$\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{ij}(k_o,k_o) = \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{i1}(k_o,k_o) - \mathbf{J}\boldsymbol{\delta}\mathbf{p}_{j1}(k_o,k_o)$$

Using these results, we have

$$\sum_{i=1}^{N} \breve{\mathbf{u}}_{3i} = \sum_{i=2}^{N} \alpha_{2i-1}\breve{\mathbf{u}}_{3i-2} + \sum_{i=2}^{N} \alpha_{2i}\breve{\mathbf{u}}_{3i-1} = \sum_{i=2}^{N} \begin{bmatrix} \breve{\mathbf{u}}_{3i-2} & \breve{\mathbf{u}}_{3i-1} \end{bmatrix} \begin{bmatrix} \alpha_{2i-1} \\ \alpha_{2i} \end{bmatrix}$$

where $\begin{bmatrix} \alpha_{2i-1} \\ \alpha_{2i} \end{bmatrix} \triangleq -\mathbf{J}\boldsymbol{\delta}\mathbf{p}_{i1}(k_o,k_o)$, $\forall i = 2,\ldots,N$. Now we obtain the desired result

$$\breve{\mathbf{u}}_3 = -\sum_{i=2}^{N} \breve{\mathbf{u}}_{3i} + \sum_{i=2}^{N} \alpha_{2i-1}\breve{\mathbf{u}}_{3i-2} + \sum_{i=2}^{N} \alpha_{2i}\breve{\mathbf{u}}_{3i-1}$$

Moreover, we notice that $\{\breve{\mathbf{u}}_i\}_{i=4}^{3N}$ are linearly independent. Therefore, the range of the matrix $\breve{\mathbf{U}}$ is spanned by its column vectors $\breve{\mathbf{u}}_i$, $i=4,\ldots,3N$, i.e.,

$$\mathrm{rng}(\breve{\mathbf{U}}) = \operatorname*{span}_{\text{col.}} \begin{bmatrix} \breve{\mathbf{u}}_4 & \cdots & \breve{\mathbf{u}}_{3N} \end{bmatrix} \tag{3.31}$$

Thus, $\mathrm{rank}(\breve{\mathbf{U}}) = 3N-3$. Analogously, we observe that in general $\breve{\mathbf{D}}\breve{\mathbf{u}}_i \neq \mathbf{0}$, for $i=4,\ldots,3N$. Moreover, we note that any vector $\mathbf{x} \in \mathrm{rng}(\breve{\mathbf{U}}) \setminus \mathbf{0}$ can be written as $\mathbf{x} =$

$\sum_{i=1}^{3N-3} \beta_i \breve{\mathbf{u}}_{i+3}$ for some $\beta_i \in \mathbb{R}$, where the β_i's are not simultaneously equal to zero. Thus, in general, $\breve{\mathbf{D}}\mathbf{x} = \sum_{i=1}^{3N-3} \beta_i \breve{\mathbf{D}} \breve{\mathbf{u}}_{i+3} \neq \mathbf{0}$, which implies that $\mathbf{x}$ does not belong to the nullspace, $\mathrm{null}(\breve{\mathbf{D}})$, of $\breve{\mathbf{D}}$. Therefore, $\dim(\mathrm{null}(\breve{\mathbf{D}}) \bigcap \mathrm{rng}(\breve{\mathbf{U}})) = 0$, and, finally, based on the matrix-product rank theorem (see (4.5.1) in [113]), $\mathrm{rank}(\breve{\mathbf{M}}) = \mathrm{rank}(\breve{\mathbf{U}}) - \dim(\mathrm{null}(\breve{\mathbf{D}}) \bigcap \mathrm{rng}(\breve{\mathbf{U}})) = \mathrm{rank}(\breve{\mathbf{U}}) = 3N-3$. □

Furthermore, by inspection, a basis for the right nullspace of $\breve{\mathbf{M}}$ is given by:

$$\mathrm{null}(\breve{\mathbf{M}}) = \underset{\text{col.}}{\mathrm{span}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\mathbf{p}_{1_{k_o}} \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\mathbf{p}_{2_{k_o}} \\ \mathbf{0}_{1\times 2} & 1 \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}\mathbf{p}_{N_{k_o}} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.32}$$

By noting the similarity of this result with that of (3.26), the physical interpretation of this nullspace is analogous to that of the two-robot case: the global translation and orientation of the state vector are unobservable.

3.4.2 Standard EKF-CL

We now study the observability properties of the standard EKF-CL, in which the Jacobians are evaluated at the latest state estimates (i.e., $\mathbf{x}^\star_{k|k-1} = \hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{x}^\star_{k|k} = \hat{\mathbf{x}}_{k|k}$, for all k). Similarly, we begin with the case of a two-robot team, and then generalize to the case where an arbitrary number of robots comprise the group.

We first introduce the following definitions, which will be useful for the ensuing derivations:

$$\mathbf{d}\hat{\mathbf{p}}_i(k) \triangleq \hat{\mathbf{p}}_{i_{k|k}} - \hat{\mathbf{p}}_{i_{k|k-1}} \tag{3.33}$$

$$\boldsymbol{\Delta}\hat{\mathbf{p}}_{ij}(k,\ell) \triangleq \hat{\mathbf{p}}_{i_{k|k-1}} - \hat{\mathbf{p}}_{j_{k_o|k_o-1}} - \sum_{\tau=k_o}^{\ell} \mathbf{d}\hat{\mathbf{p}}_j(\tau) \tag{3.34}$$

$$\boldsymbol{\delta}\hat{\mathbf{p}}_{ij}(k,\ell) \triangleq \hat{\mathbf{p}}_{i_{k|k-1}} - \hat{\mathbf{p}}_{j_{\ell|\ell-1}} \tag{3.35}$$

where k_o is the first time instant of interest, and $k, \ell \geq k_o$. In the above expressions, $\mathbf{d}\hat{\mathbf{p}}_i$ is the correction in the ith robot position estimate due to the EKF update, while $\boldsymbol{\delta}\hat{\mathbf{p}}_{ij}$ is the

estimated difference between two robot positions [see (3.17)] evaluated using the estimates after the respective propagation steps.

Two-robot case

We start by deriving an expression analogous to that of (3.18), using (3.6) and the definition of $\boldsymbol{\Delta}\hat{\mathbf{p}}_{ij}$ in (3.34):

$$\boldsymbol{\Phi}_{i_{k_o+1}}\boldsymbol{\Phi}_{i_{k_o}} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{ii}(k_o+2, k_o+1) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.36}$$

Using induction, we can show that

$$\boldsymbol{\Phi}_{i_{k_o+\ell-1}}\boldsymbol{\Phi}_{i_{k_o+\ell-2}}\cdots\boldsymbol{\Phi}_{i_{k_o}} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{ii}(k_o+\ell, k_o+\ell-1) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$$

for $\ell > 0$. As a result, the following identity is immediate:

$$\boldsymbol{\Phi}_{k_o+\ell-1}\boldsymbol{\Phi}_{k_o+\ell-2}\cdots\boldsymbol{\Phi}_{k_o} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{11}(k_o+\ell, k_o+\ell-1) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{22}(k_o+\ell, k_o+\ell-1) \\ \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.37}$$

The measurement Jacobian now is given by [see (3.20)]:

$$\begin{aligned} \mathbf{H}_{k_o+\ell} = -\mathbf{Diag}\left((\nabla \mathbf{h}^{(12)}_{k_o+\ell})\mathbf{A}(\hat{\phi}_{1_{k_o+\ell|k_o+\ell-1}}),\ (\nabla \mathbf{h}^{(21)}_{k_o+\ell})\mathbf{A}(\hat{\phi}_{2_{k_o+\ell|k_o+\ell-1}}) \right) \\ \times \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{21}(k_o+\ell, k_o+\ell) & -\mathbf{I}_2 & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 \\ -\mathbf{I}_2 & \mathbf{0}_{2\times 1} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{12}(k_o+\ell, k_o+\ell) \\ \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \end{aligned} \tag{3.38}$$

Multiplication of (3.38) and (3.37) yields:

$$\mathbf{H}_{k_o+\ell}\mathbf{\Phi}_{k_o+\ell-1}\cdots\mathbf{\Phi}_{k_o} = -\mathbf{Diag}\left((\nabla\mathbf{h}_{k_o+\ell}^{(12)})\mathbf{A}(\hat{\phi}_{1_{k_o+\ell|k_o+\ell-1}}),\ (\nabla\mathbf{h}_{k_o+\ell}^{(21)})\mathbf{A}(\hat{\phi}_{2_{k_o+\ell|k_o+\ell-1}})\right) \times \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{21(k_o+\ell,k_o+\ell-1)} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{22(k_o+\ell,k_o+\ell-1)} \\ \mathbf{0}_{1\times2} & 1 & \mathbf{0}_{1\times2} & -1 \\ -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{11(k_o+\ell,k_o+\ell-1)} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{12(k_o+\ell,k_o+\ell-1)} \\ \mathbf{0}_{1\times2} & -1 & \mathbf{0}_{1\times2} & 1 \end{bmatrix} \tag{3.39}$$

Thus, the observability matrix $\mathbf{M}$ (see (2.22)) can be written as:

$$\mathbf{M} = \underbrace{-\mathbf{Diag}\left((\nabla\mathbf{h}_{k_o}^{(12)})\mathbf{A}(\hat{\phi}_{1_{k_o|k_o-1}}),\ \cdots,\ (\nabla\mathbf{h}_{k_o+m}^{(21)})\mathbf{A}(\hat{\phi}_{2_{k_o+m|k_o+m-1}})\right)}_{\mathbf{D}} \times \underbrace{\begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{21(k_o,k_o)} & -\mathbf{I}_2 & \mathbf{0}_{2\times1} \\ \mathbf{0}_{1\times2} & 1 & \mathbf{0}_{1\times2} & -1 \\ -\mathbf{I}_2 & \mathbf{0}_{2\times1} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{12(k_o,k_o)} \\ \mathbf{0}_{1\times2} & -1 & \mathbf{0}_{1\times2} & 1 \\ & & & \\ \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{21(k_o+1,k_o)} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{22(k_o+1,k_o)} \\ \mathbf{0}_{1\times2} & 1 & \mathbf{0}_{1\times2} & -1 \\ -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{11(k_o+1,k_o)} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{12(k_o+1,k_o)} \\ \mathbf{0}_{1\times2} & -1 & \mathbf{0}_{1\times2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{21(k_o+m,k_o+m-1)} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{22(k_o+m,k_o+m-1)} \\ \mathbf{0}_{1\times2} & 1 & \mathbf{0}_{1\times2} & -1 \\ -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{11(k_o+m,k_o+m-1)} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{12(k_o+m,k_o+m-1)} \\ \mathbf{0}_{1\times2} & -1 & \mathbf{0}_{1\times2} & 1 \end{bmatrix}}_{\mathbf{U}} \tag{3.40}$$

Lemma 3.4.3. *The rank of the observability matrix,* $\mathbf{M}$*, of the standard EKF-CL in the two-robot case, is equal to 4.*

Proof. We first observe that the EKF update corrections in the robot position estimates, $\mathbf{d}\hat{\mathbf{p}}_i$ [see (3.33)], are in general different at different time steps. As a consequence, $\boldsymbol{\Delta}\hat{\mathbf{p}}_{ij}$ [see (3.34)]

are also different at different time steps, which means that columns 3 and 6 of matrix $\mathbf{U}$ are general column vectors and thus not linearly dependent on any other columns. Denoting $\mathbf{U} \triangleq \begin{bmatrix}\mathbf{u}_1 & \cdots & \mathbf{u}_6\end{bmatrix}$, it is evident that $\mathbf{u}_1 = -\mathbf{u}_4$, $\mathbf{u}_2 = -\mathbf{u}_5$, and moreover $\mathbf{u}_4$ and $\mathbf{u}_5$ are linearly independent. Therefore, one possible basis of the range of the matrix $\mathbf{U}$ is its columns vectors $\{\mathbf{u}_i\}_{i=3}^{6}$, i.e., $\mathrm{rng}(\mathbf{U}) = \underset{\text{col.}}{\mathrm{span}} \begin{bmatrix}\mathbf{u}_3 & \cdots & \mathbf{u}_6\end{bmatrix}$. Therefore, $\mathrm{rank}(\mathbf{U}) = 4$. By proceeding similarly to the proof of Lemma 3.4.1, we observe that in general $\mathbf{D}\mathbf{u}_i \neq \mathbf{0}$, for $i = 3, \ldots, 6$, and moreover any vector $\mathbf{x} \in \mathrm{rng}(\mathbf{U}) \setminus \mathbf{0}$ can be written as $\mathbf{x} = \sum_{i=1}^{4} \beta_i \mathbf{u}_{i+2}$ for some $\beta_i \in \mathbb{R}$, where the β_i's are not simultaneously equal to zero. As a result, in general, $\mathbf{D}\mathbf{x} = \sum_{i=1}^{4} \beta_i \mathbf{D}\mathbf{u}_{i+2} \neq \mathbf{0}$. Therefore, $\dim(\mathrm{null}(\mathbf{D}) \bigcap \mathrm{rng}(\mathbf{U})) = 0$, and finally, using theorem (4.5.1) in [113], $\mathrm{rank}(\mathbf{M}) = \mathrm{rank}(\mathbf{U}) - \dim(\mathrm{null}(\mathbf{D}) \bigcap \mathrm{rng}(\mathbf{U})) = \mathrm{rank}(\mathbf{U}) = 4$. □

We thus see that the linearized error-state model employed in the standard EKF-CL has different observability properties than that of the ideal EKF-CL. In particular, by processing the measurements collected in the time interval $[k_o, k_o + m]$, the EKF acquires information along the 4 directions of the state space corresponding to the observable subspace of the linearized system. However, the measurements actually provide information in only 3 directions of the state space (i.e., the robot-to-robot relative pose), and as a result, the EKF gains "spurious information" along the unobservable directions of the underlying nonlinear CL system, which leads to inconsistency.

To probe further, we note that the basis of the right nullspace of $\mathbf{M}$ is given by:

$$\mathrm{null}(\mathbf{M}) = \underset{\text{col.}}{\mathrm{span}} \begin{bmatrix}\mathbf{I}_2 \\ \mathbf{0}_{1\times 2} \\ \mathbf{I}_2 \\ \mathbf{0}_{1\times 2}\end{bmatrix} = \underset{\text{col.}}{\mathrm{span}} \begin{bmatrix}\mathbf{n}_1 & \mathbf{n}_2\end{bmatrix} \tag{3.41}$$

Note that these two vectors correspond to a shifting of the $x - y$ plane, which implies that such a shifting is unobservable. On the other hand, the direction corresponding to the rotation is "missing" from the unobservable subspace of the EKF system model [see (3.26) and (3.27)]. Therefore, the filter gains "nonexistent" information about the robots' global orientation. This leads to an unjustified reduction in the orientation uncertainty, which, in turn, further reduces the uncertainty in all state variables.

N-robot case

Similar results can be derived in the general case where N robots comprise the team, i.e.,

Lemma 3.4.4. *The rank of the observability matrix,* $\mathbf{M}$*, of the standard EKF-CL in the general* N*-robot case, is* $3N-2$*.*

Proof. See Appendix B.1. □

We can draw identical conclusions as in the two-robot case. In particular, the dimension of the nullspace of the observability matrix, $\mathbf{M}$, erroneously becomes 2. Furthermore, one possible basis for the nullspace can be shown to be:

$$\text{null}(\mathbf{M}) = \underset{\text{col.}}{\text{span}} \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{0}_{1\times 2} \\ \vdots \\ \mathbf{I}_2 \\ \mathbf{0}_{1\times 2} \end{bmatrix} \tag{3.42}$$

Thus, the global orientation is erroneously observable in this case as well, which leads to inconsistent estimates.

3.5 Observability-Constrained (OC)-EKF CL algorithms

In the preceding section, it was shown that when the filter Jacobians are evaluated using the latest state estimates, the error-state system model employed by the EKF has an observable subspace of dimension higher than that of the actual CL system. This will always lead to unjustified reduction of the covariance estimates, and thus to inconsistency. To address this problem, we propose computing the EKF Jacobians in such a way that guarantees an unobservable subspace of dimension three for the linearized error-state model, which precisely corresponds to Lemma 2.5.1, instead using the CL Jacobians.

Similarly, the selection of $\mathbf{N}$ [see (2.40)-(2.41)] is a design choice, which allows us to control the unobservable subspace of the EKF-CL system model. Ideally, we would like the column vectors of $\mathbf{N}$ to be identical to those in (3.32), which define the unobservable directions of the ideal EKF-CL system. However, this cannot be achieved in practice, since these

directions depend on the *true* values of the state, which are unavailable during any real-world implementation. A natural selection, which is realizable in practice, is to define the unobservable subspace of the observability matrix based on the first available state estimates, i.e., for the two-robot case to choose[3]

$$\mathbf{N} = \underset{\text{col.}}{\text{span}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{1_{k_o|k_o-1}} \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{2_{k_o|k_o-1}} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.43}$$

Once $\mathbf{N}$ has been selected, the next design decision to be made is to compute appropriate filter Jacobians at each time step. For the particular selection of $\mathbf{N}$ in (3.43), this amounts to choosing the propagation and measurement Jacobians for all $k > k_o$ to ensure that (2.41) holds (note that (2.40) is satisfied by construction in this case). Clearly, several options exist, each of which leads to a different algorithm within the general framework described here. In what follows, we present three OC-EKF algorithms to achieve this goal.

3.5.1 OC-EKF1

We start by describing the first version of the OC-EKF that was originally proposed [74]. The key idea of this approach is to choose the prior state estimates as the linearization points, so as to guarantee the appropriate observability properties of the EKF linearized system model. This procedure is explained in detail by the following lemma:

Lemma 3.5.1. *If the linearization points, at which the filter Jacobian matrices* $\mathbf{\Phi}_{i_k} = \mathbf{\Phi}_{i_k}(\mathbf{x}^\star_{i_{k+1|k}}, \mathbf{x}^\star_{i_{k|k}})$ *and* $\mathbf{H}_k^{(ij)} = \mathbf{H}_k(\mathbf{x}^\star_{i_{k|k-1}}, \mathbf{x}^\star_{j_{k|k-1}})$ *are evaluated, are selected as:*

$$\begin{aligned} \mathbf{x}^\star_{i_{k+1|k}} &= \hat{\mathbf{x}}_{i_{k+1|k}} \ , \quad \mathbf{x}^\star_{i_{k|k}} = \hat{\mathbf{x}}_{i_{k|k-1}} \\ \mathbf{x}^\star_{i_{k|k-1}} &= \hat{\mathbf{x}}_{i_{k|k-1}} \ , \ \mathbf{x}^\star_{j_{k|k-1}} = \hat{\mathbf{x}}_{j_{k|k-1}} \end{aligned} \tag{3.44}$$

then it is guaranteed that the unobservable subspace of the resulting EKF linearized error-state model is of dimension 3.

[3] When more than two robots (i.e., $N > 2$) are included in the state vector, $\mathbf{N}$ can be chosen analogously, augmented by a submatrix $\begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{i_{k_o|k_o-1}} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$ corresponding to each robot ($i = 1, 2, \ldots, N$) [75].

Proof. Using the linearization points (3.44), the state-propagation Jacobian $\mathbf{\Phi}_{i_k}$ [see (3.6)] is now computed as:

$$\mathbf{\Phi}'_{i_k} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{i_{k+1|k}} - \hat{\mathbf{p}}_{i_{k|k-1}}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.45}$$

The difference compared to (3.6), which is the Jacobian used in the standard EKF, is that the prior estimate of robot position, $\hat{\mathbf{p}}_{i_{k|k-1}}$, is used in place of the posterior estimate, $\hat{\mathbf{p}}_{i_{k|k}}$. In contrast, the measurement Jacobian, $\mathbf{H}_k^{(ij)}$, is computed in the same way as for the standard EKF [see (3.11)]. As a result, using the definition of $\boldsymbol{\delta}\hat{\mathbf{p}}_{ij}$ (3.35), the observability matrix $\mathbf{M}'$ in the OC-EKF1 algorithm for the two-robot case assumes the following form:

$$\mathbf{M}' = \underbrace{-\mathbf{Diag}\left((\nabla\mathbf{h}_{k_o}^{(12)})\mathbf{A}(\hat{\phi}_{1_{k_o|k_o-1}}),\ \cdots,\ (\nabla\mathbf{h}_{k_o+m}^{(21)})\mathbf{A}(\hat{\phi}_{2_{k_o+m|k_o+m-1}})\right)}_{\mathbf{D}'} \times$$

$$\underbrace{\begin{bmatrix}
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{21(k_o,k_o)} & -\mathbf{I}_2 & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 \\
-\mathbf{I}_2 & \mathbf{0}_{2\times 1} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{12(k_o,k_o)} \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 1 \\
 & & & \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{21(k_o+1,k_o)} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{22(k_o+1,k_o)} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 \\
-\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{11(k_o+1,k_o)} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{12(k_o+1,k_o)} \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 1 \\
\vdots & \vdots & \vdots & \vdots \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{21(k_o+m,k_o)} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{22(k_o+m,k_o)} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 \\
-\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{11(k_o+m,k_o)} & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{12(k_o+m,k_o)} \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 1
\end{bmatrix}}_{\mathbf{U}'} \tag{3.46}$$

It becomes evident that compared to the observability matrix of the ideal EKF-CL [see (3.23)], the only difference arising in $\mathbf{U}'$ is that $\boldsymbol{\delta}\mathbf{p}_{ij}$ is replaced by its estimate, $\boldsymbol{\delta}\hat{\mathbf{p}}_{ij}$, for $i, j = 1, 2$.

Moreover, by inspection, the right null space of $\mathbf{M}'$ is

$$\text{null}(\mathbf{M}') = \underset{\text{col.}}{\text{span}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{1_{k_o|k_o-1}} \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{2_{k_o|k_o-1}} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.47}$$

Thus, matrix $\mathbf{M}'$ has rank 3, which shows that the OC-EKF1 is based on an error-state system model whose unobservable subspace is of dimension 3.

Similarly, in the case where $N > 2$ robots comprise the team, it can be easily shown that the corresponding observability matrix $\mathbf{M}'$ follows the same structure as that of the ideal EKF-CL but where $\boldsymbol{\delta}\mathbf{p}_{ij}$ is replaced by its estimate, $\boldsymbol{\delta}\hat{\mathbf{p}}_{ij}$, for all $i, j = 1, \ldots, N$, i.e.,

$$\mathbf{M}' = \underbrace{-\mathbf{Diag}\left((\nabla \mathbf{h}_{k_o}^{(12)})\mathbf{A}(\hat{\phi}_{1_{k_o|k_o-1}}),\ \cdots,\ (\nabla \mathbf{h}_{k_o+m}^{(NN-1)})\mathbf{A}(\hat{\phi}_{N_{k_o+m|k_o+m-1}})\right)}_{\mathbf{D}'} \times \tag{3.48}$$

$$\underbrace{\begin{bmatrix}
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{21}(k_o,k_o) & -\mathbf{I}_2 & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{31}(k_o,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & \mathbf{0}_{2\times 1} & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{N1}(k_o,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & -\mathbf{I}_2 & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & -1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-\mathbf{I}_2 & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{1N}(k_o,k_o) \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{2N}(k_o,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{N-1N}(k_o,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{21}(k_o{+}1,k_o) & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{22}(k_o{+}1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{31}(k_o{+}1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{33}(k_o{+}1,k_o) & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{N1}(k_o{+}1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{NN}(k_o{+}1,k_o) \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & -1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{11}(k_o{+}1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{1N}(k_o{+}1,k_o) \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{22}(k_o{+}1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{2N}(k_o{+}1,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{N-1N}(k_o{+}1,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{21}(k_o{+}m,k_o) & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{22}(k_o{+}m,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{31}(k_o{+}m,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{33}(k_o{+}m,k_o) & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{N1}(k_o{+}m,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{NN}(k_o{+}m,k_o) \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & -1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{11}(k_o{+}m,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{1N}(k_o{+}m,k_o) \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{22}(k_o{+}m,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{2N}(k_o{+}m,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{N-1N}(k_o{+}m,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1
\end{bmatrix}}_{\mathbf{U}'}$$

It is not difficult to show that the observability matrix $\mathbf{M}'$ is of rank $3N-3$, and thus the unobservable subspace is of dimension 3. □

3.5.2 OC-EKF2

In the design of consistent estimators for CL, there are two competing goals that should be reconciled: (i) reduced linearization errors at each time step, and (ii) correct observability properties of the linearized system model. In OC-EKF1, the state-propagation Jacobian is computed using the predicted estimate $\hat{\mathbf{p}}_{i_{k|k-1}}$ for the robot position instead of the updated, and thus more accurate, estimate $\hat{\mathbf{p}}_{i_{k|k}}$. These two estimates can differ substantially after a large filter correction, which may introduce significant linearization errors. To formally address this limitation, we propose an alternative, termed OC-EKF2, which selects the linearization points of the EKF so as to minimize the expected squared error of the linearization points while satisfying the observability conditions [see (2.40) and (2.41)]. This can be formulated as a constrained minimization problem where the constraints express the observability requirements.

Specifically, at time-step $k+1$, we aim at minimizing the linearization error of the points $\mathbf{x}^\star_{k|k}$ and $\mathbf{x}^\star_{k+1|k}$, which appear in the filter Jacobians $\mathbf{\Phi}_k$ and $\mathbf{H}_{k+1}$ [see (3.8) and (3.11), respectively], subject to the observability constraint (2.41). Mathematically, this is expressed as:

$$\min_{\mathbf{x}^\star_{k|k},\, \mathbf{x}^\star_{k+1|k}} \int \left\|\mathbf{x}_k - \mathbf{x}^\star_{k|k}\right\|^2 p(\mathbf{x}_k|\mathbf{z}_{0:k}) d\mathbf{x}_k \; + \int \left\|\mathbf{x}_{k+1} - \mathbf{x}^\star_{k+1|k}\right\|^2 p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k}) d\mathbf{x}_{k+1} \tag{3.49}$$

$$\text{subject to } \mathbf{H}_{k+1}\mathbf{\Phi}_k \cdots \mathbf{\Phi}_{k_o}\mathbf{N} = \mathbf{0}\,, \;\; \forall k \geq k_o \tag{3.50}$$

In general, the constrained minimization problem (3.49)-(3.50) is intractable. However, when the two pdfs, $p(\mathbf{x}_k|\mathbf{z}_{0:k})$ and $p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k})$, are Gaussian distributions (which is the assumption employed in the EKF), we can solve the problem *analytically* and find a closed-form solution. In the following, we first show how the closed-form solution can be computed for the simple case where only two robots are included in the state vector, and the case of $N > 2$ robots is presented afterwards.

Two-robot case

We note that the following lemma will be helpful for the ensuing derivations:

Lemma 3.5.2. *When $p(\mathbf{x}_k|\mathbf{z}_{0:k})$ and $p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k})$ are Gaussian, the constrained optimization problem* (3.49)-(3.50) *is equivalent to:*

$$\min_{\mathbf{x}^\star_{k|k},\, \mathbf{x}^\star_{k+1|k}} \left\|\hat{\mathbf{x}}_{k|k}-\mathbf{x}^\star_{k|k}\right\|^2 + \left\|\hat{\mathbf{x}}_{k+1|k}-\mathbf{x}^\star_{k+1|k}\right\|^2 \tag{3.51}$$

$$\text{subject to } \mathbf{p}^\star_{2_{k|k}} - \mathbf{p}^\star_{1_{k|k}} = \mathbf{a}_k \tag{3.52}$$

where

$$\mathbf{a}_k = \mathbf{p}^\star_{2_{k|k-1}} - \mathbf{p}^\star_{1_{k|k-1}} - \sum_{\tau=k_o}^{k-1}(\mathbf{p}^\star_{2_{\tau|\tau}} - \mathbf{p}^\star_{2_{\tau|\tau-1}}) + \sum_{\tau=k_o}^{k-1}(\mathbf{p}^\star_{1_{\tau|\tau}} - \mathbf{p}^\star_{1_{\tau|\tau-1}})$$

Proof. See Appendix B.2. □

Using the technique of Lagrangian multipliers [16], the optimal solution to the problem (3.51)-(3.52) can be obtained as:

$$\begin{aligned}
\mathbf{p}^\star_{1_{k|k}} &= \hat{\mathbf{p}}_{1_{k|k}} + \frac{\boldsymbol{\lambda}_k}{2}\,, \quad & \phi^\star_{1_{k|k}} &= \hat{\phi}_{1_{k|k}} \\
\mathbf{p}^\star_{2_{k|k}} &= \hat{\mathbf{p}}_{2_{k|k}} - \frac{\boldsymbol{\lambda}_k}{2}\,, \quad & \phi^\star_{2_{k|k}} &= \hat{\phi}_{2_{k|k}} \\
\mathbf{x}^\star_{k+1|k} &= \hat{\mathbf{x}}_{k+1|k} &&
\end{aligned} \tag{3.53}$$

with

$$\boldsymbol{\lambda}_k = \hat{\mathbf{p}}_{2_{k|k}} - \hat{\mathbf{p}}_{1_{k|k}} - \mathbf{a}_k$$

We see that $\boldsymbol{\lambda}_k$ and thus the linearization point for the position of each robot, $\mathbf{p}^\star_{i_{k|k}}$, depends on all robots' estimates. This increases the complexity of implementing the algorithm, but yields the optimal linearization errors under the desired observability constraints.

Using the linearization points in (3.53), the state-propagation Jacobians in the OC-EKF2 are now computed as:

$$\boldsymbol{\Phi}''_{1_k} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{1_{k+1|k}} - \hat{\mathbf{p}}_{1_{k|k}} - \frac{\boldsymbol{\lambda}_k}{2}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.54}$$

$$\boldsymbol{\Phi}''_{2_k} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{2_{k+1|k}} - \hat{\mathbf{p}}_{2_{k|k}} + \frac{\boldsymbol{\lambda}_k}{2}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.55}$$

while the measurement Jacobians are calculated in the same way as in the standard EKF [see (3.11)].

N-robot case

We now consider the case where more than two robots (i.e., $N > 2$) are included in the state vector. Analogous to Lemma 3.5.2, we start by proving the following lemma:

Lemma 3.5.3. *When $p(\mathbf{x}_k|\mathbf{z}_{0:k})$ and $p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k})$ are Gaussian, the constrained optimization problem* (3.49)-(3.50) *in the general N-robot case, is equivalent to:*

$$\min_{\mathbf{x}^\star_{k|k},\, \mathbf{x}^\star_{k+1|k}} \left\|\hat{\mathbf{x}}_{k|k}-\mathbf{x}^\star_{k|k}\right\|^2 + \left\|\hat{\mathbf{x}}_{k+1|k}-\mathbf{x}^\star_{k+1|k}\right\|^2 \tag{3.56}$$

$$\text{subject to } \ \mathbf{p}^\star_{i_{k|k}} - \mathbf{p}^\star_{1_{k|k}} = \mathbf{a}_{i1_k}\ ,\ \forall i = 2,\ldots,N \tag{3.57}$$

where

$$\mathbf{a}_{i1_k} = \mathbf{p}^\star_{i_{k|k-1}} - \mathbf{p}^\star_{1_{k|k-1}} - \sum_{\tau=k_o}^{k-1}(\mathbf{p}^\star_{i_{\tau|\tau}} - \mathbf{p}^\star_{i_{\tau|\tau-1}}) + \sum_{\tau=k_o}^{k-1}(\mathbf{p}^\star_{1_{\tau|\tau}} - \mathbf{p}^\star_{1_{\tau|\tau-1}})$$

Proof. See Appendix B.3. □

It should be pointed out that we here assume that all the robots are connected in the RMG (otherwise, the isolated robots, which neither have measurements nor are observed by any other robots, will not impose any constraint). Now we employ the technique of Lagrangian multipliers [16] to solve the problem (3.56)-(3.57). Specifically, the Lagrangian function can be constructed as:

$$\mathcal{L}(\mathbf{x}^\star_{k|k}, \mathbf{x}^\star_{k+1|k}, \boldsymbol{\lambda}_{2_k}, \ldots, \boldsymbol{\lambda}_{N_k}) = \tag{3.58}$$

$$\left\|\hat{\mathbf{x}}_{k|k}-\mathbf{x}^\star_{k|k}\right\|^2 + \left\|\hat{\mathbf{x}}_{k+1|k}-\mathbf{x}^\star_{k+1|k}\right\|^2 + \sum_{i=2}^{N}\boldsymbol{\lambda}_{i_k}^T\left(\mathbf{p}^\star_{i_{k|k}} - \mathbf{p}^\star_{1_{k|k}} - \mathbf{a}_{i1_k}\right)$$

By setting the derivatives of the Lagrange function with respect to the optimization variables to zero, we have:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{p}^\star_{1_{k|k}}} = -2(\hat{\mathbf{p}}_{1_{k|k}} - \mathbf{p}^\star_{1_{k|k}}) - \sum_{i=2}^{N}\boldsymbol{\lambda}_{i_k} = \mathbf{0} \tag{3.59}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{p}^\star_{i_{k|k}}} = -2(\hat{\mathbf{p}}_{i_{k|k}} - \mathbf{p}^\star_{i_{k|k}}) + \boldsymbol{\lambda}_{i_k} = \mathbf{0}\ ,\ \forall i = 2,\ldots,N \tag{3.60}$$

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\lambda}_{i_k}} = \mathbf{p}^\star_{i_{k|k}} - \mathbf{p}^\star_{1_{k|k}} - \mathbf{a}_{i1_k} = \mathbf{0}\ ,\ \forall i = 2,\ldots,N \tag{3.61}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{x}^\star_{\text{other}}} = -2(\hat{\mathbf{x}}_{\text{other}} - \mathbf{x}^\star_{\text{other}}) = \mathbf{0} \tag{3.62}$$

where $\mathbf{x}^\star_{\text{other}}$ denotes all the optimization variables except the ones in (3.59)-(3.61). From the above equations, we obtain the optimal solution in closed form as follows:

$$\mathbf{p}^\star_{1_{k|k}} = \hat{\mathbf{p}}_{1_{k|k}} + \frac{\sum_{i=2}^{N} \boldsymbol{\lambda}_{i_k}}{2} \ , \ \ \mathbf{p}^\star_{i_{k|k}} = \hat{\mathbf{p}}_{i_{k|k}} - \frac{\boldsymbol{\lambda}_{i_k}}{2} \ , \ \ \mathbf{x}^\star_{\text{other}} = \hat{\mathbf{x}}_{\text{other}} \tag{3.63}$$

where the Lagrangian multipliers are attained by solving a linear system. In particular, substituting the above optimal linearization points (3.63) into (3.61), we have the following set of equations with respect to $\boldsymbol{\lambda}_{i_k}$ ($i = 2, \ldots, N$):

$$\boldsymbol{\lambda}_{j_k} + \frac{1}{2} \sum_{i=2, i\neq j}^{N} \boldsymbol{\lambda}_{i_k} = \left(\hat{\mathbf{p}}_{j_{k|k}} - \hat{\mathbf{p}}_{1_{k|k}} \right) - \mathbf{a}_{j1_k} =: \mathbf{b}_j \ , \ \forall j = 2, \ldots, N \tag{3.64}$$

By stacking the above equations into a matrix form, we have

$$\underbrace{\begin{bmatrix} \mathbf{I} & \frac{1}{2}\mathbf{I} & \cdots & \frac{1}{2}\mathbf{I} \\ \frac{1}{2}\mathbf{I} & \mathbf{I} & \cdots & \frac{1}{2}\mathbf{I} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2}\mathbf{I} & \frac{1}{2}\mathbf{I} & \cdots & 2\mathbf{I} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \boldsymbol{\lambda}_{2_k} \\ \boldsymbol{\lambda}_{3_k} \\ \vdots \\ \boldsymbol{\lambda}_{N_k} \end{bmatrix}}_{\boldsymbol{\lambda}_k} = \underbrace{\begin{bmatrix} \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{b}_N \end{bmatrix}}_{\mathbf{b}} \Rightarrow \ \boldsymbol{\lambda}_k = \mathbf{A}^{-1}\mathbf{b} \tag{3.65}$$

Using the optimal linearization points (3.63), we compute the state-propagation Jacobians in the similar form as in the two-robot case [see (3.54)-(3.55)], while the measurement Jacobians are calculated in the same way as in the standard EKF [see (3.11)].

3.5.3 OC-EKF3

We now describe our OC-EKF3 algorithm. In particular, the OC-EKF3 directly computes the measurement Jacobian by projecting the most accurate measurement Jacobian onto the observable subspace, while it calculates the propagation Jacobian identically as the standard EKF [see (3.6)]. By doing so, it ensures that the EKF-CL system model has an observable subspace of correct dimensions.

Specifically, in analogy to the OC-EKF3 for SLAM (see Section 2.5.3), we aim to find the measurement Jacobian closest to the ideal one while satisfying the observability constraint (2.41), i.e.,

$$\min_{\mathbf{H}_{k+1}} \ ||\mathbf{H}_o - \mathbf{H}_{k+1}||_F^2 \tag{3.66}$$

$$\text{subject to } \ \mathbf{H}_{k+1}\boldsymbol{\Phi}_k \cdots \boldsymbol{\Phi}_{k_o}\mathbf{N} = \mathbf{0} \tag{3.67}$$

where $\mathbf{H}_o$ ideally is the measurement Jacobian evaluated at the true states. However, since in practice the true states are generally not available, we instead compute it using the current best state estimates as in the standard EKF (3.11), i.e., $\mathbf{H}_o = \mathbf{H}_o(\hat{\mathbf{x}}_{k+1|k})$. The optimal solution is obtained by application of the following lemma:

Lemma 3.5.4. *The optimal solution to the constrained minimization problem* (3.66)-(3.67) *is:*

$$\mathbf{H}_{k+1}^{'''} = \mathbf{H}_o \left(\mathbf{I}_{\dim(\mathbf{x})} - \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\right) \tag{3.68}$$

where $\mathbf{V} \triangleq \mathbf{\Phi}_k \cdots \mathbf{\Phi}_{k_o}\mathbf{N}$.

Proof. Analogous to the proof of Lemma 2.5.4. □

Note that $\mathbf{V}$ in the above lemma is the propagated unobservable subspace at time-step $k+1$, and $\left(\mathbf{I}_{\dim(\mathbf{x})} - \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\right)$ is the subspace orthogonal to $\mathbf{V}$, i.e., the observable subspace. Hence, as seen from (3.68), the measurement Jacobian of the OC-EKF3 is the projection of the best-available measurement Jacobian onto the observable subspace. It is also important to observe that in the case of $N > 2$ robots, each measurement only depends on the poses of the observing robot and the observed robot and hence the measurement Jacobian $\mathbf{H}_{k+1}^{'''}$ typically has sparse structure [see (3.11)]. Based on this observation and by exploiting the sparse structure of the Jacobian matrix, we only need to consider the nonzero submatrices of the measurement Jacobian and solve a reduced-size problem similar to (3.66)-(3.67). Once the optimal solution is attained, we can easily construct the full measurement Jacobian by padding it with zeros [see (3.11)].

Remarks It is important to point out that, as compared to the standard EKF, the *only* change in the OC-EKFs is the way in which the state-propagation and measurement Jacobians are computed [see (3.45), (3.54), (3.55), and (3.68)], while the state estimates and covariance are propagated and updated in the same way as in the standard EKF. For clarity, the steps of the OC-EKF CL algorithms are outlined in Algorithms 2, and a simple CL example with two robots using the unicycle motion model and relative distance and bearing measurements is provided in Appendix B.4. We stress that even though a complete RMG (i.e., each robot can observe all others) is assumed at every time step in the preceding analysis, this is not a necessary assumption for the OC-EKFs, as the analysis can easily be extended to the case of limited sensor range, where multiple propagation steps occur between updates (see Section 3.7). We also point out that the

Algorithm 2 Observability-Constrained (OC)-EKFs for CL

Require: Initial state estimates and covariance

1: **loop**

2: `Propagation:` If proprioceptive (e.g., odometric) measurements are available,

3: propagate the state estimates [see (3.2) and (3.3)]

4: compute the propagation Jacobian [see (3.45) for OC-EKF1, (3.54)-(3.55) for OC-EKF2, and (3.6) for OC-EKF3]

5: propagate the state covariance:

$$\mathbf{P}_{k+1|k} = \mathbf{\Phi}_k \mathbf{P}_{k|k} \mathbf{\Phi}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \tag{3.69}$$

6: `Update:` If exteroceptive measurements are available,

7: compute the measurement residual:

$$\mathbf{r}_{k+1} = \mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}) \tag{3.70}$$

8: compute the measurement Jacobian [see (3.11) for OC-EKF1/2, and (3.68) for OC-EKF3]

9: compute the residual covariance and the Kalman gain:

$$\mathbf{S}_{k+1} = \mathbf{H}_{k+1} \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1} \tag{3.71}$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T \mathbf{S}_{k+1}^{-1} \tag{3.72}$$

10: update the state estimate and covariance:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} \mathbf{r}_{k+1} \tag{3.73}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1} \mathbf{S}_{k+1} \mathbf{K}_{k+1}^T \tag{3.74}$$

11: **end loop**

new OC-EKFs are *causal* and *realizable* in practice, since they do not utilize any knowledge of the future and true state. Interestingly, even though the proposed filters do not use the latest state estimates for computing the Jacobians (and thus the Jacobians are less accurate than those of the standard EKF), they exhibit better consistency than the standard EKF. As a final remark, it is straightforward to extend the proposed OC-EKFs to C-SLAM. In particular, by performing observability analysis, we can show that the unobservable subspace of the ideal-EKF linearized system (as well as the underlying nonlinear system) has three d.o.f. corresponding to the global translation and rotation, while the system model employed by the standard EKF has unobservable subspace of dimension two. As is the case for SLAM and CL, this is the primary cause of filter inconsistency. Thus, we can adapt the same observability-based methodology and develop OC-EKFs for C-SLAM.

3.6 Simulation results

A series of Monte-Carlo comparison studies were conducted under various conditions, in order to validate the preceding theoretical analysis and to demonstrate the capability of the OC-EKF estimators to improve the consistency of EKF-CL. The metrics used to evaluate filter performance are RMSE and NEES [14]. It is known that the NEES of an M-dimensional Gaussian random variable follows a χ^2 distribution with M d.o.f. Therefore, if a certain filter is consistent, we expect that the average NEES for each robot pose will be close to 3 for all time steps. The larger the deviation of the NEES from these values is, the worse the inconsistency of the filter is. By studying both the RMSE and NEES of all the filters considered here, we obtain a comprehensive picture of the filters' performance.

In the simulation tests, we consider a CL scenario in which four robots move randomly in an area of size 20 m $\times$ 20 m. 50 Monte-Carlo simulations were performed, and during each run, all filters process the same data, to ensure a fair comparison. The five estimators compared are: (1) the ideal EKF, (2) the standard EKF, (3) the OC-EKF1, (4) the OC-EKF2, and (5) the OC-EKF3.

For the results presented in this section, four identical robots with a simple differential drive model move on a planar surface, at a constant linear velocity of $v = 0.25$ m/sec, while the rotational velocity is drawn from the uniform distribution over $[-0.5, 0.5]$ rad/sec. The two drive wheels are equipped with encoders, which measure their revolutions and provide noisy

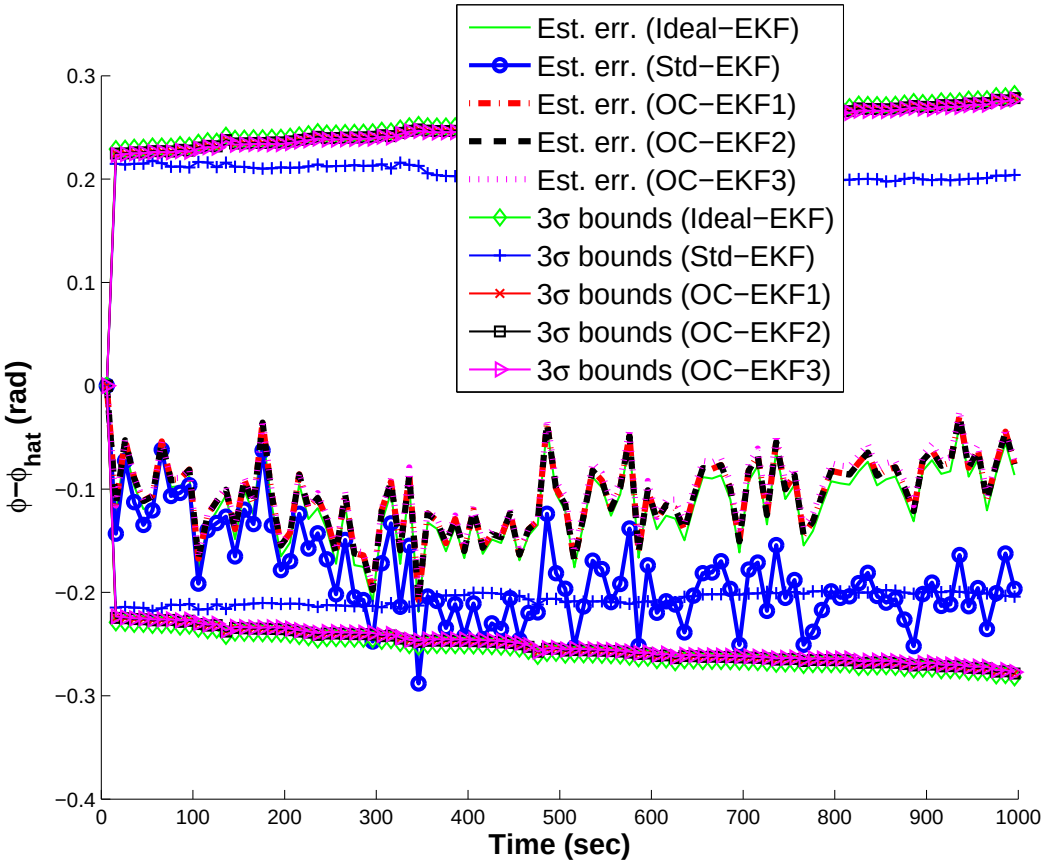
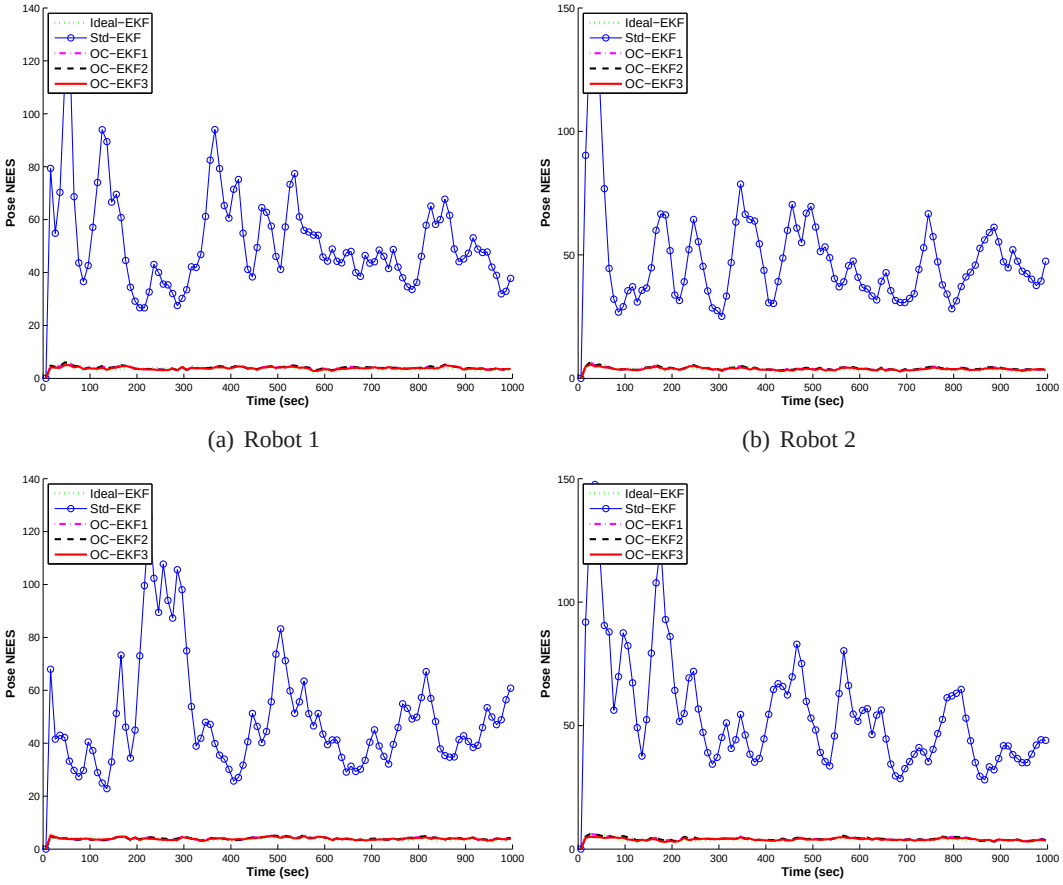


Figure 3.1: Orientation estimation errors vs. 3σ bounds for one robot of the 4-robot team, obtained from one typical realization of the CL Monte-Carlo simulations. The results for the other robots are similar to the ones presented here. The σ values are computed as the square-root of the corresponding element of the estimated covariance matrix. Note that the estimation errors as well as the 3σ bounds of the ideal and the OC-EKFs are almost identical, which makes the corresponding lines difficult to distinguish.

measurements of velocity (i.e., right and left wheel velocities, v_r and v_l, respectively), with standard deviation equal to $\sigma = 5\%v$ for each wheel. These measurements are used to obtain linear and rotational velocity measurements for each robot, which are given by $v = \frac{v_r+v_l}{2}$ and $\omega = \frac{v_r-v_l}{a}$, where $a = 0.5$ m is the distance between the drive wheels. Thus, the standard deviations of the linear and rotational velocity measurements are $\sigma_v = \frac{\sqrt{2}}{2}\sigma$ and $\sigma_\omega = \frac{\sqrt{2}}{a}\sigma$, respectively.

Each robot records distance and bearing measurements to all other robots. Note that for



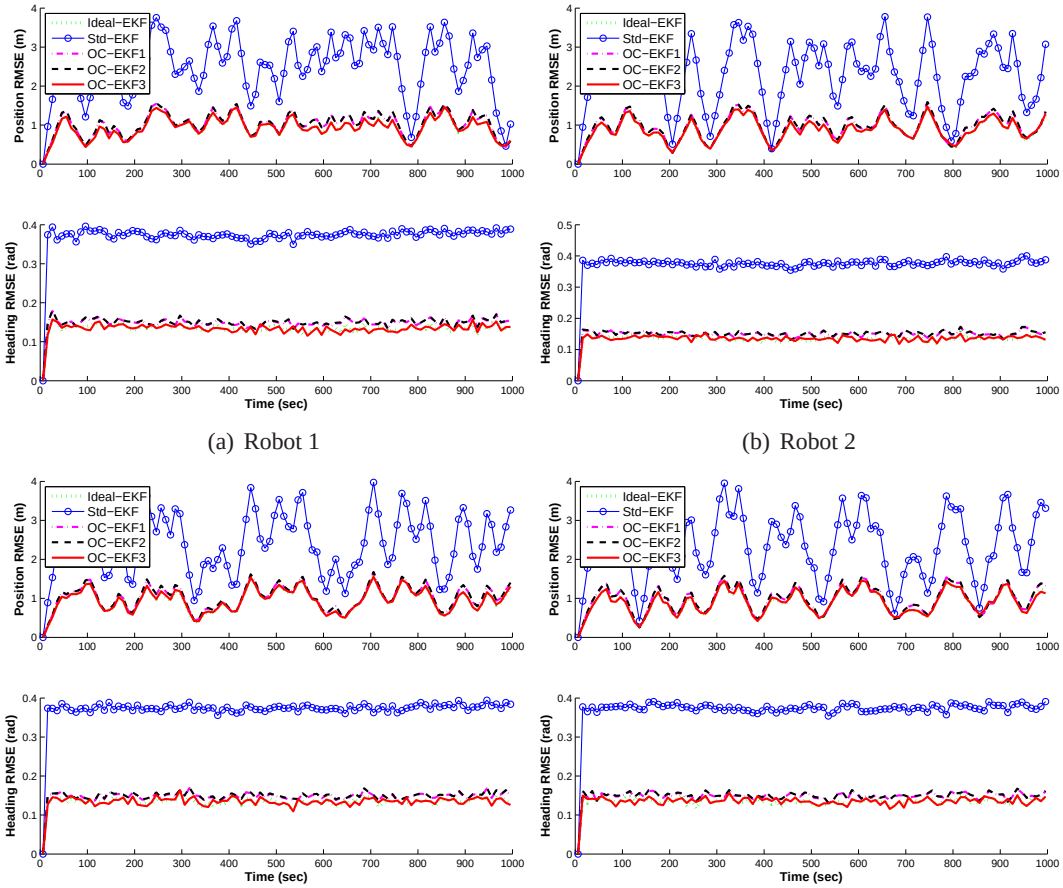
(a) Robot 1 (b) Robot 2

(c) Robot 3 (d) Robot 4

Figure 3.2: CL Monte-Carlo simulation results for the average NEES of the robot poses. In these plots, the dotted lines correspond to the ideal EKF, the solid lines with circles to the standard EKF, the dash-dotted lines to the OC-EKF1, the dashed lines to the OC-EKF2, and the solid lines to the OC-EKF3. Note that the NEES of the ideal EKF and the OC-EKFs are almost identical, which makes the corresponding lines difficult to distinguish.

simplicity we assume that all measurements occur at every time step in our simulations (but this is not the case in our real-world experiments in Section 3.7). The standard deviation of the distance and bearing measurement noise was set to $\sigma_d = 0.1$ m and $\sigma_\theta = 5$ deg, respectively.

Fig. 3.1 shows the orientation estimation errors for one of the robots, obtained from a typical simulation (the results for the other three robots are very similar and thus omitted for clarity). Clearly, the standard-EKF errors grow significantly faster than those of the ideal EKF and the OC-EKFs, which indicates that the standard EKF tends to diverge. Note also that although the



(a) Robot 1 (b) Robot 2

(c) Robot 3 (d) Robot 4

Figure 3.3: CL Monte-Carlo simulation results for the average RMSE of the robot poses. In these plots, the dotted lines correspond to the ideal EKF, the solid lines with circles to the standard EKF, the dash-dotted lines to the OC-EKF1, the dashed lines to the OC-EKF2, and the solid lines to the OC-EKF3. Note that the RMSE of the ideal EKF and the OC-EKFs are very close, which makes the corresponding lines difficult to distinguish.

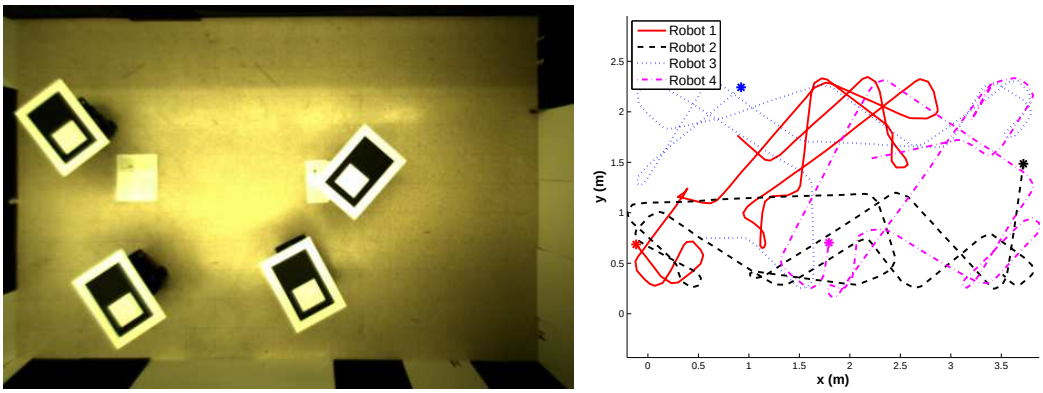
orientation errors of the ideal EKF and the OC-EKFs remain well within their corresponding 3σ bounds, those of the standard EKF exceed them. Most importantly, in contrast to those of the OC-EKFs, the 3σ bounds of the standard EKF (computed from the square-root of the corresponding element of the estimated covariance matrix) remain almost *constant* as if the orientation of the robot was observable. However, as discussed in Section 3.4, the robots have no

Table 3.1: CL simulation results for robot pose estimation performance

	Ideal-EKF	Std-EKF	OC-EKF1	OC-EKF2	OC-EKF3
Robot Position RMSE (m)					
Robot 1:	0.9378	2.5069	1.0331	1.0432	0.9388
Robot 2:	0.8807	2.2965	0.9575	0.9667	0.8861
Robot 3:	0.9533	2.4369	1.0286	1.0378	0.9594
Robot 4:	0.8950	2.3567	0.9728	0.9824	0.9013
Robot Heading RMSE (rad)					
Robot 1:	0.1356	0.3751	0.1495	0.1511	0.1358
Robot 2:	0.1358	0.3751	0.1498	0.1514	0.1362
Robot 3:	0.1357	0.3744	0.1493	0.1509	0.1356
Robot 4:	0.1355	0.3742	0.1490	0.1506	0.1362
Robot Pose NEES					
Robot 1:	3.7566	52.4160	4.1204	4.1440	3.9173
Robot 2:	3.7004	48.3953	4.0329	4.0559	3.8379
Robot 3:	3.7691	49.6612	4.0607	4.0759	3.9933
Robot 4:	3.7878	55.5077	4.2080	4.2341	3.9783

access to absolute orientation information and thus the orientation covariance should continuously grow (as is the case for the ideal EKF and the OC-EKFs). The results of Fig. 3.1 clearly demonstrate that the incorrect observability properties of the standard EKF cause an unjustified reduction of the orientation uncertainty.

Figs. 3.2 and 3.3 show the average NEES and RMSE, respectively, for all four robots. These plots show the average errors over all Monte-Carlo runs, plotted over time, while Table 3.1 presents the average error values over all time steps. As evident, the performance of both the OC-EKFs is *almost identical* to that of the ideal EKF, and substantially better than the standard EKF, both in terms of RMSE and NEES. This occurs even though the Jacobians used in the OC-EKFs are less accurate than those used in the standard EKF, as explained in the preceding section. This fact indicates that the errors introduced by the use of inaccurate Jacobians have a less detrimental effect on consistency than the use of an error-state system model with



(a) (b)

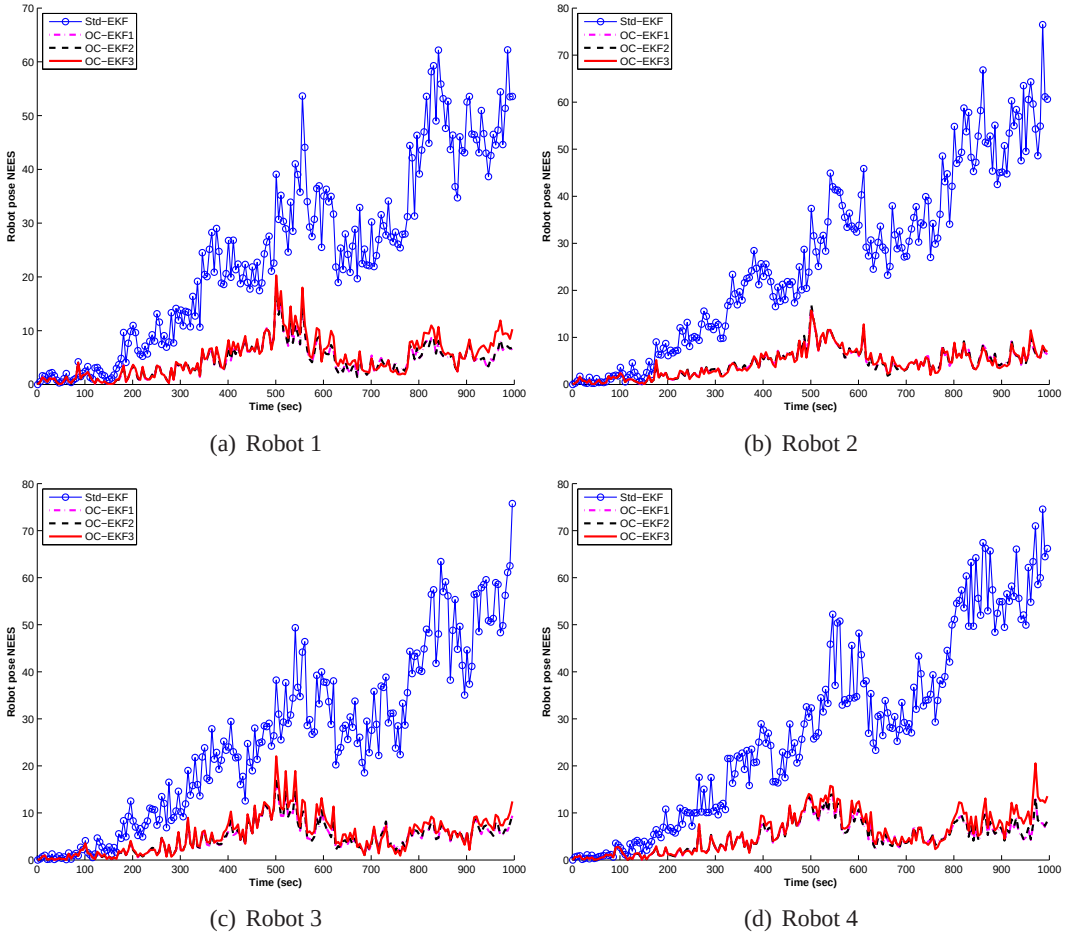
Figure 3.4: CL experimental setup: (a) Calibrated image of four Pioneer I robots with targets mounted on top of them. (b) Trajectories of four Pioneer I robots that move inside a 2.5 m $\times$ 4.5 m arena during the indoor experiment. For presentation clarity, only the parts of the trajectories corresponding to the first 200 sec are plotted. Starting positions are marked by $*$.

observable subspace of dimension higher than that of the actual CL system.

3.7 Experimental results

In what follows, we describe one real-world experiment performed to further validate the OC-EKF algorithms. During the test, a team of four Pioneer I robots move in a rectangular area of 2.5 m $\times$ 4.5 m, within which the positions of the robots are being tracked by an overhead camera. For this purpose, rectangular targets are mounted on top of the robots and the vision system is calibrated in order to provide ground-truth measurements of the robots' poses in a global coordinate frame. The standard deviation of the noise in these measurements is approximately 0.5 deg for orientation and 0.01 m, along each axis, for position. The robots were commanded to move at a constant velocity of $v = 0.1$ m/sec while avoiding collision with the boundaries of the arena as well as with their teammates. Fig. 3.4(a) shows the experimental setup. The trajectories of the four robots are shown in Fig. 3.4(b), where only partial trajectories are plotted in order to keep the figure clear.

Although four identical robots were used, calibration of their odometric sensors showed that the accuracy of the wheel-encoders' measurements is not identical for all robots. Specifically, the measurement errors are well-modeled as Gaussian zero-mean white noise processes

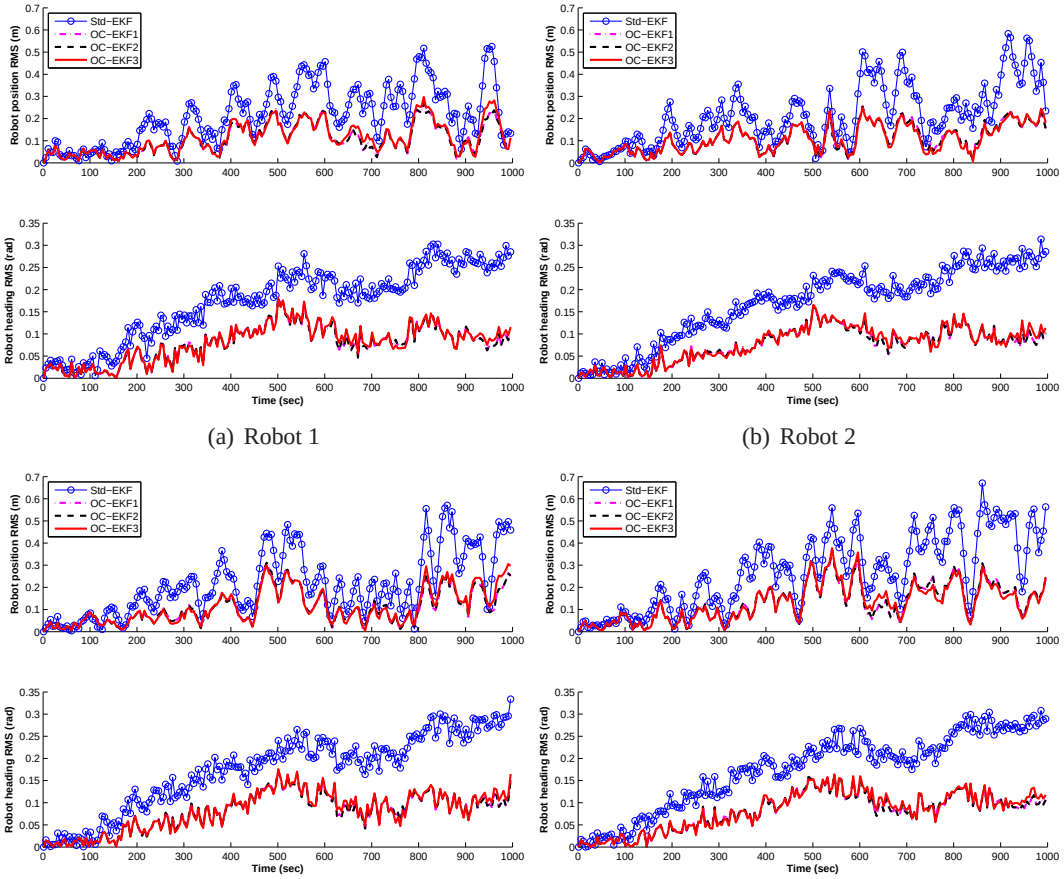


(a) Robot 1 (b) Robot 2

(c) Robot 3 (d) Robot 4

Figure 3.5: CL experimental results for the NEES of the robot poses. In these plots, the solid lines with circles correspond to the standard EKF, the dash-dotted lines to the OC-EKF1, the dashed lines to the OC-EKF2, and the solid lines to the OC-EKF3. Note that the NEES of the two OC-EKFs are almost identical, which makes the corresponding lines difficult to distinguish.

and the standard deviation of the velocity measurements ranges from $\sigma_{v_{\min}} = 3.8\%v$ for the most accurate odometer to $\sigma_{v_{\max}} = 6.9\%v$ for the robot with the highest noise levels. Similarly, the standard deviations of the rotational velocity measurements have values between $\sigma_{\omega_{\min}} = 0.0078$ rad/sec and $\sigma_{\omega_{\max}} = 0.02$ rad/sec for the four robots. We observe that as a result of the variability of sensor characteristics, attributed to manufacturing imperfections, the experiment involves a *heterogeneous* robot team, despite all robots being the same model, equipped with the same sensors. This gives us the opportunity to test the performance of the



(a) Robot 1 (b) Robot 2

(c) Robot 3 (d) Robot 4

Figure 3.6: CL experimental results for the estimation errors of the robot poses. In these plots, the solid lines with circles correspond to the standard EKF, the dash-dotted lines to the OC-EKF1, the dashed lines to the OC-EKF2, and the solid lines to the OC-EKF3. Note that the estimation errors of the OC-EKFs are almost identical, which makes the corresponding lines difficult to distinguish.

OC-EKF algorithms in a realistic scenario. We stress that the derivations of the OC-EKFs in Section 3.5 require neither the homogeneity of robot teams, nor a complete RMG at every time step. Besides the previous simulations in which a homogeneous robot team was used, this experiment demonstrates the superior performance of the OC-EKFs versus the standard EKF also for heterogeneous robot teams.

Relative distance-and-bearing measurements are produced synthetically using the differences in the positions of the robots, as these are recorded by the overhead camera, with the

Table 3.2: CL experimental results for robot pose estimation performance

	Std-EKF	OC-EKF1	OC-EKF2	OC-EKF3
Robot Position RMSE (m)				
Robot 1:	0.2132	0.1070	0.1066	0.1121
Robot 2:	0.2127	0.1083	0.1080	0.1060
Robot 3:	0.2104	0.1076	0.1073	0.1105
Robot 4:	0.2699	0.1317	0.1313	0.1301
Robot Heading RMSE (rad)				
Robot 1:	0.1721	0.0785	0.0782	0.0800
Robot 2:	0.1694	0.0760	0.0757	0.0776
Robot 3:	0.1732	0.0794	0.0791	0.0808
Robot 4:	0.1749	0.0810	0.0807	0.0827
Robot Pose NEES				
Robot 1:	24.2458	4.4080	4.4289	5.0305
Robot 2:	26.4881	4.5423	4.5801	4.5385
Robot 3:	25.3439	4.6060	4.6270	5.2248
Robot 4:	27.6313	4.9182	4.9501	5.7714

addition of noise. For the experimental results shown in this section, the distance and bearing measurements are corrupted by zero-mean white Gaussian noise processes, with standard deviation $\sigma_d = 0.05$ m and $\sigma_\theta = 2$ deg, respectively.

Four filters were implemented: (1) the standard EKF, (2) the OC-EKF1, (3) the OC-EKF2, and (4) the OC-EKF3. Comparative results for the three filters are presented in Figs. 3.5 and 3.6, while Table 3.2 shows the averaged NEES and RMSE of the robot pose, respectively. From the experimental results it becomes clear that the three OC-EKFs outperform the standard EKF, in terms of both accuracy and consistency, while both perform almost identically. This agrees with the simulation results presented in the preceding section. Both the real-world and simulation results thus support our conjecture that the mismatch in the dimension of the unobservable subspace between the linearized CL system and the underlying nonlinear system is a fundamental cause of filter inconsistency.

3.8 Summary

In this chapter, we have studied in depth the consistency issue of EKF-CL from an observability perspective. By comparing the observability properties of the linearized error-state model employed in the EKF with those of the underlying nonlinear CL system, we proved that the observable subspace of the standard EKF system model is always of higher dimension than that of the actual CL system. As a result, the estimated covariance of the EKF undergoes reduction in directions of the state space where no information is available, thus leading to inconsistency. Moreover, based on the analysis, we proposed three new OC-EKF algorithms, which significantly improve the consistency of EKF-CL. The design methodology followed is based on appropriately computing the EKF Jacobians to ensure that the observable subspace of the linearized error-state system model is of the same dimensions as that of the underlying nonlinear system. Extensive simulation tests and real-world experiments have verified that the proposed OC-EKFs perform better, in terms of both accuracy and consistency, than the standard EKF.

Chapter 4

Quadratic-Complexity Observability-Constrained UKF for SLAM

In this chapter, we adapt the observability-based methodology presented in the previous chapters to the UKF, and develop an Observability-Constrained (OC)-UKF to improve the UKF consistency. Moreover, we introduce a new sampling scheme to reduce the UKF computational complexity, which samples only a small (constant-size) subset of the states involved in the process and measurement models at each time step. As a result, when applied to the SLAM problem, the proposed UKF with the new sampling strategy attains quadratic complexity. Parts of this chapter have been published in [68, 71].

4.1 Introduction

For autonomous vehicles exploring unknown environments, the ability to perform simultaneous localization and mapping (SLAM) is essential. Among the algorithms developed thus far to solve the SLAM problem, the EKF remains a popular choice and has been used in many applications [92, 125, 169], primarily due to its relative low computational complexity and ease of implementation. However, EKF-based SLAM is vulnerable to linearization errors, which

can cause poor performance or even divergence, and its state estimates are typically *inconsistent* [12, 66, 70, 72, 78, 79, 85]. As defined in [14], a state estimator is *consistent* if the estimation errors are zero-mean, and have covariance smaller or equal to the one calculated by the estimator. Consistency is one of the primary criteria for evaluating the performance of any estimator; if an estimator is inconsistent, then the accuracy of the computed state estimates is unknown, which in turn makes the estimator unreliable. In order to reduce the estimation errors due to linearization, the UKF [84] was introduced. The UKF has been shown to generally perform better than the EKF in nonlinear estimation problems, and one would expect similar gains in the case of SLAM.

However, one of the main limitations of the standard (i.e., original) UKF algorithm [84] is its computational complexity, which is *cubic* in the size of the state vector. In the case of SLAM, where hundreds of landmarks are typically included in the state vector, this increased computational burden can preclude real-time operation. Moreover, when applied to SLAM, the performance gains of the UKF over the EKF are generally not overwhelming (see [62, 63, 110]). Most importantly, empirical evidence suggests [62, 63, 71, 110] that the UKF also results in inconsistent estimates in SLAM, even though its performance is better than the EKF in this respect.

Our objective in this chapter is to address the aforementioned limitations of UKF-based SLAM. In particular, the main contributions of this work are the following:

- We introduce a new sampling strategy for UKF-based SLAM that has *constant* computational cost, regardless of the number of landmarks included in the state vector. This sampling scheme is provably optimal, in the sense that it minimizes the expected squared error between the nonlinear function and its linear approximation employed by the UKF. Using this strategy, the computational cost of UKF-based SLAM becomes *linear* during propagation and *quadratic* during update, which is of the same order as that of EKF-based SLAM. We stress that this new UKF sampling strategy is applicable to a large class of nonlinear estimation problems (not only the SLAM problem) where the measurements at each time step are of dimension lower than the state.

- We analytically examine the consistency of UKF-based SLAM, by studying the observability properties of the statistically-linearized (i.e., linear-regression-based) system

model employed by the UKF. This analysis identifies a mismatch between the observability properties of this model and those of the underlying nonlinear system, which is a fundamental cause of inconsistency. Based on this theoretical analysis, we propose a novel UKF-based SLAM algorithm, termed Observability-Constrained (OC)-UKF SLAM. By imposing the appropriate observability constraints on the linear regression carried out by the UKF, the proposed OC-UKF ensures that its system model has observability properties similar to those of the underlying nonlinear SLAM system. As a result, the OC-UKF outperforms the standard UKF as well as other state-of-the-art algorithms, in terms of accuracy and consistency, as validated by both simulation and experimental tests.

4.2 Related work

The SLAM problem has received considerable attention over the past two decades. Since [144] first introduced a stochastic-mapping solution to this problem, rapid and exciting progress has been made, resulting in many competing solutions, including both filtering and smoothing approaches. In particular, filtering methods such as the EKF and the UKF recursively estimate a state vector consisting of the current robot pose and the observed landmarks [11, 43, 155, 157]. Due to the fact that any (implicit or explicit) linearization-based filter marginalizes out the previous robot poses, it cannot relinearize the nonlinear system and measurement models at the past states, which may result in large linearization errors and thus degrade the filter's performance.

To better deal with nonlinearity, batch iterative optimization methods can be applied to the SLAM problem [36, 53–55, 95, 100, 150]. These methods, following the paradigm of bundle-adjustment (BA) algorithms originally developed in photogrammetry and computer vision [20–22, 52, 59, 153, 163], iteratively minimize a cost function involving the residuals of all the measurements, with respect to the entire robot trajectory and all landmarks (i.e., with no marginalization). These BA-based approaches exploit the sparsity of the measurement graph so as to speed up computation. However, for large-scale SLAM problems, a batch solution may be too computationally expensive to obtain in real time [87].

In order to reduce the computational complexity of BA, different approximate methods have been developed that either use a subset of the data to optimize over only few variables, or solve the BA problem only intermittently. Specifically, sliding-window filters (e.g., [67, 142]), compute a solution for a constant-size, sliding window of states (robot poses and landmark positions)

using only the measurements corresponding to that time interval. Similarly, keyframe-based approaches (e.g., [93, 94, 96]), perform batch optimization over only a subset of views/keyframes. On the other hand, incremental approaches to BA such as the iSAM algorithm [88] reduce computation by employing factorization-updating methods which allow reusing the information-matrix factorization available from previous time steps. Computationally demanding procedures, such as relinearization and batch factorization, are only performed intermittently. Alternatively, the iSAM2 algorithm [87] uses the Bayes tree data structure [86], which allows for fluid or just-in-time relinearization (i.e., relinearizing only when the linearization point significantly deviates from the current estimate), as well as partial variable reordering at every update (instead of only periodic batch reordering as in iSAM [88]). Nevertheless, incremental methods can also suffer from increased computational cost. For example, due to the accumulation of fill-in between periodic batch steps, iSAM's efficiency degrades with frequent loop closures (e.g., if the number of constraints is more than five times the number of poses as reported in [87]); while in iSAM2, since typically many variables are affected by every fluid relinearization, the complexity of the algorithm can be negatively impacted [87].

Even though both filtering and smoothing approaches have been widely used, to this date, very little is known about which conditions favor the use of one over the other. In particular, Strasdat et al. [149, 150] recently argued that BA is, in general, better than filtering in terms of accuracy and efficiency. However, their analysis focused exclusively on the restrictive scenario of "small-scale" visual SLAM where overlapping views of the same scene are assumed over a short trajectory (less than 16 camera poses in total) and without any loop closure. Clearly, based on this limiting case study, one cannot make inferences about the relative accuracy and efficiency of filtering and smoothing algorithms in more realistic SLAM scenarios (i.e., lengthy paths with varying number of visible landmarks and loop closure events).

Although such a general study is beyond the scope of this work, in this chapter we have compared the proposed OC-UKF and the state-of-the-art iSAM algorithm [88] in various SLAM scenarios, both in simulations and in real-world datasets. In particular, as shown in Sections 4.7 and 4.8, iSAM does *not* necessarily perform better than the proposed OC-UKF (in terms of estimation accuracy/consistency and computational cost). Specifically, while BA methods are certainly preferable in problems involving thousands of landmarks and few loop closures, filtering-based methods are still competitive in the case of sparser environments (e.g., tens to a few hundreds of landmarks), long-term operation, and frequent loop closures. This is due to the fact

that, in the latter scenario, the computational cost of smoothing methods will continuously increase with the length of the robot's path, while the runtime of filtering algorithms will remain bounded. It is worth noting that many applications of interest (e.g., a service robot operating inside a home for an extended time period) fall under the second category.

Since in this work, we primarily focus on the computational complexity and consistency of UKF-based SLAM, in what follows, we discuss in more detail the closely related work within this category.

4.2.1 UKF computational complexity

A number of researchers have applied the standard UKF to the SLAM problem (e.g., [29, 104, 110]). However, this requires computing the square root of the state covariance matrix at each time step, which has computational complexity *cubic* in the number of landmarks, and thus is not suitable for real-time operation in larger environments. To address this problem, Holmes et al. [62, 63] proposed the square-root UKF (SRUKF) for monocular visual SLAM, which has computational complexity *quadratic* both in the propagation and in the update phases. This approach offers a significant improvement in terms of computational complexity, at the cost of a considerably more complicated implementation. Additionally, as shown in [62, 63], the algorithm is an order of magnitude slower than the standard EKF, due to the need to carry out expensive numerical computations.

Andrade-Cetto et al. [6] presented a "hybrid" EKF/UKF algorithm, where the EKF is employed in the update phase, while the UKF is used during propagation for computing *only* the robot pose estimate and its covariance. The cross-correlation terms during propagation are handled in a fashion identical to the EKF. Even though this algorithm achieves computational complexity linear during propagation and quadratic during updates, the positive definiteness of the state covariance matrix cannot be guaranteed during propagation. Moreover, the use of the EKF for updates makes the approach vulnerable to large linearization errors.

In contrast to the aforementioned approaches, the proposed algorithm described in Section 4.4 employs the unscented transformation *both* in the propagation and update phases, is simple to implement, and attains computational complexity *linear* during propagation, and *quadratic* during update.

4.2.2 UKF consistency

The *consistency* of UKF-based SLAM has received limited attention in the literature. In [62, 63, 110], the consistency of the UKF was empirically examined, but, to the best of our knowledge, no theoretical analysis exists to date. On the other hand, the consistency of EKF-based SLAM has been studied in a number of publications [12, 66, 70, 72, 78, 79, 85]. In particular, in our recent work [66, 70, 72] (also see Chapter 2), we have presented an analytical study of this issue from the perspective of the observability properties of the EKF linearized system model.

In this chapter, we extend this analysis to the case of UKF-based SLAM. We analytically show that the implicit (statistical) linearization performed by the UKF results in a system model with "incorrect" observability properties, which is a fundamental cause of inconsistency. Moreover, we introduce the OC-UKF, which attains better performance than the standard UKF, by ensuring that the observability requirements on the filter's system model are satisfied. It is important to point out that, as compared to our previously-developed OC-EKF [72], the OC-UKF proposed in this chapter introduces a new paradigm for computing filter Jacobians. Specifically, the OC-EKF employs a derivative-based approach to find the filter Jacobians, and subsequently optimizes the selection of linearization points. In contrast, the OC-UKF uses statistical linearization and *directly* calculates the optimal (inferred) Jacobians by solving an observability-constrained optimization (linear-regression) problem.

4.3 LRKF and UKF

In this section, we present the UKF in the context of the LRKF. As shown in [106], the UKF is closely related to the LRKF (with its sample points chosen deterministically, instead of randomly in the LRKF) and it can be viewed as performing an implicit *statistical linearization* of the nonlinear propagation and update models. In what follows, we present the details of this linearization mechanism, which will be instrumental in the development of the quadratic-complexity UKF in Section 4.4.

4.3.1 Linear regression

The LRKF seeks to approximate a nonlinear function $\mathbf{y} = \mathbf{g}(\mathbf{x})$ with a linear model $\mathbf{y} \simeq \mathbf{A}\mathbf{x} + \mathbf{b}$, where $\mathbf{A}$ and $\mathbf{b}$ are the regression matrix and vector, respectively, and $\mathbf{e} \triangleq \mathbf{y} -$

$(\mathbf{A}\mathbf{x}+\mathbf{b})$ denotes the linearization error. Once this linear approximation is computed, the LRKF proceeds by applying the regular Kalman filter equations. In particular, in computing the linear approximation of $\mathbf{g}(\mathbf{x})$, it aims to minimize the expected value of the linearization error square:

$$\min_{\mathbf{A},\mathbf{b}} \int_{-\infty}^{+\infty} [\mathbf{y}-(\mathbf{A}\mathbf{x}+\mathbf{b})]^T [\mathbf{y}-(\mathbf{A}\mathbf{x}+\mathbf{b})]\, p(\mathbf{x}) d\mathbf{x} \tag{4.1}$$

where $p(\mathbf{x})$ is the probability density function (pdf) of the state $\mathbf{x}$. Due to the nonlinearity of $\mathbf{y}=\mathbf{g}(\mathbf{x})$, it is generally intractable to compute the optimal solution of this minimization problem in closed form. To solve this problem, the LRKF instead first selects $r+1$ weighted sample points, $\{\mathcal{X}_i, w_i\}_{i=0}^r$, so that their sample mean and covariance are equal to the mean and covariance of $\mathbf{x}$:[1]

$$\bar{\mathbf{x}} = \sum_{i=0}^{r} w_i \mathcal{X}_i \ = \mathbb{E}(\mathbf{x}) \tag{4.2}$$

$$\bar{\mathbf{P}}_{\mathbf{xx}} = \sum_{i=0}^{r} w_i \left(\mathcal{X}_i - \bar{\mathbf{x}}\right)\left(\mathcal{X}_i - \bar{\mathbf{x}}\right)^T = \mathbb{E}\left[(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}})^T\right] \tag{4.3}$$

where $\mathbb{E}(\cdot)$ denotes the expectation operator. Then, using the standard sample-based approximation $p(\mathbf{x}) \simeq \sum_{i=0}^{r} w_i \delta(\mathbf{x}-\mathcal{X}_i)$, where $\delta(\cdot)$ is the Dirac delta function, the linear regression problem (4.1) becomes:

$$\min_{\mathbf{A},\mathbf{b}} \sum_{i=0}^{r} w_i \left[\mathcal{Y}_i - (\mathbf{A}\mathcal{X}_i+\mathbf{b})\right]^T \left[\mathcal{Y}_i - (\mathbf{A}\mathcal{X}_i+\mathbf{b})\right] \tag{4.4}$$

where $\mathcal{Y}_i \triangleq \mathbf{g}(\mathcal{X}_i)$ are the regression points. We denote the linearization error corresponding to the sample point $\mathcal{X}_i$ by $\mathbf{e}_i \triangleq \mathcal{Y}_i - (\mathbf{A}\mathcal{X}_i+\mathbf{b})$. Note that the above cost function is identical to the one in [106], and hence the optimal solutions for $\mathbf{A}$ and $\mathbf{b}$ are given by [106]:

$$\mathbf{A} = \bar{\mathbf{P}}_{\mathbf{yx}} \bar{\mathbf{P}}_{\mathbf{xx}}^{-1}, \quad \mathbf{b} = \bar{\mathbf{y}} - \mathbf{A}\bar{\mathbf{x}} \tag{4.5}$$

[1] Throughout this chapter, $\bar{\mathbf{x}}$ and $\bar{\mathbf{P}}_{\mathbf{xx}}$ denote the sample mean and covariance of sample points $\mathcal{X}_i$, drawn from the pdf of the random variable $\mathbf{x}$. $\bar{\mathbf{P}}_{\mathbf{xy}}$ denotes the sample cross-correlation between the sets of samples $\mathcal{X}_i$ and $\mathcal{Y}_i$, drawn from the pdfs of the random variables $\mathbf{x}$ and $\mathbf{y}$, respectively.

where

$$\bar{\mathbf{y}} = \sum_{i=0}^{r} w_i \mathcal{Y}_i \tag{4.6}$$

$$\bar{\mathbf{P}}_{\mathbf{yx}} = \sum_{i=0}^{r} w_i \left(\mathcal{Y}_i - \bar{\mathbf{y}}\right)\left(\mathcal{X}_i - \bar{\mathbf{x}}\right)^T \tag{4.7}$$

$$\bar{\mathbf{P}}_{\mathbf{yy}} = \sum_{i=0}^{r} w_i \left(\mathcal{Y}_i - \bar{\mathbf{y}}\right)\left(\mathcal{Y}_i - \bar{\mathbf{y}}\right)^T \tag{4.8}$$

In addition, using (4.5), (4.7) and (4.8), the sample covariance of the linearization errors is computed by:

$$\bar{\mathbf{P}}_{\mathbf{ee}} = \sum_{i=0}^{r} w_i \mathbf{e}_i \mathbf{e}_i^T = \bar{\mathbf{P}}_{\mathbf{yy}} - \mathbf{A}\bar{\mathbf{P}}_{\mathbf{xx}}\mathbf{A}^T \tag{4.9}$$

During recursive estimation, the LRKF employs the above statistical linearization procedure to approximate the nonlinear process and measurement models. It is important to note that, in this case, the regression matrix $\mathbf{A}$ serves as an *inferred Jacobian* matrix, analogous to the Jacobian matrices in the EKF. The details are explained next.

4.3.2 LRKF propagation

During propagation, the LRKF approximates the nonlinear process model by a linear function:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{o}_k) \tag{4.10}$$

$$= \breve{\mathbf{\Phi}}_k \mathbf{x}_k + \breve{\mathbf{G}}_k \boldsymbol{o}_k + \mathbf{b}_k + \mathbf{e}_k \tag{4.11}$$

$$= \underbrace{\begin{bmatrix} \breve{\mathbf{\Phi}}_k & \breve{\mathbf{G}}_k \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{o}_k \end{bmatrix} + \mathbf{b}_k + \mathbf{e}_k \tag{4.12}$$

where $\mathbf{x}_\ell$ is the state vector at time-step $\ell \in \{k, k+1\}$, $\boldsymbol{o}_k = \boldsymbol{o}_{m_k} - \mathbf{w}_k$ is the control input (e.g., odometry), $\boldsymbol{o}_{m_k}$ is the corresponding measurement, and $\mathbf{w}_k$ is the process noise vector, assumed to be zero-mean white Gaussian, with covariance matrix $\mathbf{Q}_k$. The matrices $\breve{\mathbf{\Phi}}_k$ and $\breve{\mathbf{G}}_k$ can be viewed as inferred Jacobians, in an analogy to the corresponding Jacobians in the EKF. We hereafter use the symbol "$\breve{}$" to denote the inferred Jacobians.

In the LRKF propagation step, $r+1$ sample points $\{\mathcal{X}_i(k|k)\}_{i=0}^{r}$ are selected based on the *augmented* vector that comprises the filter state and the control input [see (4.12)]. The sample

mean and sample covariance of $\{\mathcal{X}_i(k|k)\}_{i=0}^r$ are thus chosen as:

$$\bar{\mathbf{x}}_{k|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \boldsymbol{o}_{m_k} \end{bmatrix}, \quad \bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k|k}} = \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} \tag{4.13}$$

Subsequently, the LRKF produces the regression points, $\{\mathcal{Y}_i(k+1|k) = \mathbf{f}(\mathcal{X}_i(k|k))\}_{i=0}^r$, by passing the sample points through the nonlinear process function (4.10). The sample mean, $\bar{\mathbf{y}}_{k+1|k}$, and sample covariance, $\bar{\mathbf{P}}_{\mathbf{y}\mathbf{y}_{k+1|k}}$, of the regression points $\mathcal{Y}_i$ are used as the mean, $\hat{\mathbf{x}}_{k+1|k}$, and covariance, $\mathbf{P}_{k+1|k}$, of the propagated state estimates, respectively, i.e.,

$$\hat{\mathbf{x}}_{k+1|k} = \bar{\mathbf{y}}_{k+1|k}\,, \quad \mathbf{P}_{k+1|k} = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{y}_{k+1|k}} \tag{4.14}$$

Moreover, the inferred Jacobian matrices, $\breve{\mathbf{\Phi}}_k$ and $\breve{\mathbf{G}}_k$, which will be needed later on, are given by [see (4.5) and (4.12)]:

$$\mathbf{A} = \begin{bmatrix} \breve{\mathbf{\Phi}}_k & \breve{\mathbf{G}}_k \end{bmatrix} = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_{k|k}} \bar{\mathbf{P}}^{-1}_{\mathbf{x}\mathbf{x}_{k|k}} \tag{4.15}$$

where $\bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_{k|k}}$ is computed as in (4.7). Substituting (4.5) in (4.11) and using (4.13), (4.14) and (4.15), we have:

$$\begin{aligned} \mathbf{x}_{k+1} &= \breve{\mathbf{\Phi}}_k \mathbf{x}_k + \breve{\mathbf{G}}_k \boldsymbol{o}_k + \bar{\mathbf{y}}_{k+1|k} - \mathbf{A}\bar{\mathbf{x}}_{k|k} + \mathbf{e}_k \\ &= \breve{\mathbf{\Phi}}_k \mathbf{x}_k + \breve{\mathbf{G}}_k \boldsymbol{o}_k + \hat{\mathbf{x}}_{k+1|k} - \begin{bmatrix} \breve{\mathbf{\Phi}}_k & \breve{\mathbf{G}}_k \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \boldsymbol{o}_{m_k} \end{bmatrix} + \mathbf{e}_k \\ \Rightarrow\ \tilde{\mathbf{x}}_{k+1|k} &= \breve{\mathbf{\Phi}}_k \tilde{\mathbf{x}}_k + \breve{\mathbf{G}}_k \mathbf{w}_k + \mathbf{e}_k \end{aligned} \tag{4.16}$$

This last equation describes the linearized (based on regression) error-state propagation model used by the LRKF.

4.3.3 LRKF update

During update, the LRKF employs statistical linearization to approximate the nonlinear measurement function:

$$\mathbf{z}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1} \tag{4.17}$$

$$= \breve{\mathbf{H}}_{k+1} \mathbf{x}_{k+1} + \mathbf{b}'_{k+1} + \mathbf{e}'_{k+1} + \mathbf{v}_{k+1} \tag{4.18}$$

where $\mathbf{z}_{k+1}$ is the measurement and $\mathbf{v}_{k+1}$ is the zero-mean white Gaussian measurement noise, with covariance matrix $\mathbf{R}_{k+1}$. A set of $r+1$ sample points, $\{\mathcal{X}_i(k+1|k)\}_{i=0}^{r}$, are selected, whose sample mean and sample covariance are equal to $\hat{\mathbf{x}}_{k+1|k}$ and $\mathbf{P}_{k+1|k}$, respectively, i.e.,

$$\bar{\mathbf{x}}_{k+1|k} = \hat{\mathbf{x}}_{k+1|k} \ , \quad \bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k+1|k}} = \mathbf{P}_{k+1|k} \tag{4.19}$$

We pass these sample points through the nonlinear measurement function in (4.17), to obtain the regression points, $\{\mathcal{Z}_i(k+1|k) = \mathbf{h}(\mathcal{X}_i(k+1|k))\}_{i=0}^{r}$. The regression matrix (i.e., inferred measurement Jacobian) $\breve{\mathbf{H}}_{k+1}$is computed by [see (4.5)]:

$$\breve{\mathbf{H}}_{k+1} = \bar{\mathbf{P}}_{\mathbf{z}\mathbf{x}_{k+1|k}} \bar{\mathbf{P}}^{-1}_{\mathbf{x}\mathbf{x}_{k+1|k}} \tag{4.20}$$

where $\bar{\mathbf{P}}_{\mathbf{z}\mathbf{x}_{k+1|k}}$ is computed as in (4.7). Subsequently, the state and covariance are updated using the EKF update equations:

$$\mathbf{S}_{k+1} = \bar{\mathbf{P}}_{\mathbf{z}\mathbf{z}_{k+1|k}} + \mathbf{R}_{k+1} \tag{4.21}$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \breve{\mathbf{H}}^T_{k+1} \mathbf{S}^{-1}_{k+1} \tag{4.22}$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}(\mathbf{z}_{k+1} - \bar{\mathbf{z}}_{k+1|k}) \tag{4.23}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1} \mathbf{S}_{k+1} \mathbf{K}^T_{k+1} \tag{4.24}$$

where $\bar{\mathbf{z}}_{k+1|k}$ and $\bar{\mathbf{P}}_{\mathbf{z}\mathbf{z}_{k+1|k}}$ are computed from (4.6) and (4.8), respectively.

4.3.4 UKF sampling

In contrast to the LRKF [106], where the sample points are drawn randomly, in the UKF, $r+1 = 2n+1$ so-called sigma points $\mathcal{X}_i$ are *deterministically* chosen along with their weights w_i, $i = 1, \ldots, n$, according to the following equations [84]:

$$\begin{aligned}
\mathcal{X}_0(\ell|k) &= \bar{\mathbf{x}}_{\ell|k} \ , \ w_0 = \frac{2\kappa}{2(n+\kappa)} \\
\mathcal{X}_i(\ell|k) &= \bar{\mathbf{x}}_{\ell|k} + \left[\sqrt{(n+\kappa)\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{\ell|k}}}\right]_i \ , \ w_i = \frac{1}{2(n+\kappa)} \\
\mathcal{X}_{i+n}(\ell|k) &= \bar{\mathbf{x}}_{\ell|k} - \left[\sqrt{(n+\kappa)\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{\ell|k}}}\right]_i \ , \ w_{i+n} = \frac{1}{2(n+\kappa)}
\end{aligned} \tag{4.25}$$

where n is the dimension of $\bar{\mathbf{x}}_{\ell|k}$ [see (4.13) and (4.19)], $\left[\sqrt{(n+\kappa)\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{\ell|k}}}\right]_i$ is the i-th column of the matrix $\sqrt{(n+\kappa)\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{\ell|k}}}$, $\ell \in \{k, k+1\}$, and κ is a design parameter in the selection of the sigma points, usually chosen so that $n + \kappa = 3$. This set of sigma points captures the moments of the underlying distribution up to the third-order for the Gaussian case [84].

4.4 Quadratic-Complexity UKF SLAM

In this section, we show how the computational cost of the UKF, when applied to the SLAM problem, can be reduced. In particular, in this chapter we focus on 2D SLAM, in which the state vector consists of the robot pose (position and orientation) and the positions of M landmarks:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{R_k}^T & \mathbf{p}_L^T \end{bmatrix}^T \tag{4.26}$$

$$= \begin{bmatrix} \mathbf{x}_{R_k}^T & \mathbf{p}_{L_1}^T & \cdots & \mathbf{p}_{L_M}^T \end{bmatrix}^T \tag{4.27}$$

where $\mathbf{x}_{R_k} \triangleq \begin{bmatrix} \mathbf{p}_{R_k}^T & \phi_{R_k} \end{bmatrix}^T$ denotes the robot pose (position and orientation), and $\mathbf{p}_{L_i}$ ($i = 1 \ldots M$) is the position of the i-th landmark.

In the UKF algorithm presented in the preceding section, the main bottleneck is the computation of the square root of the covariance matrix [see (4.25)], which has complexity $\mathcal{O}(M^3)$. Clearly, in a scenario where a large number of landmarks are included in the state vector, carrying out this operation during each propagation and update would incur an unacceptable computational burden. To address this problem, we here propose a new sampling scheme for the UKF, which has computational cost $\mathcal{O}(1)$, and hence reduces the complexity of the propagation and update steps to linear and quadratic, respectively. The derivation of this sampling scheme is based on the observation that, during SLAM, only a *small subset* of the state vector appears in the nonlinear propagation and measurement models. In particular, during propagation only the robot state changes, while at each update, every measurement involves only the robot pose and one observed landmark.[2] To take advantage of this important property, we employ the following lemma:

Lemma 4.4.1. *Consider a nonlinear function* $\mathbf{y} = \mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}_1)$*, where only the state entries* $\mathbf{x}_1$ *of the vector* $\mathbf{x}$ *partitioned as* $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$ *appear in* $\mathbf{g}(\mathbf{x})$*. Moreover, consider the regression matrix* $\mathbf{A}$ *of the linear regression problem* (4.4) *accordingly partitioned as* $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{bmatrix}$*, i.e.,*

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} + \mathbf{e} = \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 + \mathbf{b} + \mathbf{e} \tag{4.28}$$

[2] When more than one landmarks are detected concurrently, their measurements can be processed sequentially, given that the measurement noise in different observations is independent.

Then, the optimal solution to (4.4) *is:*

$$\mathbf{A}_1 = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1}\mathbf{P}^{-1}_{\mathbf{x_1}\mathbf{x_1}}, \quad \mathbf{A}_2 = \mathbf{0}, \quad \mathbf{b} = \bar{\mathbf{y}} - \mathbf{A}_1\hat{\mathbf{x}}_1 \tag{4.29}$$

Proof. See Appendix C.1. □

This lemma shows that, in order to minimize the expected squared error of the statistical linearization (4.4), it suffices to draw sample points from the pdf of $\mathbf{x}_1$. As mentioned before, in SLAM the number of states participating in the nonlinear process and measurement models is constant. Thus, we can reduce the cost of UKF sampling to $\mathcal{O}(1)$ by applying the unscented transformation only to the pertinent state entries, instead of sampling over the whole state. Compared to the EKF-SLAM, the proposed UKF-SLAM only incurs a small computational overhead (for computing the square roots of constant-size matrices), and has computational complexity of the same order. In the following, we present in detail this new sampling strategy used in the UKF-SLAM. We stress again that apart from the particular problem of SLAM treated in this chapter, this new UKF sampling scheme is applicable to *any* problem where the measurements are of lower dimension than the state.

4.4.1 Propagation

During propagation, only the robot pose and the control input (odometry) participate in the process model [see (4.10)]. Therefore, we are able to reduce the computational complexity by applying the unscented transformation only to the part of the state comprising the robot pose and the control input, instead of the full state vector. The resulting Jacobians are then used for efficiently propagating the covariance matrix corresponding to the entire state.

We start by drawing the sigma points $\mathcal{X}_i(k|k)$ based on the vector with the following mean and covariance [see (4.13)]:

$$\bar{\mathbf{x}}_{k|k} = \begin{bmatrix} \hat{\mathbf{x}}_{R_{k|k}} \\ \boldsymbol{o}_{m_k} \end{bmatrix}, \quad \bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k|k}} = \begin{bmatrix} \mathbf{P}_{RR_{k|k}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} \tag{4.30}$$

where $\mathbf{P}_{RR_{k|k}}$ is the covariance matrix corresponding to the robot pose, obtained by partitioning the state covariance matrix as follows [see (4.26)]:

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{RR_{k|k}} & \mathbf{P}_{RL_{k|k}} \\ \mathbf{P}^T_{RL_{k|k}} & \mathbf{P}_{LL_{k|k}} \end{bmatrix} \tag{4.31}$$

Note that the vector $\bar{\mathbf{x}}_{k|k}$ in (4.30) is of dimension $n = 5$ (assuming that the odometry measurement $\boldsymbol{o}_{m_k}$ is two-dimensional), and thus the computational cost of computing the sigma points is very low.

Subsequently, we transform the sigma points, $\{\mathcal{X}_i(k|k)\}_{i=0}^{10}$, using the process model (4.10), to obtain the regression points of the propagated robot pose, $\{\mathcal{Y}_i(k+1|k) = \mathbf{f}(\mathcal{X}_i(k|k))\}_{i=0}^{10}$. This enables us to compute the mean, $\hat{\mathbf{x}}_{R_{k+1|k}} = \bar{\mathbf{y}}_{k+1|k}$, and covariance, $\mathbf{P}_{RR_{k+1|k}} = \bar{\mathbf{P}}_{\mathbf{yy}_{k+1|k}}$, of the propagated robot pose, in the same way as in the standard LRKF/UKF [see (4.14)]. Moreover, we can evaluate the *inferred* robot state and odometry Jacobians as [see (4.29) and (4.15)]:

$$\mathbf{A}_1 = \bar{\mathbf{P}}_{\mathbf{yx}_{k|k}} \bar{\mathbf{P}}^{-1}_{\mathbf{xx}_{k|k}} = \begin{bmatrix} \breve{\mathbf{\Phi}}_{R_k} & \breve{\mathbf{G}}_{R_k} \end{bmatrix} \tag{4.32}$$

while $\mathbf{A}_2 = \mathbf{0}$.

Next, using (4.16), we compute the propagated cross-correlation between the robot and the landmarks as follows:

$$\begin{aligned} \mathbf{P}_{RL_{k+1|k}} &= \mathbb{E}\left[\tilde{\mathbf{x}}_{R_{k+1|k}} \tilde{\mathbf{p}}^T_{L_{k|k}}\right] \\ &= \mathbb{E}\left[\left(\breve{\mathbf{\Phi}}_{R_k} \tilde{\mathbf{x}}_{R_{k|k}} + \breve{\mathbf{G}}_{R_k} \mathbf{w}_k + \mathbf{e}_k\right) \tilde{\mathbf{p}}^T_{L_{k|k}}\right] \\ &= \breve{\mathbf{\Phi}}_{R_k} \mathbf{P}_{RL_{k|k}} \end{aligned} \tag{4.33}$$

Thus, the propagated state covariance matrix is given by:

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \bar{\mathbf{P}}_{\mathbf{yy}_{k+1|k}} & \breve{\mathbf{\Phi}}_{R_k} \mathbf{P}_{RL_{k|k}} \\ \mathbf{P}^T_{RL_{k|k}} \breve{\mathbf{\Phi}}^T_{R_k} & \mathbf{P}_{LL_{k|k}} \end{bmatrix} \tag{4.34}$$

which is evaluated with cost only *linear* in the size of the state vector, similarly to the EKF.

The matrix $\breve{\mathbf{\Phi}}_{R_k}$ derived in (4.32) is the inferred propagation Jacobian for the robot state. To compute the inferred Jacobian matrix for the entire SLAM state vector, which will be useful for our ensuing analysis, we use (4.9), (4.30) and (4.32) to write

$$\begin{aligned} \bar{\mathbf{P}}_{\mathbf{yy}_{k+1|k}} =& \mathbf{A}_1 \bar{\mathbf{P}}_{\mathbf{xx}_{k|k}} \mathbf{A}_1^T + \bar{\mathbf{P}}_{\mathbf{ee}_k} \\ =& \breve{\mathbf{\Phi}}_{R_k} \mathbf{P}_{RR_{k|k}} \breve{\mathbf{\Phi}}^T_{R_k} + \breve{\mathbf{G}}_{R_k} \mathbf{Q}_k \breve{\mathbf{G}}^T_{R_k} + \bar{\mathbf{P}}_{\mathbf{ee}_k} \end{aligned} \tag{4.35}$$

and therefore, (4.34) can equivalently be written as:

$$\mathbf{P}_{k+1|k} = \breve{\mathbf{\Phi}}_k \mathbf{P}_{k|k} \breve{\mathbf{\Phi}}^T_k + \mathbf{Q}^*_k \tag{4.36}$$

where

$$\mathbf{Q}_k^* = \begin{bmatrix} \breve{\mathbf{G}}_{R_k}\mathbf{Q}_k\breve{\mathbf{G}}_{R_k}^T + \bar{\mathbf{P}}_{\mathbf{e}\mathbf{e}_k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{4.37}$$

$$\breve{\mathbf{\Phi}}_k = \begin{bmatrix} \breve{\mathbf{\Phi}}_{R_k} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{2M} \end{bmatrix} \tag{4.38}$$

In the above expression, $\breve{\mathbf{\Phi}}_k$ is the inferred propagation Jacobian matrix for the SLAM state vector comprising the robot pose and the landmark positions.

4.4.2 Update

Any measurement used for updating involves only the robot pose and the position of the observed landmark. Therefore, we can apply the unscented transformation only to this subset of states so as to reduce the computational cost. In particular, assume that the j-th landmark, L_j, is observed at time-step $k+1$. Then, the set of sigma points $\{\mathcal{X}_i(k+1|k)\}_{i=0}^{10}$ are drawn from a distribution with the following mean and covariance:

$$\bar{\mathbf{x}}_{k+1|k} = \begin{bmatrix} \hat{\mathbf{x}}_{R_{k+1|k}} \\ \hat{\mathbf{p}}_{L_{j,k+1|k}} \end{bmatrix}, \; \bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k+1|k}} = \begin{bmatrix} \mathbf{P}_{RR_{k+1|k}} & \mathbf{P}_{RL_{j,k+1|k}} \\ \mathbf{P}_{L_jR_{k+1|k}} & \mathbf{P}_{L_jL_{j,k+1|k}} \end{bmatrix} \tag{4.39}$$

where $\mathbf{P}_{RR_{k+1|k}}$ and $\mathbf{P}_{L_jL_{j,k+1|k}}$ are the covariance matrices of the robot and the landmark, respectively, while $\mathbf{P}_{RL_{j,k+1|k}} = \mathbf{P}_{L_jR_{k+1|k}}^T$ is the corresponding cross-correlation matrix, obtained from the following partitioning of the state covariance matrix:

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \mathbf{P}_{RR_{k+1|k}} & \cdots & \mathbf{P}_{RL_{j,k+1|k}} & \cdots & \mathbf{P}_{RL_{M,k+1|k}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{P}_{L_jR_{k+1|k}} & \cdots & \mathbf{P}_{L_jL_{j,k+1|k}} & \cdots & \mathbf{P}_{L_jL_{M,k+1|k}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{P}_{L_MR_{k+1|k}} & \cdots & \mathbf{P}_{L_ML_{j,k+1|k}} & \cdots & \mathbf{P}_{L_ML_{M,k+1|k}} \end{bmatrix}$$

Note that the matrix used for generating the sigma points has constant size [see (4.39)], regardless of the number of landmarks in the state vector.

Once the set of sigma points are generated, the linear regression of the LRKF update (see Section 4.3.3) is applied to obtain the *inferred* measurement Jacobian for the pertinent

states [see (4.20) and (4.29)]:

$$\mathbf{A}_1 = \bar{\mathbf{P}}_{\mathbf{z}\mathbf{x}_{k+1|k}} \bar{\mathbf{P}}^{-1}_{\mathbf{x}\mathbf{x}_{k+1|k}} = \begin{bmatrix} \breve{\mathbf{H}}_{R_{k+1}} & \breve{\mathbf{H}}_{L_{j,k+1}} \end{bmatrix} \tag{4.40}$$

where the submatrix $\breve{\mathbf{H}}_{R_{k+1}}$ corresponds to the robot pose, while $\breve{\mathbf{H}}_{L_{j,k+1}}$ corresponds to the j-th landmark. To construct the inferred measurement Jacobian for the entire state vector, we note that according to the new sampling scheme the unscented transformation is not applied to the landmarks that are not currently observed (their regression matrices are zero according to Lemma 4.4.1). Therefore, the *inferred* measurement Jacobian for the entire SLAM state vector is:

$$\breve{\mathbf{H}}_{k+1} = \begin{bmatrix} \breve{\mathbf{H}}_{R_{k+1}} & \mathbf{0} & \cdots & \mathbf{0} & \breve{\mathbf{H}}_{L_{j,k+1}} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \tag{4.41}$$

Once this matrix is available, (4.21)-(4.24) are applied to update the state estimate and covariance in the UKF. It is important to point out that the computational cost of the proposed UKF update equations is dominated by the covariance update (4.24), and hence is quadratic in the number of landmarks, similarly to the EKF.

4.4.3 Landmark initialization

Suppose that the j-th landmark, L_j, is first observed at time-step k_o. The corresponding measurement is given by:[3]

$$\mathbf{z}_{k_o} = \mathbf{h}(\mathbf{x}_{R_{k_o}}, \mathbf{p}_{L_j}) + \mathbf{v}_{k_o} = \mathbf{z}^*_{k_o} + \mathbf{v}_{k_o} \tag{4.42}$$

By solving $\mathbf{z}^*_{k_o} = \mathbf{h}(\mathbf{x}_{R_{k_o}}, \mathbf{p}_{L_j})$ for $\mathbf{p}_{L_j}$, we can express the landmark position as a (generally nonlinear) function of the robot pose and the noiseless measurement:

$$\mathbf{p}_{L_j} = \mathbf{g}(\mathbf{x}_{R_{k_o}}, \mathbf{z}^*_{k_o}) \tag{4.43}$$

[3] To preserve the clarity of presentation, we consider the case where a single measurement suffices to initialize the landmark. This includes the distance-bearing measurement model, commonly used in practice. However, this is not a necessary assumption and our analysis can be extended to the case where multiple measurements at different time steps are needed to initialize the new landmark (e.g., bearing-only or distance-only measurements).

In order to carry out the landmark initialization in the LRKF/UKF framework, we approximate this nonlinear initialization function by a linear function:

$$\mathbf{p}_{L_j} = \mathbf{A}_{\mathbf{x}}\mathbf{x}_{R_{k_o}} + \mathbf{A}_{\mathbf{z}}\mathbf{z}^*_{k_o} + \mathbf{b}''_{k_o} + \mathbf{e}''_{k_o} \tag{4.44}$$

$$= \underbrace{\begin{bmatrix}\mathbf{A}_{\mathbf{x}} & \mathbf{A}_{\mathbf{z}}\end{bmatrix}}_{\mathbf{A}_1} \begin{bmatrix}\mathbf{x}_{R_{k_o}} \\ \mathbf{z}^*_{k_o}\end{bmatrix} + \mathbf{b}''_{k_o} + \mathbf{e}''_{k_o} \tag{4.45}$$

where $\mathbf{A}_{\mathbf{x}}$ and $\mathbf{A}_{\mathbf{z}}$ are the regression matrices corresponding to the robot pose and the measurement, respectively. These matrices are computed by statistical linearization, similarly to the cases of propagation and update.

Specifically, it becomes clear from (4.43) that only the robot pose and the measurement of the newly detected landmark are involved in the initialization process. Therefore, we can apply the result of Lemma 4.4.1 to draw the sigma points based on the vector with the following mean and covariance:

$$\bar{\mathbf{x}}_{k_o|k_o-1} = \begin{bmatrix}\hat{\mathbf{x}}_{R_{k_o|k_o-1}} \\ \mathbf{z}_{k_o}\end{bmatrix}, \quad \bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k_o|k_o-1}} = \begin{bmatrix}\mathbf{P}_{RR_{k_o|k_o-1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{k_o}\end{bmatrix} \tag{4.46}$$

Suppose the measurement vector is of dimension m. Then, the UKF will choose $r+1 = 2\times(3+m)+1$ sigma points, $\{\mathcal{X}_i(k_o|k_o-1)\}_{i=0}^r$, and transform them through the nonlinear initialization model (4.43) to obtain the regression points of the new landmark position $\{\mathcal{Y}_i(k_o|k_o-1) = \mathbf{g}(\mathcal{X}_i(k_o|k_o-1))\}_{i=0}^r$. The sample mean of the regression points is used to initialize the new landmark position:

$$\hat{\mathbf{p}}_{L_{j,k_o|k_o}} = \bar{\mathbf{y}}_{k_o|k_o-1} \tag{4.47}$$

In order to compute the covariance matrix of the augmented state vector comprising the robot pose, the previously initialized landmarks, and the new landmark, we first note that the regression matrix in (4.45) is [see (4.29)]:

$$\mathbf{A}_1 = \begin{bmatrix}\mathbf{A}_{\mathbf{x}} & \mathbf{A}_{\mathbf{z}}\end{bmatrix} = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_{k_o|k_o-1}}\bar{\mathbf{P}}^{-1}_{\mathbf{x}\mathbf{x}_{k_o|k_o-1}} \tag{4.48}$$

Subsequently, using (4.44), (4.46), (4.47), and (4.5), we compute the error in the posterior estimate for the position of the j-th landmark:

$$\begin{aligned}\tilde{\mathbf{p}}_{L_{j,k_o|k_o}} &= \mathbf{p}_{L_j} - \hat{\mathbf{p}}_{L_{j,k_o|k_o}} \\ &= \mathbf{A}_{\mathbf{x}}\mathbf{x}_{R_{k_o}} + \mathbf{A}_{\mathbf{z}}\mathbf{z}^*_{k_o} + \mathbf{b}''_{k_o} + \mathbf{e}''_{k_o} - \mathbf{A}_{\mathbf{x}}\hat{\mathbf{x}}_{R_{k_o|k_o-1}} - \mathbf{A}_{\mathbf{z}}\mathbf{z}_{k_o} - \mathbf{b}''_{k_o} \\ &= \mathbf{A}_{\mathbf{x}}\tilde{\mathbf{x}}_{R_{k_o|k_o-1}} + \mathbf{A}_{\mathbf{z}}\mathbf{v}_{k_o} + \mathbf{e}''_{k_o}\end{aligned} \tag{4.49}$$

Based on (4.49), the cross-correlation terms between the new landmark and the robot and the old landmarks are given by:

$$\mathbf{P}_{L_j R_{k_o|k_o}} = \mathbb{E}\left[\tilde{\mathbf{p}}_{L_{j,k_o|k_o}} \tilde{\mathbf{x}}^T_{R_{k_o|k_o-1}}\right] = \mathbf{A}_{\mathbf{x}} \mathbf{P}_{R R_{k_o|k_o-1}} \tag{4.50}$$

$$\mathbf{P}_{L_j L_{j',k_o|k_o}} = \mathbb{E}\left[\tilde{\mathbf{p}}_{L_{j,k_o|k_o}} \tilde{\mathbf{p}}^T_{L_{j',k_o|k_o-1}}\right] = \mathbf{A}_{\mathbf{x}} \mathbf{P}_{R L_{j',k_o|k_o-1}} \tag{4.51}$$

for $j' = 1, \ldots, M$ and $j' \neq j$. Hence, the covariance matrix of the augmented state vector becomes:

$$\mathbf{P}_{k_o|k_o} = \begin{bmatrix} \mathbf{P}_{k_o|k_o-1} & \mathbf{P}_{k_o|k_o-1} \begin{bmatrix} \mathbf{A}^T_{\mathbf{x}} \\ \mathbf{0}_{2M\times 2} \end{bmatrix} \\ \begin{bmatrix} \mathbf{A}_{\mathbf{x}} & \mathbf{0}_{2\times 2M} \end{bmatrix} \mathbf{P}_{k_o|k_o-1} & \mathbf{P}_{L_j L_{j,k_o|k_o}} \end{bmatrix} \tag{4.52}$$

where $\mathbf{P}_{L_j L_{j,k_o|k_o}} = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{y}_{k_o|k_o-1}}$ is the sample covariance of the set of the regression points $\{\mathcal{Y}_i(k_o|k_o-1)\}_{i=0}^r$. Note that the computational complexity of the UKF landmark initialization is linear in the number of landmarks, which is of the same order as in the EKF.

For our derivations in the following sections, it will be necessary to compute the inferred measurement Jacobian matrices, $\breve{\mathbf{H}}_{R_{k_o}}$ and $\breve{\mathbf{H}}_{L_{j,k_o}}$, which correspond to the measurement used for initializing the landmark. For this purpose, by solving (4.44) for $\mathbf{z}^*_{k_o}$ and then substituting in (4.42), we have:

$$\mathbf{z}_{k_o} = -\mathbf{A}_{\mathbf{z}}^{-1} \mathbf{A}_{\mathbf{x}} \mathbf{x}_{R_{k_o}} + \mathbf{A}_{\mathbf{z}}^{-1} \mathbf{p}_{L_j} - \mathbf{A}_{\mathbf{z}}^{-1} \mathbf{b}''_{k_o} - \mathbf{A}_{\mathbf{z}}^{-1} \mathbf{e}''_{k_o} + \mathbf{v}_{k_o} \tag{4.53}$$

We thus conclude that the *inferred* measurement Jacobians corresponding to this measurement are:

$$\breve{\mathbf{H}}_{R_{k_o}} = -\mathbf{A}_{\mathbf{z}}^{-1} \mathbf{A}_{\mathbf{x}} \,, \quad \breve{\mathbf{H}}_{L_{j,k_o}} = \mathbf{A}_{\mathbf{z}}^{-1} \tag{4.54}$$

4.5 SLAM observability analysis

As discussed in Section 4.3, the UKF carries out recursive state estimation based on a linear approximation (i.e., using sigma points) of the nonlinear system model. In this section, we examine the *observability* properties of the UKF linear-regression-based system model, since they can affect the filter's performance. To the best of our knowledge, no such analysis has appeared in the literature prior to [71].

4.5.1 Background

Our motivation arises from our previous work [66, 70, 72], where it was shown that the observability properties of the EKF's linearized system model greatly impact the filter's consistency in SLAM. Specifically, we have proven in [66, 70, 72] that the system model of an *ideal* EKF, whose Jacobians are evaluated at the *true* state, has *3 unobservable degrees of freedom (d.o.f.)*. These correspond to the global position and orientation, and match the unobservable directions of the underlying *nonlinear* SLAM system [72, 105]. Moreover, it was shown that the ideal EKF exhibits excellent performance in terms of consistency. By contrast, the system model of the (standard) EKF, which uses the *current* state estimates for computing the Jacobians, has *only 2 unobservable d.o.f.*, corresponding to the global position. As a result, the standard EKF becomes inconsistent since it acquires non-existent information along the direction of the global orientation. Based on this analysis, in [66, 70], we derived the First-Estimates Jacobian (FEJ)-EKF, which, by evaluating the Jacobians at the first available state estimates, achieves the desired observability properties (i.e., its system model has *3 unobservable d.o.f.*). However, the first state estimates may be inaccurate and result in large linearization errors, thus degrading the filter's performance. To improve the FEJ-EKF, in [72], we developed the Observability-Constrained (OC)-EKF which instead selects linearization points that not only ensure the linearized system model has the *correct number of unobservable d.o.f.*, but also *minimize the linearization errors*. As a result, the OC-EKF attains consistency better than that of the FEJ-EKF and comparable to that of the ideal EKF.

In this work, we adopt an analogous approach where we first examine the observability properties of the UKF-SLAM system model and compare them to those of the underlying nonlinear SLAM system. Based on this analysis, we introduce an efficient algorithm for computing the appropriate *inferred* measurement Jacobians that preserve the dimensions of the unobservable subspace, thus improving consistency.

4.5.2 UKF-SLAM observability

To examine the observability properties of the UKF-SLAM system model, we form the observability matrix [112] for the time interval $[k_o, k_o + k]$ as follows:

$$\mathbf{M} = \begin{bmatrix} \breve{\mathbf{H}}_{k_o} \\ \breve{\mathbf{H}}_{k_o+1}\breve{\mathbf{\Phi}}_{k_o} \\ \vdots \\ \breve{\mathbf{H}}_{k_o+k}\breve{\mathbf{\Phi}}_{k_o+k-1}\cdots\breve{\mathbf{\Phi}}_{k_o} \end{bmatrix} \tag{4.55}$$

where the inferred measurement Jacobian, $\breve{\mathbf{H}}_{k_o+\ell}$, $\ell \in \{0, \ldots, k\}$, and inferred state propagation Jacobian, $\breve{\mathbf{\Phi}}_{k_o+\ell-1}$, $\ell \in \{1, \ldots, k\}$, are computed based on the UKF regression matrices [see (4.32), (4.38), (4.40), and (4.41)].

Since the UKF approximates the nonlinear SLAM model by a regression-based linearized system [see (4.12) and (4.18)], it is desirable that its observability properties match those of the underlying nonlinear system, and thus those of the ideal EKF. That is, the UKF-SLAM system model should have 3 unobservable d.o.f., or equivalently its observability matrix, $\mathbf{M}$, should have a nullspace of dimension 3.

However, this is generally *not* the case. In fact, when numerically computing the dimension of the nullspace of $\mathbf{M}$, we find that it is 3 only at time-step k_o, when a landmark is initialized. At that time, the observability matrix comprises only the first inferred measurement Jacobian, i.e., $\mathbf{M} = \breve{\mathbf{H}}_{k_o}$, which is a 2×5 matrix and thus generally has a nullspace of dimension 3. Later on and as more measurements become available, the dimension of the nullspace of the observability matrix decreases fast. Typically, the observability matrix $\mathbf{M}$ becomes full-rank after two time steps of consecutive observations.

A full-rank observability matrix indicates that the linear-regression-based system model employed by the UKF is observable, which contradicts the observability analysis of the nonlinear SLAM system [72, 105]. In practice, this implies that the UKF obtains "spurious" information, in all directions of the state space, even in directions where no information is available, such as the global position and orientation. This, in turn, leads to an unjustified reduction of the state estimates' covariance matrix, which cannot be compensated for by the noise covariance increase that the UKF uses to account for linearization errors [see (4.36)]. As shown in the simulation and experimental results in Sections 4.7 and 4.8, the inconsistency due to the mismatch between the observability properties of the UKF linear-regression-based system model and the

nonlinear (or equivalently the ideal EKF) system model, causes a significant degradation in the filter's performance.

4.6 Observability-Constrained (OC)-UKF SLAM

In this section, we introduce a novel OC-UKF algorithm that employs a linear-regression-based system model with observability properties similar to those of the underlying nonlinear SLAM system. Specifically, we construct the "inferred" Jacobians of the UKF in such a way that the resulting system model has an *unobservable subspace of dimension 3*.

In particular, the propagation phase of the OC-UKF is identical to that of the standard UKF. The difference arises in the update phase, where, instead of employing the unconstrained minimization of (4.4) for computing the regression matrix, we formulate a *constrained* minimization problem that enforces the desired observability properties. Specifically, if the first landmark was observed at time-step k_o, we require that [see (4.55)]:

$$\mathbf{M}\mathbf{N} = \mathbf{0} \;\Leftrightarrow \tag{4.56}$$

$$\begin{cases} \breve{\mathbf{H}}_{k_o}\mathbf{N} = \mathbf{0}, \;\; \text{for } \ell = 0 \\ \breve{\mathbf{H}}_{k_o+\ell}\breve{\mathbf{\Phi}}_{k_o+\ell-1}\cdots\breve{\mathbf{\Phi}}_{k_o}\mathbf{N} = \mathbf{0}, \;\; \text{for } \ell > 0 \end{cases} \tag{4.57}$$

In the above expressions, $\mathbf{N}$ is a $(3+2M)\times 3$ matrix, whose columns span the desired nullspace. These constraints ensure that all the block rows of the observability matrix $\mathbf{M}$ (4.55) have the *same* nullspace, which coincides with the *unobservable subspace* of the filter's system model. By ensuring that its inferred system model has an unobservable subspace of dimension 3, the OC-UKF avoids the infusion of erroneous information, and is empirically shown to attain significantly improved consistency (see Sections 4.7 and 4.8).

In what follows, we show how the nullspace matrix $\mathbf{N}$ is determined, and based on that, we compute the inferred measurement Jacobians.

4.6.1 Computing the nullspace matrix $\mathbf{N}$

Consider the following partitioning of the matrix $\mathbf{N}$:

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_R^T & \mathbf{N}_{L_1}^T & \cdots & \mathbf{N}_{L_M}^T \end{bmatrix}^T \tag{4.58}$$

where $\mathbf{N}_R$ is a 3×3 submatrix corresponding to the robot pose, and $\mathbf{N}_{L_i}$, $i = 1, \ldots, M$, are 2×3 submatrices corresponding to the i-th landmark. It is important to note that landmarks are typically observed and initialized at different time instants, and hence the number of submatrices comprising $\mathbf{N}$ will increase over time, as new landmarks are included into the state vector.

Initialization of the first landmark

When the first landmark is initialized at time-step k_o, we choose $\mathbf{N}$ to be a matrix whose columns span the nullspace of the 2×5 inferred Jacobian $\breve{\mathbf{H}}_{k_o} = \begin{bmatrix} \breve{\mathbf{H}}_{R_{k_o}} & \breve{\mathbf{H}}_{L_{1,k_o}} \end{bmatrix}$ [see (4.54)], i.e.,

$$\text{null}(\breve{\mathbf{H}}_{k_o}) = \underset{\text{col.}}{\text{span}}(\mathbf{N}) \Rightarrow \begin{bmatrix} \breve{\mathbf{H}}_{R_{k_o}} & \breve{\mathbf{H}}_{L_{1,k_o}} \end{bmatrix} \begin{bmatrix} \mathbf{N}_R \\ \mathbf{N}_{L_1} \end{bmatrix} = \mathbf{0} \tag{4.59}$$

Thus, $\mathbf{N}$ can be readily computed via the singular value decomposition (SVD) of $\breve{\mathbf{H}}_{k_o}$ [49].

Initialization of subsequent landmarks

Suppose that the j-th landmark is detected for the first time at time-step k_o+k. This implies that the state vector already contains the first $(j-1)$ landmarks and thus $\mathbf{N}_R$ and $\mathbf{N}_{L_i}$ ($i = 1, \ldots, j-1$) have been computed. The nullspace matrix $\mathbf{N}$ now will have to be augmented by $\mathbf{N}_{L_j}$, corresponding to the new landmark, L_j. To determine $\mathbf{N}_{L_j}$, we first notice that, based on the structure of the measurement and state-propagation inferred Jacobians [see (4.41) and (4.38)], the corresponding block row of the observability matrix at this time step, denoted by $\mathbf{M}_{k_o+k}$, can be obtained as [see (4.55)]:

$$\begin{aligned} \mathbf{M}_{k_o+k} &\triangleq \breve{\mathbf{H}}_{k_o+k} \breve{\mathbf{\Phi}}_{k_o+k-1} \cdots \breve{\mathbf{\Phi}}_{k_o} = \\ &\begin{bmatrix} \breve{\mathbf{H}}_{R_{k_o+k}} \breve{\mathbf{\Phi}}_{R_{k_o+k-1}} \cdots \breve{\mathbf{\Phi}}_{R_{k_o}} & \mathbf{0} & \cdots & \mathbf{0} & \breve{\mathbf{H}}_{L_{j,k_o+k}} \end{bmatrix} \end{aligned} \tag{4.60}$$

Since this is the newest landmark, it is appended at the end of the state vector. Then, we compute $\mathbf{N}_{L_j}$ based on the requirement that each block row of the observability matrix $\mathbf{M}$ has the same nullspace, spanned by $\mathbf{N}$, i.e.,

$$\text{null}(\mathbf{M}_{k_o+k}) = \underset{\text{col.}}{\text{span}}(\mathbf{N}) \;\Rightarrow\; \mathbf{M}_{k_o+k}\mathbf{N} = \mathbf{0} \tag{4.61}$$

Substitution of (4.58) (using $M = j$) and (4.60) in (4.61) yields:

$$\breve{\mathbf{H}}_{R_{k_o+k}} \breve{\mathbf{\Phi}}_{R_{k_o+k-1}} \cdots \breve{\mathbf{\Phi}}_{R_{k_o}} \mathbf{N}_R + \breve{\mathbf{H}}_{L_{j,k_o+k}} \mathbf{N}_{L_j} = \mathbf{0} \Rightarrow$$
$$\mathbf{N}_{L_j} = -\breve{\mathbf{H}}^{-1}_{L_{j,k_o+k}} \breve{\mathbf{H}}_{R_{k_o+k}} \breve{\mathbf{\Phi}}_{R_{k_o+k-1}} \cdots \breve{\mathbf{\Phi}}_{R_{k_o}} \mathbf{N}_R \tag{4.62}$$

4.6.2 Computing the inferred measurement Jacobians

We know from Lemma 4.4.1 that we only need to determine the regression matrix $\mathbf{A}_1$, instead of the full regression matrix $\mathbf{A}$, in order to compute the inferred measurement Jacobians [see (4.40) and (4.41)]. Therefore, once the nullspace matrix $\mathbf{N}$ is available, at each update step, we formulate the following *constrained* linear-regression problem with respect to $\mathbf{A}_1$ and $\mathbf{b}$ [see (4.4)]:

$$\min_{\mathbf{A}_1, \mathbf{b}} \quad \sum_{i=0}^{10} w_i \left[\mathcal{Z}_i - (\mathbf{A}_1 \mathcal{X}_i + \mathbf{b})\right]^T \left[\mathcal{Z}_i - (\mathbf{A}_1 \mathcal{X}_i + \mathbf{b})\right] \tag{4.63}$$
$$\text{s.t.} \quad \mathbf{A}_1 \breve{\mathbf{\Phi}}'_{k_o+k-1} \cdots \breve{\mathbf{\Phi}}'_{k_o} \mathbf{N}_j = \mathbf{0} \tag{4.64}$$

where $\breve{\mathbf{\Phi}}'_{k_o+\ell} \triangleq \begin{bmatrix} \breve{\mathbf{\Phi}}_{R_{k_o+\ell}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix}$, $\ell = 0, \ldots, k-1$, denotes the reduced-size regression matrix obtained from propagation [see (4.32) and (4.38)], corresponding to the part of the state comprising only the robot pose and the observed j-th landmark at time-step $k_o + \ell$; and $\mathbf{N}_j \triangleq \begin{bmatrix} \mathbf{N}_R^T & \mathbf{N}_{L_j}^T \end{bmatrix}^T$ [see (4.58)] contains the corresponding block rows of $\mathbf{N}$. The sigma points used in the minimization problem (4.63) are computed by the procedure described in Section 4.4.2. The optimal solution of $\mathbf{A}_1$ is obtained in *closed form* using the following lemma:

Lemma 4.6.1. *The optimal solution to the constrained minimization problem* (4.63)-(4.64) *is given by:*

$$\begin{aligned} \mathbf{A}_1 &= \begin{bmatrix} \breve{\mathbf{H}}_{R_{k_o+k}} & \breve{\mathbf{H}}_{L_{j,k_o+k}} \end{bmatrix} \\ &= \bar{\mathbf{P}}_{\mathbf{z}\mathbf{x}_{k_o+k|k_o+k-1}} \mathbf{L}^T \left(\mathbf{L} \bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k_o+k|k_o+k-1}} \mathbf{L}^T \right)^{-1} \mathbf{L} \end{aligned} \tag{4.65}$$

with

$$\mathbf{L} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m \times (5-m)} \end{bmatrix} \left(\mathbf{I}_5 - \mathbf{U} (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \right) \tag{4.66}$$
$$\mathbf{U} \triangleq \breve{\mathbf{\Phi}}'_{k_o+k-1} \cdots \breve{\mathbf{\Phi}}'_{k_o} \mathbf{N}_j \tag{4.67}$$

where m is the dimension of the measurement vector.

Proof. See Appendix C.2. □

Lastly, once we construct the full inferred measurement Jacobian matrix $\breve{\mathbf{H}}_{k_o+k}$ in (4.41) from the regression matrix $\mathbf{A}_1$ in (4.65), we update the state estimate and covariance based on (4.21)-(4.24). In summary, the main steps of the OC-UKF SLAM are outlined in Algorithm 3.

We stress that if multiple landmarks are observed concurrently, the above process for determining the inferred measurement Jacobians is repeated sequentially for each of the landmarks. Note also that the maximum dimension of all the matrices involved in (4.65)-(4.66) is 5 [see (4.39)], and thus computing the regression matrix $\mathbf{A}_1$ incurs only a *constant* computational overhead, regardless of the number of landmarks in the state. As a result, the overall computational cost of the OC-UKF update step remains quadratic (as is the case for EKF-SLAM).

Algorithm 3 Observability-Constrained (OC)-UKF SLAM

Require: Initial state estimate and covariance

loop

`Propagation:` When an odometry measurement is received,
- determine sigma points by (4.25) with mean and covariance (4.30).
- produce regression points by passing the sigma points through (4.10).
- compute the state estimate (4.14).
- compute the regression matrix via (4.32).
- compute the propagated covariance via (4.34).

`Update:` When a robot-to-landmark measurement is received,
- determine sigma points by (4.25) with mean and covariance (4.39).
- produce regression points by passing the sigma points through the nonlinear function in (4.17).
- compute the regression matrix via (4.65) and (4.40).
- update the state and covariance via (4.21)-(4.24).

`Initialization:` When a new landmark is detected,
- determine sigma points by (4.25) with mean and covariance (4.46).
- produce regression points by passing the sigma points through (4.43).
- compute the inferred Jacobian matrices via (4.54).
- initialize the new landmark position (4.47) and update the state covariance (4.52).
- if this is the first observed landmark, compute the nullspace matrix $\mathbf{N}$ via (4.59), else augment the nullspace matrix $\mathbf{N}$ with $\mathbf{N}_{L_j}$, corresponding to the new landmark (4.62).

end loop

4.7 Simulation results

A series of Monte-Carlo comparison studies were conducted under various conditions, in order to verify the preceding consistency analysis and to compare the performance of the proposed OC-UKF to that of the standard UKF/EKF and the OC-EKF [72] as well as the iSAM algorithm [88], in terms of consistency and accuracy. The metrics used to evaluate estimation performance are RMSE and NEES [14]. The former provides a measure of accuracy, while the latter is a standard criterion for evaluating estimator consistency. Specifically, it is known that the NEES of an N-dimensional Gaussian random variable follows a χ^2 distribution with N d.o.f.. Therefore, if an estimator is consistent, we expect that the average NEES for the robot pose will be close to 3 for all time steps, and the average landmark NEES will be close to 2. The larger the deviations of the NEES from these values, the worse the inconsistency of the estimator. Note that when two estimators produce comparable RMSE, the one whose NEES value is closer to the expected is also the one whose estimated covariance is closer to the true one.[4] By studying both the RMSE and NEES of an estimator, we obtain a comprehensive picture of the estimator's performance.

4.7.1 SLAM with range-and-bearing measurements

In the simulation tests presented in this section, a robot with a simple differential-drive model drove on a planar surface, at a constant velocity of $v = 0.25$ m/sec. The two drive wheels were equipped with encoders, which measure their revolutions and provide measurements of velocity (i.e., right and left wheel velocities, v_r and v_l, respectively), with standard deviation equal to $\sigma = 2\%v$ for each wheel. These measurements were used to obtain the linear and rotational velocity measurements for the robot, which are given by:

$$v = \frac{v_r + v_l}{2}, \qquad \omega = \frac{v_r - v_l}{a}$$

where $a = 0.5$ m is the distance between the active wheels. The robot recorded distance and bearing measurements to landmarks lying within its sensing range of 5 m. The standard deviation of the distance-measurement noise was equal to 10% of the robot-to-landmark distance, while the standard deviation of the bearing-measurement noise was set to 1° deg. It should be

[4] It is important to stress that knowing the uncertainty of the computed estimates is often as important as the estimates themselves. An inconsistent estimator that reports covariance values smaller than the true ones can be unreliable for use in practice.

noted that the sensor-noise levels selected for the simulations are larger than what is typically encountered in practice. This was done on purpose in order to make the effects of inconsistency more apparent.

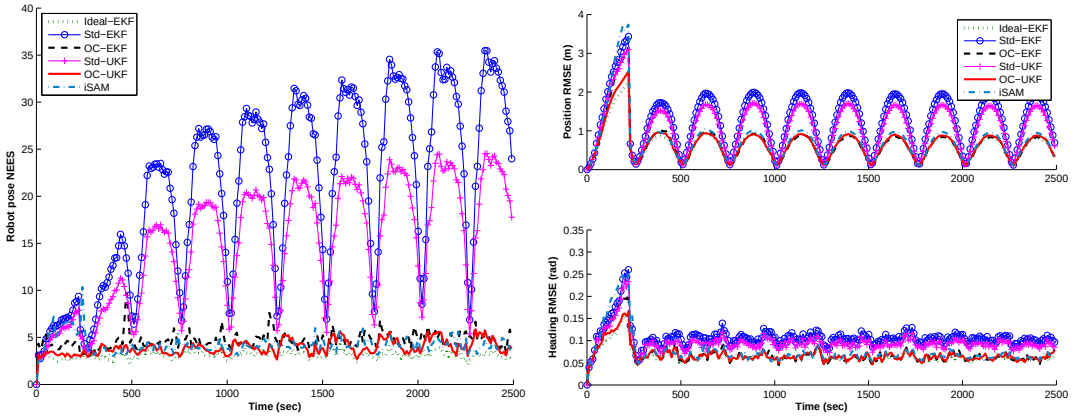
For the results shown here, a SLAM scenario with multiple loop closures was considered, where during each run, the robot executed 10 loops on a circular trajectory, and observed 20 landmarks in total. The reported results were averaged over 50 Monte-Carlo trials. During the test, six estimators processed the same data, to ensure a fair comparison.[5] The compared estimators were: (i) the ideal EKF, (ii) the standard EKF, (iii) the OC-EKF [72], (iv) the standard UKF, (v) the OC-UKF, and (vi) the iSAM algorithm [88]. Note that, as shown in [87], the performance of iSAM is very similar to (or even slightly better than) that of iSAM2 in landmark-based SLAM, which is the case considered in this work. Hence, in this test, we compared our algorithm to iSAM [88], using version 1.6 of its open-source implementation [89] with standard parameters, i.e., solving at every time step and reordering/relinearizing every 100 time steps. We also point out that, in order to ensure a fair comparison, we report the current-state estimates (instead of the final batch estimates) of the iSAM algorithm at each time step, which are computed by processing the measurements up to the current time step, without using any future measurements' information. Clearly, these incremental *causal* estimates are of more practical importance in any real-time robotic operation. Lastly, it is important to note that the ideal EKF is *not* realizable in practice since its Jacobians are evaluated at the (unknown) true values of the state. However, we included it as a benchmark in our simulations, since it has been shown to possess the *correct* observability properties and exhibit the best performance in terms of both consistency and accuracy [66, 70–72].

The comparative results for all the estimators are presented in Fig. 4.1 and Table 4.1. Specifically, Figs. 4.1(a) and 4.1(b) show the average NEES and RMSE, respectively, over all Monte-Carlo runs for each time step for the robot pose. On the other hand, Table 4.1 presents the average values of all relevant performance metrics for the landmarks and the robot. For the landmarks, we computed the average RMSE and NEES by averaging over all Monte-Carlo runs, all landmarks, and all time steps. For the robot position and orientation RMSEs and the robot pose NEES, we averaged the corresponding quantities over all Monte-Carlo runs and all

[5] In [72], the OC-EKF was shown to perform better, in terms of accuracy and consistency, than both the FEJ-EKF [70] and the robocentric mapping algorithm [27], which aims at improving the consistency of EKF-SLAM by expressing the landmarks in a robot-relative frame. Therefore, in this chapter we omitted the comparison between the proposed OC-UKF and the FEJ-EKF as well as the robocentric mapping filter.

Table 4.1: [Simulation Results. Range-and-bearing SLAM] Robot pose and landmark position estimation performance

Ideal-EKF	Std-EKF	OC-EKF	Std-UKF	OC-UKF	iSAM
Robot Position RMSE (m)					
0.6297	1.2664	0.6771	1.1002	0.6635	0.7587
Robot Heading RMSE (rad)					
0.0648	0.1070	0.0696	0.0954	0.0680	0.0760
Robot Pose NEES					
3.1284	20.6195	4.6896	14.8696	3.9305	4.2649
Landmark Position RMSE (m)					
0.6071	1.2552	0.6539	1.0890	0.6325	0.7732
Landmark Position NEES					
2.1569	19.5556	4.6150	13.7205	2.8303	10.1408



(a) (b)

Figure 4.1: [Simulation Results. Range-and-bearing SLAM] Monte-Carlo results for a SLAM scenario with multiple loop closures: (a) average NEES of the robot-pose errors, and (b) average RMSE for the robot pose (position and orientation). In these plots, the dotted lines correspond to the ideal EKF, the solid lines with circles to the standard EKF, the dashed lines to the OC-EKF, the solid lines with crosses to the standard UKF, the solid lines to the OC-UKF, and the dash-dotted lines to the iSAM algorithm. Note that the RMSE of the ideal EKF, the OC-EKF, the OC-UKF and the iSAM algorithm are very close, which makes the corresponding lines difficult to distinguish.

time steps.

Several interesting conclusions can be drawn from these results. Firstly, it becomes clear that the performance of the proposed OC-UKF is *very close* to that of the ideal EKF, and substantially better than both the standard EKF and the standard UKF, in terms of both RMSE (accuracy) and NEES (consistency). The observed performance gain indicates that the observability properties of the linear-regression-based system model employed in the UKF play a key role in determining the filter consistency: When these properties differ from those of the underlying nonlinear system, which is the case for the standard EKF and UKF, the filter's consistency is negatively impacted.

A second observation is that both the OC-UKF and the OC-EKF attain slightly better performance than the iSAM algorithm, in terms of consistency and accuracy (see Fig. 4.1 and Table 4.1). This can be justified by the fact that in order to reduce its processing requirements, the iSAM algorithm does *not* iteratively update the whole measurement Jacobian matrix (and thus the square-root information matrix) at every time step. Instead, it reuses partial results from

the previous time steps and only updates the Jacobian matrix incrementally by appending to it new rows corresponding to the most recent measurements. However, the previously-computed parts of the Jacobian matrix can be quite inaccurate (especially right before a loop closure event or in the presence of large measurement noise). Moreover, incremental updating process does not guarantee the appropriate observability properties. These factors can lead to significant estimation errors, which will propagate in time and degrade the iSAM algorithm's performance for all time steps except the ones where batch relinearization is applied. Clearly, this issue can be mitigated by performing periodic relinearization more frequently, which, however, will significantly increase the computational cost.

Lastly, the OC-UKF also outperforms the OC-EKF [72], by a smaller margin, in terms of both RMSE and NEES. It is interesting to note that the advantage of the OC-UKF over the OC-EKF is more pronounced in terms of NEES. This indicates that the OC-UKF provides a more accurate uncertainty measure (covariance) than the OC-EKF, and also implies that the filter's inconsistency primarily affects the covariance, rather than the state estimates. To further highlight this performance difference, in the next section, we also compare these algorithms in the case of bearing-only SLAM (BOSLAM), whose severe nonlinearities make the need for a better linearization scheme, such as the one offered by the OC-UKF, more evident.

4.7.2 SLAM with bearing-only measurements

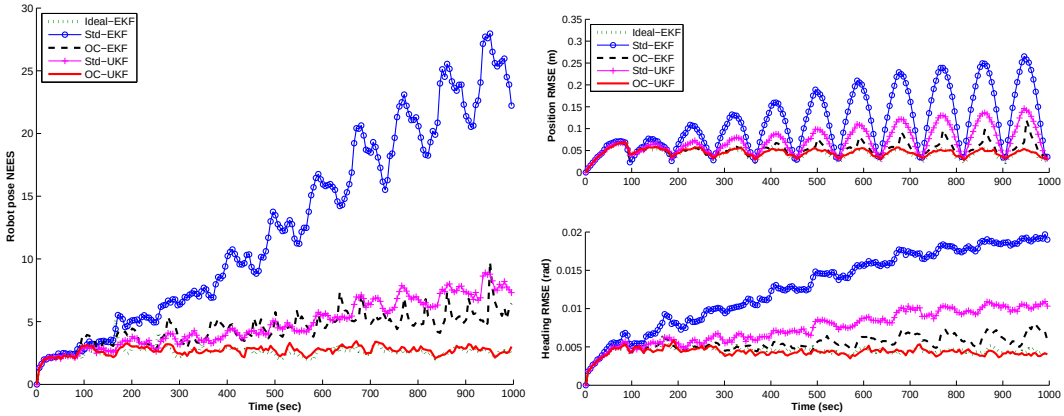
In this BOSLAM simulation test, we employed the same simulation setup as in the preceding case, with some changes in the parameters. Specifically, the robot moved on a circular trajectory at a constant velocity of $v = 0.5$ m/sec, with wheel-velocity measurement noise standard deviation equal to $\sigma = 1\%v$, while the standard deviation of the bearing-measurement noise was set to 2 deg. Note that we doubled the robot velocity in this simulation, because a larger linear velocity increases the baseline between two consecutive time steps, leading to a more reliable triangulation-based landmark initialization [10].

The comparative results[6] of the robot pose and landmark position estimation are shown in Fig. 4.2 and Table 4.2. As evident, in the case of BOSLAM where the measurement nonlinearity is more significant than that of the range-and-bearing SLAM considered earlier, the standard UKF performs substantially better than the standard EKF, in terms of both consistency

[6] Since the current implementation of iSAM [89] does not include the bearing-only case, in this test we omit the comparison of the OC-UKF to the iSAM algorithm while focusing on that to the OC-EKF.

Table 4.2: [Simulation Results. Bearing-only SLAM] Robot pose and landmark position estimation performance

Ideal-EKF	Std-EKF	OC-EKF	Std-UKF	OC-UKF
Robot Position RMSE (m)				
0.0427	0.1132	0.0529	0.0707	0.0455
Robot Heading RMSE (rad)				
0.0045	0.0130	0.0055	0.0075	0.0043
Robot Pose NEES				
2.6054	12.6715	4.4730	4.8453	2.6917
Landmark Position RMSE (m)				
0.1066	0.1770	0.1305	0.1630	0.1471
Landmark Position NEES				
1.8964	12.7627	12.6085	6.1927	4.3216



(a) (b)

Figure 4.2: [Simulation Results. Bearing-only SLAM] Monte-Carlo results for a SLAM scenario with multiple loop closures: (a) average NEES of the robot-pose errors, and (b) average RMSE for the robot pose (position and orientation). In these plots, the dotted lines correspond to the ideal EKF, the solid lines with circles to the standard EKF, the dashed lines to the OC-EKF, the solid lines with crosses to the standard UKF, and the solid lines to the OC-UKF. Note that the RMSE of the ideal EKF and the OC-UKF are almost identical, which makes the corresponding lines difficult to distinguish.

(NEES) and accuracy (RMSE). This performance gain is also shared by the OC-UKF over the OC-EKF. We thus see that the OC-UKF combines the benefits of the OC-EKF (i.e., correct observability properties) with those of the UKF (i.e., better linearization), to form an estimator whose performance is comparable to that of the ideal EKF.

4.8 Experimental results

To further test the proposed OC-UKF SLAM algorithm, we also conducted real-world experiments in both indoor and outdoor environments. These tests also allow us to examine the algorithm's runtime, as compared to the OC-EKF and the state-of-the-art iSAM algorithm. All the timing results presented in this section were obtained on a Mac laptop with an Intel i5 processor at 2.53 GHz, and 4GB of RAM.

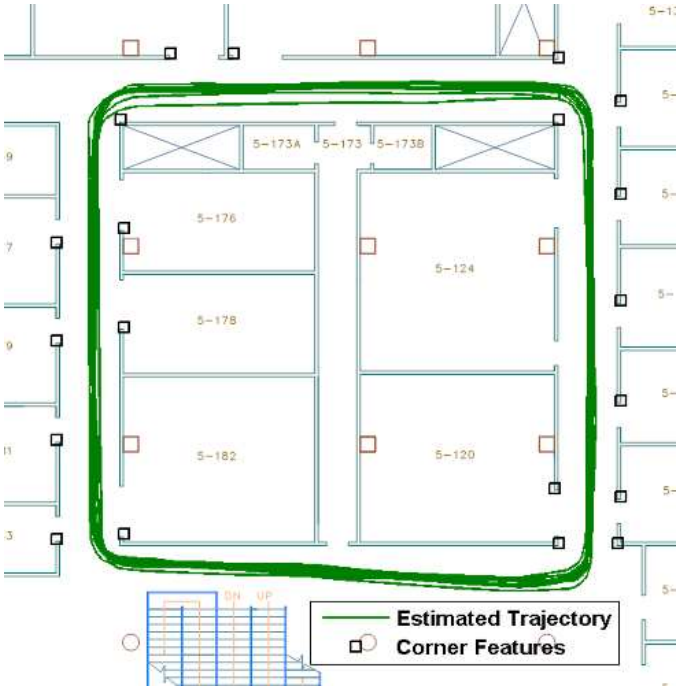


Figure 4.3: [Indoor Experiment] The batch maximum-a-posteriori (MAP) estimate of the robot trajectory during the indoor experiment (solid line), overlaid on the blueprint of the building. The boxes (□) denote the corners whose exact locations were known from the building's blueprints. The batch MAP estimates of the robot poses and the known corners were used as ground truth for computing the NEES and RMSE values shown in Table 4.3 and Fig. 4.4.

4.8.1 Indoor environment

We first present the results of the indoor experiment conducted in an office building. The robot was commanded to perform 11 loops around a square with sides approximately equal to 20 m (see Fig. 4.3). This trajectory was selected since repeated re-observation of the same landmarks tends to make the effects of inconsistency more apparent, and facilitates discerning the performance of the various estimators. A Pioneer robot equipped with a SICK LMS200 laser range-finder and wheel encoders was used in this experiment. From the laser-range data, corner features were extracted and used as landmarks, while the wheel encoders provided the linear and rotational velocity measurements. In particular, this dataset was recorded over about 40 minutes, and contains 23425 robot poses and 63 landmarks with 11392 measurements to them.

Since no ground truth for the robot pose could be obtained using external sensors (e.g.,

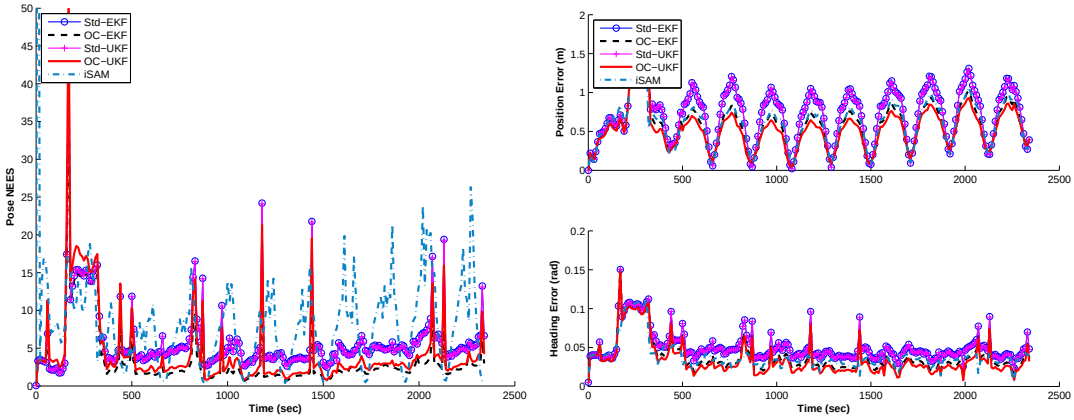
overhead cameras) in this experiment, we obtained a reference trajectory, treated as ground truth, by utilizing the known map of the area where the experiment took place. Specifically, the exact locations of 20 corners were known from the blueprints of the building. Measurements to these corners, as well as all other measurements obtained by the robot (including those to corners whose locations were not known *a priori*), were processed offline using a batch MAP estimator [91] to obtain an accurate estimate of the entire trajectory. This estimate, as well as the locations of the known corners, are shown in Fig. 4.3. This constitutes the ground truth against which the performance of the following five estimators was compared: (i) the standard EKF, (ii) the OC-EKF, (iii) the standard UKF, (iv) the OC-UKF, and (v) the iSAM. Clearly, due to the way the ground truth is computed, the estimation errors are expected to have some correlation to the errors in the ground truth. However, since these correlations are the same for all estimators, we can still have a fair comparison of their relative performance.

The comparative results for all estimators are presented in Figs. 4.4(a) and 4.4(b), while Table 4.3 shows the averaged NEES and RMSE of the robot pose and landmark position, respectively. We point out that during the experiment the robot detected a number of landmarks that were *not* included in the set of 20 known corners (e.g., movable objects such as furniture). Since no ground truth was available for these objects, we only used the 20 known corners for computing the landmarks' error statistics. From the experimental results, it becomes evident that the OC-UKF outperforms both the standard EKF and UKF, and also achieves better accuracy than the OC-EKF. This agrees with the simulation results presented in the preceding section. It should be noted that the reported NEES in Fig. 4.4(a) was computed only from a single run (i.e., this is not an average over many Monte-Carlo runs as in the simulations). To evaluate an estimator's consistency, the average NEES over many Monte-Carlo runs is a suitable metric, while the NEES values in a single experiment do not dictate which estimator is consistent or not. Regardless, we show these results mainly to demonstrate the large difference in performance between the OC-EKF/UKF and the standard EKF/UKF. These experimental results, along with those from the simulations, further support our conjecture that the mismatch in the dimension of the unobservable subspace between the statistically-linearized SLAM system and the underlying nonlinear system is a fundamental cause of filter inconsistency.

As evident from Fig. 4.4(b) and Table 4.3, the OC-EKF/UKF achieve similar accuracy to, and better consistency than, the iSAM algorithm. As mentioned in the previous section, one possible explanation for this is that the iSAM algorithm does not iteratively update the whole

Table 4.3: [Indoor Experiment] Robot pose and landmark position estimation performance and runtime

Std-EKF	OC-EKF	Std-UKF	OC-UKF	iSAM
Robot Position RMSE (m)				
0.7323	0.5896	0.7268	0.5384	0.6108
Robot Heading RMSE (rad)				
0.0512	0.0392	0.0508	0.0349	0.0388
Robot Pose NEES				
6.0939	3.4575	6.0307	4.5442	9.1270
Landmark Position RMSE (m)				
0.9929	0.8438	0.9894	0.8183	0.6528
Landmark Position NEES				
7.3180	6.0354	7.2928	7.0123	9.6627
Total CPU Execution Time (sec)				
304.761	304.251	306.689	307.930	350.379



(a) (b)

Figure 4.4: [Indoor Experiment] Experimental results: (a) NEES of the robot-pose errors, and (b) estimation errors of the robot pose (position and orientation). In these plots, the solid lines with circles correspond to the standard EKF, the dashed lines to the OC-EKF, the solid lines with crosses to the standard UKF, the solid lines to the OC-UKF, and the dash-dotted lines to the iSAM algorithm. Note that the NEES and estimation error values of the standard EKF and the standard UKF are almost identical, and the estimation errors of the OC-UKF, the OC-EKF and the iSAM algorithm are also very close to each other, which makes the corresponding lines difficult to distinguish.

measurement Jacobian at each time step, which may incur large linearization and thus estimation errors. Inaccuracies in the measurement Jacobian propagate into the covariance estimated by the iSAM algorithm, which results in significantly higher NEES values as compared to the OC-EKF/UKF. Interestingly, as seen from Table 4.3, the OC-UKF has a lower computational cost than the iSAM in this experiment, although all the algorithms attain faster-than-real-time performance. This can be justified by the fact that the computational cost of the iSAM algorithm increases as the robot trajectory grows. Moreover, the 11 loop-closing events occurring along the robot trajectory in this experiment significantly increase fill-in in the square-root information matrix and thus the computational complexity for solving the system.

At this point we should note that in this indoor experiment (as well as the outdoor experiment presented in the next section) the measurement correspondences were known. If not, then to solve the data association problem, the iSAM algorithm would need to recover marginal covariances, which will significantly increase its processing requirements [88]. By contrast, since

Table 4.4: [Outdoor Experiment] Estimation accuracy and runtime

iSAM	OC-EKF	OC-UKF
Robot Position RMSE (m)		
4.2111	5.9069	3.8084
Total CPU Execution Time (sec)		
31.5482	35.6811	34.6138

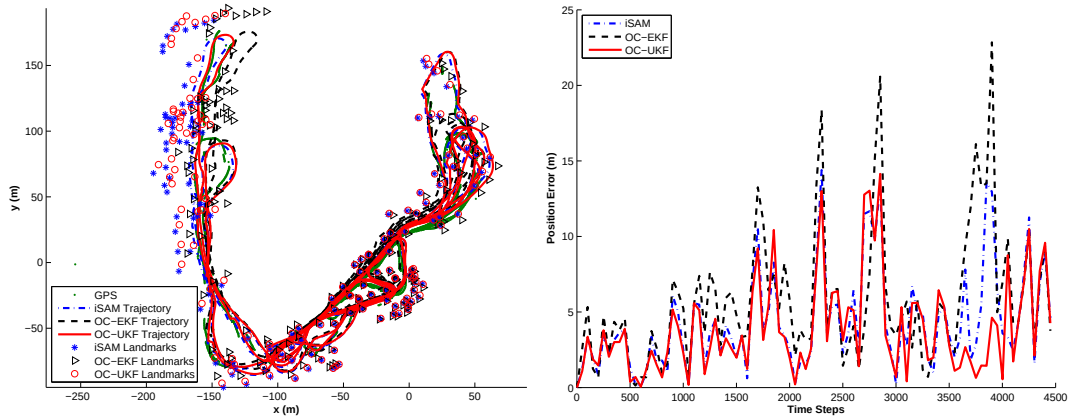
the covariance matrix is maintained in the OC-EKF/UKF, the marginal covariances are immediately available and hence the maximum-likelihood data association incurs minimal overhead.

Lastly, it is very important to observe from Table 4.3 that the two UKFs (i.e., the standard UKF and the proposed OC-UKF) have similar timing performance as the two EKFs (i.e, the standard EKF and the OC-EKF). This is attributed to the proposed sampling strategy (see Lemma 4.4.1), which results in the UKF having computational complexity of the same order as that of the EKF.

4.8.2 Outdoor environment

To further examine the performance of the proposed OC-UKF, we tested our algorithm on a publicly available SLAM dataset, the Sydney Victoria Park dataset. The experimental platform was a 4-wheeled vehicle equipped with a kinematic GPS, a laser sensor, and wheel encoders. The GPS system was used to provide ground truth for the robot position. Wheel encoders were used to provide odometry measurements, and propagation was carried out using the Ackerman model. In this particular application, since the most common features in the environment were trees, the profiles of trees were extracted from the laser data, and the centers of the trunks were then used as the point landmarks. It should be pointed out that in this test, to ensure a fair comparison with the iSAM algorithm, we employed the preprocessed dataset which is also available in the iSAM package [89]. This preprocessed dataset contains 6969 robot poses and 151 landmarks with 3640 measurements, recorded over 26 minutes.

Since the OC-EKF and the OC-UKF were already shown in the preceding simulations and experiment to perform significantly better, in terms of accuracy and consistency, than the standard EKF and UKF, in this test, we omitted the comparison to the two latter filters for clarity of



(a) (b)

Figure 4.5: [Outdoor Experiment] Experimental results: (a) Robot trajectory and landmark estimates as compared to the GPS data, and (b) robot position estimation errors. Note that, since in this test the GPS satellite signals were not always available, we computed the estimation errors only when GPS was available. In these plots, the dash-dotted lines and stars correspond to the iSAM estimates of the trajectory and the landmarks, respectively, the dashed lines and triangles to the OC-EKF, and the solid lines and circles to the OC-UKF, while the dots denote the sparse GPS data points.

presentation. Instead, we focus on the accuracy comparison of the OC-UKF with the OC-EKF and the iSAM algorithm. In this experiment, true landmark positions and true robot orientations were not available. We hence only compared the position-estimation performance of the three approaches (i.e., the OC-EKF, the OC-UKF, and the iSAM algorithm). Note also that, as mentioned in Section 4.8.1, the NEES computed from a single experimental run is not well-suited for analyzing the consistency of the estimators, and thus we hereafter focus on the comparison of accuracy and processing requirements. Specifically, Fig. 4.5(a) depicts the trajectory and landmark estimates produced by the three estimators as compared to the GPS ground truth, while Fig. 4.5(b) shows the corresponding estimation errors of the robot position. Table 4.4 shows the average estimation errors (i.e., RMSE) of robot position as well as the total CPU runtime for the three estimators compared. Clearly, the OC-UKF achieves better accuracy than both the OC-EKF and iSAM, while incurring comparable computational cost. In particular, the OC-UKF attains 36% and 10% reduction in robot position estimation errors as compared to the

OC-EKF and iSAM, respectively, while at 3% lower [7] and only 10% higher computational cost. We repeat that the timing result for iSAM does not include the runtime of computing marginal covariances for data association. These results agree with what we have observed in the indoor experiment presented in Section 4.8.1.

4.9 Summary

This work focuses on UKF-based SLAM, and in particular on the issues of computational complexity and filter inconsistency. The first contribution of this work is the formulation of a novel UKF-based SLAM algorithm that has computational complexity of the same order as that of EKF-based SLAM. In particular, we have proposed a new sampling scheme in which the unscented transformation employed by the UKF is only applied to the subset of states that appear in the nonlinear process and measurement models, instead of the entire state. Thus, by adopting this new sampling scheme, the UKF-based SLAM requires computing the square root of small, constant-size matrices, which leads to computational complexity *linear* during propagation, and *quadratic* during update. Furthermore, we have shown that a mismatch between the observability properties of the linear-regression-based system model employed by the UKF, and those of the underlying nonlinear SLAM system, causes inconsistency. To address this issue, we have introduced a novel Observability-Constrained (OC)-UKF, which ensures that the UKF system model has an unobservable subspace of appropriate dimensions, by enforcing observability constraints on the filter's *inferred* Jacobians. Through extensive Monte-Carlo simulations and real-world experiments, the OC-UKF is shown to achieve comparable or better performance, in terms of consistency, accuracy and computational complexity, as compared to other state-of-the-art SLAM algorithms such as the OC-EKF and iSAM.

[7] Note that in this experiment, the OC-UKF has lower cost than the OC-EKF, primarily because the Mahalanobis-distance test [14] in the OC-UKF rejects more outlier measurements than that in the OC-EKF.

Chapter 5

Observability-Constrained SWF for SLAM

In this chapter, we study the consistency issue of smoothing approaches, in particular, the sliding-window filter (SWF) when applied to the SLAM problem. By adapting our observability-based methodology that was presented within the filtering framework, we develop a novel Observability-Constrained (OC)-SWF. Part of this chapter has been published in [67].

5.1 Introduction

As discussed before, among the existing approaches for robot localization, the EKF is one of the most popular methods, which is primarily due to its ease of implementation and relatively low processing requirements. However, the EKF, as well as any linearization-based filtering approach, may suffer from the accumulation of linearization errors. This is because once linearization points are selected at a given time step for computing the filter Jacobians, they cannot be updated at later times, when more measurements become available for improving them. In contrast, a batch-MAP estimator [91] can improve the estimation accuracy by computing consistent state estimates for all time steps based on all available measurements. Under a Gaussian prior and measurement noise assumption (which is common in practice), finding the MAP estimates requires solving a nonlinear least-squares problem (see Section 5.2), whose counterpart in computer vision is known as bundle adjustment [163]. A variety of iterative algorithms have been employed for solving this problem. For example, the square-root SAM method [36] solves

the SLAM problem efficiently by using variable reordering, a well-known technique for sparse linear systems. However, since the size of the state vector in the batch-MAP estimator increases continuously over time, the processing and memory requirements become too high for real-time operation in large-scale problems (e.g., a robot exploring a large environment with millions of landmarks).

To overcome this limitation, a sliding-window filter (SWF) [142] (also called a fixed-lag smoother (FLS) [38, 111, 132]) can be used to estimate the states over a sliding time window at a fixed computational cost. The SWF concurrently processes all the measurement constraints between states in the window, and better addresses the nonlinearity of the problem by iteratively relinearizing the process and measurement equations. This approach is resource-adaptive: depending on the available computational resources, it can scale from the iterated EKF solution if only a single time step is maintained, to the optimal batch-MAP solution if the sliding window spans the entire time horizon.

The key characteristic of the SWF is the marginalization of old states from the sliding window, a process that appropriately models the uncertainty of these states [132, 142, 163]. However, due to marginalization, *different estimates* of the *same states* are used as *linearization points* in computing the Hessian matrix during estimation (see Section 5.3 and [38]). This results in different parameter observability properties [14] as compared to the batch-MAP estimator. Specifically, the Hessian (Fisher information matrix) of the standard SWF has a nullspace of lower dimension than that of the batch-MAP estimator. This implies that the estimator erroneously *believes* it has information along more directions of the state space than those contained in the measurements. This leads to inconsistent estimates, i.e., estimates whose accuracy and uncertainty measure are worse than the actual ones. This estimation inconsistency is a serious problem, since when an estimator is inconsistent, the accuracy of the produced estimates is unknown, which in turn makes the estimator unreliable [14].

In order to improve the consistency and accuracy of the SWF, in this chapter we propose an Observability-Constrained (OC)-SWF as a general smoothing framework. In particular, we postulate that by ensuring the Hessian matrix has a nullspace of appropriate dimension, we can avoid the influx of spurious information in the unobservable directions of the parameter (state) space, thus improving the consistency of the estimates. Based on this insight, the OC-SWF extends the observability-based methodology for designing consistent EKFs (see Chapter 2

and [72]). The key idea behind our approach is to select the linearization points for computing the Jacobians, and thus the Hessian, so as to ensure that its nullspace dimension does not arbitrarily decrease.

It should be pointed out that a prior-linearization (PL)-SWF for motion estimation was proposed in [38]. In particular, the PL-SWF computes the Hessian using the prior, instead of the current, estimates, for the states connected via measurements to marginalized states. This ensures the same estimates for the same states are used, and the appropriate dimension of the Hessian's nullspace is preserved. However, if the prior estimates are inaccurate, the linearization errors will be large and may degrade the estimator's performance. In contrast, the proposed OC-SWF selects *optimal* linearization points for computing the Hessian, in the sense that they not only ensure the correct dimension for the nullspace of the Hessian, but also minimize the linearization errors. We stress that apart from the SLAM problem treated in this work, the proposed OC-SWF is applicable to a large class of nonlinear estimation problems in robotics and computer vision, such as visual odometry [126] and vision-aided inertial navigation [120].

5.2 SLAM batch-MAP formulation

In this section, we describe the batch-MAP formulation of the SLAM problem, which forms the basis for the ensuing derivations of the SWF. In particular, we aim at estimating the entire robot trajectory up to the current time-step k, as compared to the only current robot pose in the filtering approaches, as well as the positions of all observed landmarks [see (2.1)]:

$$\mathbf{x}_{0:k} = \begin{bmatrix} \mathbf{x}_{R_0}^T & \mathbf{x}_{R_1}^T & \cdots & \mathbf{x}_{R_k}^T & \mathbf{p}_{L_1}^T & \ldots & \mathbf{p}_{L_M}^T \end{bmatrix}^T \tag{5.1}$$

In what follows, we start by presenting the general motion and measurement models that are similar to the ones used in the EKF-SLAM (see Section 2.3). Subsequently, we describe the batch-MAP estimator.

5.2.1 Motion model

Consider a robot equipped with an odometry sensor moving on a plane. The odometry serves as the control input to propagate the robot pose, according to the following motion model [also

see (2.2) and (2.3)]:

$$\mathbf{p}_{R_k} = \mathbf{p}_{R_{k-1}} + \mathbf{C}(\phi_{R_{k-1}})\, {}^{R_{k-1}}\mathbf{p}_{R_k} \tag{5.2}$$

$$\phi_{R_k} = \phi_{R_{k-1}} + {}^{R_{k-1}}\phi_{R_k} \tag{5.3}$$

where $\mathbf{C}(\cdot)$ denotes the 2×2 rotation matrix, and $\mathbf{u}_{k-1} = {}^{R_{k-1}}\mathbf{x}_{R_k} = [{}^{R_{k-1}}\mathbf{p}_{R_k}^T \quad {}^{R_{k-1}}\phi_{R_k}]^T$ is the true odometry (control input), i.e., the robot's motion between time-steps $k-1$ and k, expressed with respect to the robot's frame at time-step $k-1$, $\{R_{k-1}\}$. The corresponding odometry measurement, $\mathbf{u}_{m_{k-1}}$, is assumed to be corrupted by zero-mean white Gaussian noise, $\mathbf{w}_{k-1} = \mathbf{u}_{k-1} - \mathbf{u}_{m_{k-1}}$, with covariance $\mathbf{Q}_{k-1}$. This motion model is described by the following generic nonlinear function:

$$\mathbf{g}(\mathbf{x}_{0:k}, \mathbf{u}_{k-1}) = \mathbf{x}_{R_k} - \mathbf{f}(\mathbf{x}_{R_{k-1}}, \mathbf{u}_{m_{k-1}} + \mathbf{w}_{k-1}) = \mathbf{0} \tag{5.4}$$

To employ the batch-MAP estimator, it is necessary to linearize (5.4) and compute the Jacobians with respect to the state vector (5.1) and the noise, respectively, i.e.,

$$\mathbf{\Phi}_{k-1} \triangleq \frac{\partial \mathbf{g}}{\partial \mathbf{x}_{0:k}}\bigg|_{\{\mathbf{x}^\star_{0:k}, \mathbf{0}\}} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \cdots & \mathbf{\Phi}_{R_{k-1}} & \mathbf{I}_3 & \mathbf{0}_{3\times 2} & \cdots & \mathbf{0}_{3\times 2} \end{bmatrix} \tag{5.5}$$

$$\mathbf{G}_{k-1} \triangleq \frac{\partial \mathbf{g}}{\partial \mathbf{w}_{k-1}}\bigg|_{\{\mathbf{x}^\star_{0:k}, \mathbf{0}\}} = \begin{bmatrix} \mathbf{C}(\phi^\star_{R_{k-1}}) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{5.6}$$

with

$$\mathbf{\Phi}_{R_{k-1}} = -\begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\mathbf{p}^\star_{R_k} - \mathbf{p}^\star_{R_{k-1}}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{5.7}$$

where $\mathbf{x}^\star_{0:k}$ denotes the linearization point for the state (5.1), while a zero vector is used as the linearization point for the noise. Clearly, the values of the Jacobian matrices depend on the choice of linearization points, which is the key fact our approach relies on.

5.2.2 Measurement model

As discussed before, the robot-to-landmark measurements in SLAM are a function of the relative position of the observed landmark with respect to the robot [also see (2.10)]:

$$\mathbf{z}_{ij} = \mathbf{h}_{ij}(\mathbf{x}_{0:k}) + \mathbf{v}_{ij} = \mathbf{h}\left({}^{R_j}\mathbf{p}_{L_i}\right) + \mathbf{v}_{ij} \tag{5.8}$$

where ${}^{R_j}\mathbf{p}_{L_i} = \mathbf{C}^T(\phi_{R_j})(\mathbf{p}_{L_i} - \mathbf{p}_{R_j})$ is the position of the i-th landmark with respect to the robot at time-step j, and $\mathbf{v}_{ij}$ is zero-mean Gaussian measurement noise with covariance $\mathbf{R}_{ij}$. In this work, we allow $\mathbf{h}(\cdot)$ to be *any* measurement function (e.g., a direct measurement of relative position, a pair of range and bearing measurements, bearing-only measurements, etc.). In general, the measurement function is nonlinear, and its Jacobian matrix is computed as:

$$\mathbf{H}_{ij} \triangleq \frac{\partial \mathbf{h}_{ij}}{\partial \mathbf{x}_{0:k}}\bigg|_{\{\mathbf{x}^\star_{0:k},\mathbf{0}\}} = \begin{bmatrix}\mathbf{0} & \cdots & \mathbf{H}_{R_{ij}} & \mathbf{0} & \cdots & \mathbf{H}_{L_{ij}} & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix} \tag{5.9}$$

with

$$\mathbf{H}_{R_{ij}} = (\nabla \mathbf{h}_{ij})\mathbf{C}^T(\phi^\star_{R_j}) \begin{bmatrix}-\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}^\star_{L_i} - \mathbf{p}^\star_{R_j})\end{bmatrix} \tag{5.10}$$

$$\mathbf{H}_{L_{ij}} = (\nabla \mathbf{h}_{ij})\mathbf{C}^T(\phi^\star_{R_j}) \tag{5.11}$$

where $\mathbf{H}_{R_{ij}}$ and $\mathbf{H}_{L_{ij}}$ are the Jacobians with respect to the robot pose at time-step j and the i-th landmark position, respectively, and $\nabla \mathbf{h}_{ij}$ denotes the Jacobian of $\mathbf{h}_{ij}$ with respect to the robot-relative landmark position, ${}^{R_j}\mathbf{p}_{L_i}$, evaluated at the linearization point, $\mathbf{x}^\star_{0:k}$.

5.2.3 Batch-MAP estimator

The batch-MAP estimator utilizes all the available information to estimate the state vector (5.1). The information used includes: (i) the prior information about the initial state, described by a Gaussian pdf with mean $\hat{\mathbf{x}}_{0|0}$ and covariance $\mathbf{P}_{0|0}$, (ii) the motion information (5.4), and (iii) the sensor measurements (5.8). In particular, the batch-MAP estimator seeks to determine the estimate $\hat{\mathbf{x}}_{0:k|k}$ that maximizes the posterior pdf:

$$p(\mathbf{x}_{0:k}|\mathcal{Z}_{0:k}) \propto p(\mathbf{x}_{R_0}) \prod_{\kappa=1}^{k} p(\mathbf{x}_{R_\kappa}|\mathbf{x}_{R_{\kappa-1}}) \prod_{\mathbf{z}_{ij}\in\mathcal{Z}_{0:k}} p(\mathbf{z}_{ij}|\mathbf{x}_{R_j}, \mathbf{p}_{L_i}) \tag{5.12}$$

where $\mathcal{Z}_{0:k}$ denotes all the available measurements in the time interval $[0, k]$. For Gaussian state and measurement noise [see (5.4), and (5.8), respectively], this pdf (5.12) can be written as:

$$
\begin{aligned}
& p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k}) \propto \\
& \frac{1}{\sqrt{(2\pi)^n|\mathbf{P}_{0|0}|}} \exp\left(-\frac{1}{2}||\mathbf{x}_{R_0} - \hat{\mathbf{x}}_{0|0}||^2_{\mathbf{P}_{0|0}}\right) \times \\
& \prod_{\kappa=1}^{k} \frac{1}{\sqrt{(2\pi)^3|\mathbf{Q}'_{\kappa-1}|}} \exp\left(-\frac{1}{2}||\mathbf{x}_{R_\kappa} - \mathbf{f}(\mathbf{x}_{R_{\kappa-1}}, \mathbf{u}_{m_{\kappa-1}})||^2_{\mathbf{Q}'_{\kappa-1}}\right) \times \\
& \prod_{\mathbf{z}_{ij}\in\mathcal{Z}_{0:k}} \frac{1}{\sqrt{(2\pi)^m|\mathbf{R}_{ij}|}} \exp\left(-\frac{1}{2}||\mathbf{z}_{ij} - \mathbf{h}_{ij}(\mathbf{x}_{0:k})||^2_{\mathbf{R}_{ij}}\right)
\end{aligned}
\tag{5.13}
$$

where $n = \dim(\mathbf{x}_{R_0})$ is the dimension of the prior state $\mathbf{x}_{R_0}$, and $m = \dim(\mathbf{z}_k)$ is the dimension of the measurement $\mathbf{z}_k$. In the above expression, we have also employed the notations, $||\mathbf{a}||^2_{\mathbf{M}} \triangleq \mathbf{a}^T\mathbf{M}^{-1}\mathbf{a}$ and $\mathbf{Q}'_k \triangleq \mathbf{G}_{R_k}\mathbf{Q}_k\mathbf{G}^T_{R_k}$ [see (5.4)]. Hence, maximizing (5.13) is equivalent to minimizing the following cost function:

$$
\begin{aligned}
c(\mathbf{x}_{0:k}) =& \frac{1}{2}||\mathbf{x}_{R_0} - \hat{\mathbf{x}}_{0|0}||^2_{\mathbf{P}_{0|0}} + \\
& \sum_{\kappa=1}^{k} \frac{1}{2}||\mathbf{x}_{R_\kappa} - \mathbf{f}(\mathbf{x}_{R_{\kappa-1}}, \mathbf{u}_{m_{\kappa-1}})||^2_{\mathbf{Q}'_{\kappa-1}} + \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_{0:k}} \frac{1}{2}||\mathbf{z}_{ij} - \mathbf{h}_{ij}(\mathbf{x}_{0:k})||^2_{\mathbf{R}_{ij}}
\end{aligned}
\tag{5.14}
$$

The cost function $c(\mathbf{x}_{0:k})$ is nonlinear, and a standard approach to determine its minimum is to employ Guass-Newton iterative minimization [163]. Specifically, at the ℓ-th iteration of this method, a correction, $\delta\mathbf{x}^{(\ell)}_{0:k}$, to the current estimate, $\hat{\mathbf{x}}^{(\ell)}_{0:k|k}$, is computed by minimizing the second-order Taylor-series approximation of the cost function which is given by:

$$
c(\hat{\mathbf{x}}^{(\ell)}_{0:k|k} + \delta\mathbf{x}^{(\ell)}_{0:k}) \simeq c(\hat{\mathbf{x}}^{(\ell)}_{0:k|k}) + \mathbf{b}^{(\ell)T}_b \delta\mathbf{x}^{(\ell)}_{0:k} + \frac{1}{2}\delta\mathbf{x}^{(\ell)T}_{0:k}\mathbf{A}^{(\ell)}_b\delta\mathbf{x}^{(\ell)}_{0:k} \tag{5.15}
$$

where

$$
\mathbf{b}^{(\ell)}_b \triangleq \nabla_{\mathbf{x}_{0:k}} c(\cdot)\Big|_{\{\mathbf{x}^\star_{0:k}=\hat{\mathbf{x}}^{(\ell)}_{0:k|k}\}} \tag{5.16}
$$

$$
\mathbf{A}^{(\ell)}_b \triangleq \nabla^2_{\mathbf{x}_{0:k}} c(\cdot)\Big|_{\{\mathbf{x}^\star_{0:k}=\hat{\mathbf{x}}^{(\ell)}_{0:k|k}\}} \tag{5.17}
$$

are the gradient and Hessian of $c(\cdot)$ with respect to $\mathbf{x}_{0:k}$, evaluated at the current state estimate $\hat{\mathbf{x}}^{(\ell)}_{0:k|k}$, i.e., $\mathbf{x}^{\star}_{0:k} = \hat{\mathbf{x}}^{(\ell)}_{0:k|k}$. Specifically, at the ℓ-th iteration, $\mathbf{b}^{(\ell)}_b$ is [see (5.5) and (5.9)]:

$$\mathbf{b}^{(\ell)}_b = \mathbf{\Pi}^T \mathbf{P}^{-1}_{0|0} \left(\hat{\mathbf{x}}^{(\ell)}_{R_{0|k}} - \hat{\mathbf{x}}_{0|0} \right) - \sum_{\mathbf{z}_{ij} \in \mathcal{Z}_{0:k}} \mathbf{H}^{(\ell)^T}_{ij} \mathbf{R}^{-1}_{ij} \left(\mathbf{z}_{ij} - \mathbf{h}_{ij}(\hat{\mathbf{x}}^{(\ell)}_{0:k|k}) \right) + \sum_{\kappa=1}^{k} \mathbf{\Phi}^{(\ell)^T}_{\kappa-1} \mathbf{Q}'^{-1}_{\kappa-1} \left(\hat{\mathbf{x}}^{(\ell)}_{R_{\kappa|k}} - \mathbf{f}(\hat{\mathbf{x}}^{(\ell)}_{R_{\kappa-1|k}}, \mathbf{u}_{m_{\kappa-1}}) \right) \tag{5.18}$$

where $\mathbf{\Pi} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$. On the other hand, the Hessian matrix, $\mathbf{A}^{(\ell)}_b$, is approximated in the Gauss-Newton method by [see (5.5) and (5.9)]:

$$\mathbf{A}^{(\ell)}_b \simeq \mathbf{\Pi}^T \mathbf{P}^{-1}_{0|0} \mathbf{\Pi} + \sum_{\mathbf{z}_{ij} \in \mathcal{Z}_{0:k}} \mathbf{H}^{(\ell)^T}_{ij} \mathbf{R}^{-1}_{ij} \mathbf{H}^{(\ell)}_{ij} + \sum_{\kappa=1}^{k} \mathbf{\Phi}^{(\ell)^T}_{\kappa-1} \mathbf{Q}'^{-1}_{\kappa-1} \mathbf{\Phi}^{(\ell)}_{\kappa-1} \tag{5.19}$$

which is a good approximation for small-residual problems [163]. Due to the sparse structure of the matrices $\mathbf{H}^{(\ell)}_{ij}$ and $\mathbf{\Phi}^{(\ell)}_{\kappa}$ [see (5.5) and (5.9)], the matrix $\mathbf{A}^{(\ell)}_b$ is also sparse, which can be exploited to speed-up the solution of the linear system in (5.20) [163]. The value $\delta\mathbf{x}^{(\ell)}_{0:k}$ that minimizes (5.15) is found by solving the following linear system:

$$\mathbf{A}^{(\ell)}_b \delta\mathbf{x}^{(\ell)}_{0:k} = -\mathbf{b}^{(\ell)}_b \tag{5.20}$$

Once $\delta\mathbf{x}^{(\ell)}_{0:k}$ is found, the new state estimate is computed as:

$$\hat{\mathbf{x}}^{(\ell+1)}_{0:k|k} = \hat{\mathbf{x}}^{(\ell)}_{0:k|k} + \delta\mathbf{x}^{(\ell)}_{0:k} \tag{5.21}$$

Given an initial estimate $\hat{\mathbf{x}}^{(0)}_{0:k|k}$ that resides within the attraction basin of the global optimum, this iterative algorithm will compute the global minimum (i.e., MAP estimate) for the entire state given all measurements up to time-step k.

5.3 SWF-based SLAM

It is clear from the preceding section that, as the robot continuously moves and observes new landmarks, the size of the state vector of the batch-MAP estimator, $\mathbf{x}_{0:k}$, increases. Consequently, the computational cost of obtaining a state estimate continuously grows, and at some point it will inevitably become too high for real-time operation. In order to adapt to the available computational resources, marginalization [38, 132, 142] can be used to discard old, matured states. This results in a constant-cost SWF which maintains a constant-size window of

states [142, 143]. In this section, we describe the effects of the marginalization used by the standard SWF on the system's observability properties. This analysis forms the basis for our proposed algorithm (see Section 5.4). For more details on the derivation of the marginalization equations, the interested reader is referred to [38].

We consider a general scenario where marginalization of old states is carried out at time-step k_o, when all the measurements during the time interval $[0, k_o]$ are available. Subsequently the robot keeps moving and collects new measurements in the time interval $[k_o + 1, k]$, and estimation takes place again at time-step k. The old states that are marginalized out at time-step k_o are denoted by:

$$\mathbf{x_M} \triangleq \begin{bmatrix} \mathbf{x}_{R_{0:k_m}}^T & \mathbf{p}_{L_{M_1}}^T & \cdots & \mathbf{p}_{L_{M_m}}^T \end{bmatrix}^T$$

Note that it is not necessary to sequentially marginalize out the old robot poses, $\mathbf{x}_{R_{0:k_m}}$; instead, we can selectively discard the most matured (i.e., accurately estimated) ones. The remaining states that stay active in the sliding window after marginalization are denoted by:

$$\mathbf{x_R} \triangleq \begin{bmatrix} \mathbf{x}_{R_{k_m+1:k_o}}^T & \mathbf{p}_{L_{R_1}}^T & \cdots & \mathbf{p}_{L_{R_r}}^T \end{bmatrix}^T$$

Upon marginalization, all the states in $\mathbf{x_M}$, as well as all the measurements that involve these states (denoted by $\mathcal{Z}_\mathbf{M}$) are discarded. In their place, we maintain a Gaussian pdf, that describes the information that the discarded measurements convey about the active states, $\mathbf{x_R}$. The information matrix of this Gaussian is given by:

$$\mathbf{A}_p(k_o) = \mathbf{A_{RR}}(k_o) - \mathbf{A_{RM}}(k_o)\mathbf{A}_{\mathbf{MM}}^{-1}(k_o)\mathbf{A_{MR}}(k_o) \tag{5.22}$$

where the matrices appearing in the above equation are defined as the partitions of the following matrix:

$$\mathbf{A}_m(k_o) = \mathbf{\Pi}^T\mathbf{P}_{0|0}^{-1}\mathbf{\Pi} + \sum_{\kappa=0}^{k_m-1}\mathbf{\Phi}_\kappa^T(k_o)\mathbf{Q}_\kappa'^{-1}\mathbf{\Phi}_\kappa(k_o) + \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_\mathbf{M}}\mathbf{H}_{ij}^T(k_o)\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k_o) \tag{5.23}$$

$$=: \begin{bmatrix} \mathbf{A_{MM}}(k_o) & \mathbf{A_{MR}}(k_o) \\ \mathbf{A_{RM}}(k_o) & \mathbf{A_{RR}}(k_o) \end{bmatrix} \tag{5.24}$$

Close inspection reveals that $\mathbf{A}_m(k_o)$ is the matrix describing the information contained in all the discarded measurements (odometry, robot-to-landmark, and prior). Thus, $\mathbf{A}_p(k_o)$, which is the Schur complement of $\mathbf{A_{MM}}(k_o)$ in $\mathbf{A}_m(k_o)$, describes the information that the discarded measurements give us about $\mathbf{x_R}$. We also note that, in the above, the time index (k_o) has been

added to denote the fact that all the Jacobians are computed using the estimate $\hat{\mathbf{x}}_{0:k_o|k_o}$ as the linearization point.

After marginalization the robot continues moving in its environment, and new states are added to the state vector during $[k_o+1, k]$. These are denoted by:

$$\mathbf{x_N} \triangleq \begin{bmatrix} \mathbf{x}_{R_{k_o+1:k}}^T & \mathbf{p}_{L_{N_1}}^T & \cdots & \mathbf{p}_{L_{N_n}}^T \end{bmatrix}^T$$

Now, at time step k, the "active states" are $\mathbf{x_R}$ and $\mathbf{x_N}$. In order to compute estimates for the active states, the SWF employs the "active" measurements, $\mathcal{Z}_\mathbf{A} = \mathcal{Z}_{0:k} \setminus \mathcal{Z}_\mathbf{M}$, along with the motion model and the information from the marginalized states [expressed by $\mathbf{A}_p(k_o)$ in (5.22)] [38].

As described above, the key idea in the SWF is that the information of all the marginalized measurements is represented using a single Gaussian distribution. While this entails an approximation, it also enables the SWF to maintain constant computational complexity, that depends only on the number of currently active states, and not on the past history of marginalized states.

5.3.1 Parameter observability properties

We now examine the parameter observability properties [14] of the standard SWF-based SLAM, which, for the time being, is considered as a parameter (instead of state) estimation problem. The study of parameter observability examines whether the information provided by the available measurements is sufficient for estimating the parameters without ambiguity. When parameter observability holds, the Fisher information matrix (i.e., the Hessian matrix) is invertible. Since the Fisher information matrix describes the information available in the measurements, by studying its nullspace we can gain insight about the directions in the parameter (state) space along which the estimator acquires information. In what follows, we will compare the parameter observability properties of the standard SWF with those of the batch-MAP estimator, to draw conclusions about the estimator's consistency.

We first notice that the nullspace of the Hessian matrix of the batch-MAP estimator (5.19)

at time-step k, is given by:[1]

$$\text{null}\left(\mathbf{A}_b(k)\right) = \underset{\text{col.}}{\text{span}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{R_{0|k}} \\ \mathbf{0}_{1\times 2} & 1 \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{R_{k|k}} \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{L_{1|k}} \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{L_{M|k}} \end{bmatrix} \tag{5.25}$$

which is of dimension three. This agrees with the fact that in SLAM, three d.o.f. corresponding to the global translation and rotation are unobservable (see Chapter 2 and [72]). However, as shown below, this is not the case for the standard SWF.

In the standard SWF, the matrix that describes the information for the *entire* history of states, $\mathbf{x}_{0:k} = \begin{bmatrix} \mathbf{x}_{\mathbf{M}}^T & \mathbf{x}_{\mathbf{R}}^T & \mathbf{x}_{\mathbf{N}}^T \end{bmatrix}^T$, is given by [38]:

$$\begin{aligned} \mathbf{A}(k) = & \underbrace{\sum_{\kappa=0}^{k_m-1} \mathbf{\Phi}_\kappa^T(k_o)\mathbf{Q}_\kappa'^{-1}\mathbf{\Phi}_\kappa(k_o) + \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_{\mathbf{M}}} \mathbf{H}_{ij}^T(k_o)\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k_o)}_{\mathbf{A}_1(k_o)} \\ & + \underbrace{\sum_{\kappa=k_m}^{k-1} \mathbf{\Phi}_\kappa^T(k)\mathbf{Q}_\kappa'^{-1}\mathbf{\Phi}_\kappa(k) \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_{\mathbf{A}}} \mathbf{H}_{ij}^T(k)\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k)}_{\mathbf{A}_2(k)} \end{aligned} \tag{5.26}$$

where the matrix $\mathbf{A}_1(k_o)$ contains all the information pertaining to the marginalized states, and $\mathbf{A}_2(k)$ the information pertaining to the active states at time-step k. Again, we note that the time indices (k_o) and (k) indicate the state estimates ($\hat{\mathbf{x}}_{0:k_o|k_o}$ and $\hat{\mathbf{x}}_{0:k|k}$, respectively) used as linearization points in computing each of the above terms. The Hessian $\mathbf{A}(k)$ in (5.26) has the

[1] Since we are interested in the information contained in the available measurements, the case without prior (i.e., $\mathbf{P}_{0|0} \to \boldsymbol{\infty}$) is considered here.

following interesting structure:

$$\mathbf{A}(k) = \underbrace{\begin{bmatrix} \mathbf{A_{MM}}(k_o) & \mathbf{A_{MR}}(k_o) & \mathbf{0} \\ \mathbf{A_{RM}}(k_o) & \mathbf{A_{RR}}(k_o) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_1(k_o)} + \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_{RR}}(k) & \mathbf{A_{RN}}(k) \\ \mathbf{0} & \mathbf{A_{NR}}(k) & \mathbf{A_{NN}}(k) \end{bmatrix}}_{\mathbf{A}_2(k)}$$
$$= \begin{bmatrix} \mathbf{A_{MM}}(k_o) & \mathbf{A_{MR}}(k_o) & \mathbf{0} \\ \mathbf{A_{RM}}(k_o) & \mathbf{A_{RR}}(k_o) + \mathbf{A_{RR}}(k) & \mathbf{A_{RN}}(k) \\ \mathbf{0} & \mathbf{A_{NR}}(k) & \mathbf{A_{NN}}(k) \end{bmatrix} \tag{5.27}$$

It is clear now that different estimates, $\hat{\mathbf{x}}_{\mathbf{R}}(k_o)$ and $\hat{\mathbf{x}}_{\mathbf{R}}(k)$, are used in computing the Hessian matrix. This occurs because some of the states in $\mathbf{x_R}$ are involved in measurements both in $\mathcal{Z}_{\mathbf{M}}$ and $\mathcal{Z}_{\mathbf{A}}$. As a result of the above structure, it can be shown that the last column of the matrix in (5.25) does *not* belong in the nullspace of $\mathbf{A}(k)$ [38]. Instead, the nullspace of $\mathbf{A}(k)$ is spanned by only the first two columns of (5.25), which in turn shows that the rank of the Hessian in the SWF is *higher* than the rank of the Hessian of the batch MAP. Clearly, this difference is not desirable, since both estimators process the same measurements, and thus have access to the same amount of information.

5.4 Observability-Constrained (OC)-SWF SLAM

As seen from the preceding section, due to marginalization, the standard SWF possesses different parameter observability properties from the batch-MAP estimator, since its Hessian has a nullspace of lower dimension than that of the batch-MAP estimator. This implies that the standard SWF acquires spurious information along one direction of the state space (the one corresponding to global orientation), which can lead to inconsistency. To address this issue, we adopt the idea of observability-based rules for choosing linearization points that was originally proposed in our previous work [72] (also see Section 2.5), and develop a new Observability-Constrained (OC)-SWF within the smoothing framework.

The key idea of the proposed approach is that the linearization points used in computing the Hessian matrix are selected so as to ensure that the Hessian has a nullspace of the same dimension as that of the batch-MAP estimator [see (5.25)]. Different approaches for selecting linearization points are possible to satisfy this observability condition. For example, the prior-linearization (PL)-SWF proposed in [38] employs a simple linearization scheme to achieve

this goal based on [70]. Specifically, when computing the Hessian, it uses the prior estimates, $\hat{\mathbf{x}}_{\mathbf{R}}(k_o)$, instead of the current estimates $\hat{\mathbf{x}}_{\mathbf{R}}(k)$, for the states in $\mathbf{x}_{\mathbf{R}}$ that are connected to marginalized states. By doing so, it is guaranteed that the same estimate is used as the linearization point for each of these states. However, even though the PL-SWF typically performs substantially better than the standard SWF (see Section 5.5), the prior estimates $\hat{\mathbf{x}}_{\mathbf{R}}(k_o)$ used as linearization points could be inaccurate, and thus can result in large linearization errors, which can degrade the estimator's performance. Therefore, in the proposed OC-SWF, we select the linearization points for the states $\mathbf{x}_{\mathbf{R}}$ and $\mathbf{x}_{\mathbf{N}}$ (i.e., the states that are still "active" in the minimization), in a way that not only ensures the correct dimension for the nullspace of the Hessian matrix, but also minimizes their difference from the current best available estimates (see [72]). This can be formulated as the following constrained minimization problem:[2]

$$\min_{\mathbf{x}^\star_{\mathbf{R}},\mathbf{x}^\star_{\mathbf{N}}} \; ||\mathbf{x}^\star_{\mathbf{R}} - \hat{\mathbf{x}}_{\mathbf{R}}(k)||^2 + ||\mathbf{x}^\star_{\mathbf{N}} - \hat{\mathbf{x}}_{\mathbf{N}}(k)||^2 \tag{5.28}$$

$$\text{subject to} \;\; \mathbf{A}(k)\mathbf{N}_k = \mathbf{0} \tag{5.29}$$

In this formulation, $\mathbf{N}_k$ is a design choice that defines the desired nullspace with correct dimension. Ideally, we would like to have the same nullspace as (5.25). However, this is not possible, as in the SWF some of the old states have been marginalized, and thus we do not maintain the up-to-date estimates for them. We next describe our choice of estimates used for constructing $\mathbf{N}_k$, and denote these estimates by the symbol "$\bar{}$". Specifically, during the $(\ell+1)$-th Gauss-Newton iteration, we use the following estimates to construct the matrix $\mathbf{N}_k$: (i) For the new states, $\mathbf{x}_{\mathbf{N}}$, as well as those states in $\mathbf{x}_{\mathbf{R}}$ for which no prior exists, we use the estimates from the ℓ-th iteration, i.e., $\bar{\mathbf{x}}_i = \hat{\mathbf{x}}_i(k)$; (ii) For all marginalized states, $\mathbf{x}_{\mathbf{M}}$, as well as for states in $\mathbf{x}_{\mathbf{R}}$ for which a prior exists, we use the prior estimate, i.e., $\bar{\mathbf{x}}_i = \hat{\mathbf{x}}_i(k_o)$. By replacing the pertinent state estimates in (5.25) by the estimates selected above, $\bar{\mathbf{x}}_{0:k} = \begin{bmatrix}\bar{\mathbf{x}}_{\mathbf{M}}^T & \bar{\mathbf{x}}_{\mathbf{R}}^T & \bar{\mathbf{x}}_{\mathbf{N}}^T\end{bmatrix}^T$, we obtain the desired nullspace, $\mathbf{N}_k = \mathbf{N}_k(\bar{\mathbf{x}}_{0:k})$.

By construction, the nullspace $\mathbf{N}_k(\bar{\mathbf{x}}_{0:k})$ always satisfies the equality $\mathbf{A}_1(k_o)\mathbf{N}_k = \mathbf{0}$. Thus,

[2] For the clarity of presentation, hereafter the superscript (ℓ) is dropped, since, without loss of generality, we consider the $(\ell+1)$-th iteration in Gauss-Newton given that the results from the ℓ-th iteration are available.

the condition $\mathbf{A}(k)\mathbf{N}_k(\bar{\mathbf{x}}_{0:k}) = \mathbf{0}$ can be written as [see (5.26)]:

$$\begin{aligned}
&\mathbf{A}_2(k)\mathbf{N}_k = \mathbf{0} \\
\Rightarrow &\left(\sum_{\kappa=k_m}^{k-1} \mathbf{\Phi}_\kappa^T \mathbf{Q}_\kappa'^{-1} \mathbf{\Phi}_\kappa + \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_\mathbf{A}} \mathbf{H}_{ij}^T \mathbf{R}_{ij}^{-1} \mathbf{H}_{ij} \right) \mathbf{N}_k = \mathbf{0} \\
\Rightarrow &\begin{cases} \mathbf{\Phi}_\kappa \mathbf{N}_k = \mathbf{0}\ , \ \ \forall \kappa = k_m, \ldots, k-1 \\ \mathbf{H}_{ij}\mathbf{N}_k = \mathbf{0}\ , \ \ \forall \mathbf{z}_{ij} \in \mathcal{Z}_\mathbf{A} \end{cases}
\end{aligned} \tag{5.30}$$

Using the structure of the Jacobians $\mathbf{\Phi}_k$ and $\mathbf{H}_{ij}$ [see (5.5) and (5.9)] and that of the matrix $\mathbf{N}_k$ (5.25), the above constraints (5.30) can be written as follows:

$$\mathbf{\Phi}_\kappa \mathbf{N}_k = \mathbf{0} \Rightarrow \mathbf{p}^\star_{R_\kappa} - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{R_{\kappa+1}} - \mathbf{p}^\star_{R_{\kappa+1}} = \mathbf{0} \tag{5.31}$$

$$\mathbf{H}_{ij}\mathbf{N}_k = \mathbf{0} \Rightarrow \mathbf{p}^\star_{R_j} - \bar{\mathbf{p}}_{R_j} + \bar{\mathbf{p}}_{L_i} - \mathbf{p}^\star_{L_i} = \mathbf{0} \tag{5.32}$$

Therefore, the problem (5.28)-(5.29) can be simplified as:

$$\min_{\mathbf{x}^\star_\mathbf{R}, \mathbf{x}^\star_\mathbf{N}} \ ||\mathbf{x}^\star_\mathbf{R} - \hat{\mathbf{x}}_\mathbf{R}(k)||^2 + ||\mathbf{x}^\star_\mathbf{N} - \hat{\mathbf{x}}_\mathbf{N}(k)||^2 \tag{5.33}$$

$$\text{subject to } \begin{cases} \mathbf{p}^\star_{R_\kappa} - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{R_{\kappa+1}} - \mathbf{p}^\star_{R_{\kappa+1}} = \mathbf{0}\ , \ \forall \kappa = k_m, \ldots, k-1 \\ \mathbf{p}^\star_{R_j} - \bar{\mathbf{p}}_{R_j} + \bar{\mathbf{p}}_{L_i} - \mathbf{p}^\star_{L_i} = \mathbf{0}\ , \ \forall \mathbf{z}_{ij} \in \mathcal{Z}_\mathbf{A} \end{cases} \tag{5.34}$$

We now derive an analytical solution to the constrained minimization problem (5.33)-(5.34). In particular, the approach of Lagrangian multipliers [16] is employed. The Lagrangian function is constructed as follows:

$$\begin{aligned}
\mathcal{L} = &||\mathbf{x}^\star_\mathbf{R} - \hat{\mathbf{x}}_\mathbf{R}(k)||^2 + ||\mathbf{x}^\star_\mathbf{N} - \hat{\mathbf{x}}_\mathbf{N}(k)||^2 \\
&+ \sum_{\kappa=k_m}^{k-1} \boldsymbol{\mu}_\kappa^T \left(\mathbf{p}^\star_{R_\kappa} - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{R_{\kappa+1}} - \mathbf{p}^\star_{R_{\kappa+1}} \right) + \sum_{(i,j),\mathbf{z}_{ij}\in\mathcal{Z}_\mathbf{A}} \boldsymbol{\lambda}_{ij}^T \left(\mathbf{p}^\star_{R_j} - \bar{\mathbf{p}}_{R_j} + \bar{\mathbf{p}}_{L_i} - \mathbf{p}^\star_{L_i} \right)
\end{aligned} \tag{5.35}$$

By setting the derivatives with respect to the state and Lagrangian-multiplier variables equal to zero, we have:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{p}^\star_{R_\kappa}} = \begin{cases} 2(\mathbf{p}^\star_{R_\kappa} - \hat{\mathbf{p}}_{R_{\kappa|k}}) + \boldsymbol{\mu}_\kappa + \sum\limits_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_\mathbf{A}} \boldsymbol{\lambda}_{i\kappa} = \mathbf{0}\ , & \text{if } \kappa = k_m \\ 2(\mathbf{p}^\star_{R_\kappa} - \hat{\mathbf{p}}_{R_{\kappa|k}}) - \boldsymbol{\mu}_{\kappa-1} + \sum\limits_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_\mathbf{A}} \boldsymbol{\lambda}_{i\kappa} = \mathbf{0}\ , & \text{if } \kappa = k \\ 2(\mathbf{p}^\star_{R_\kappa} - \hat{\mathbf{p}}_{R_{\kappa|k}}) + \boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_{\kappa-1} + \sum\limits_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_\mathbf{A}} \boldsymbol{\lambda}_{i\kappa} = \mathbf{0}\ , & \text{else} \end{cases} \tag{5.36}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{p}^{\star}_{L_i}} = 2(\mathbf{p}^{\star}_{L_i} - \hat{\mathbf{p}}_{L_{i|k}}) - \sum_{j,\mathbf{z}_{ij}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{ij} = \mathbf{0} \tag{5.37}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_{\kappa}} = \mathbf{p}^{\star}_{R_\kappa} - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{R_{\kappa+1}} - \mathbf{p}^{\star}_{R_{\kappa+1}} = \mathbf{0} \tag{5.38}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}_{ij}} = \mathbf{p}^{\star}_{R_j} - \bar{\mathbf{p}}_{R_j} + \bar{\mathbf{p}}_{L_i} - \mathbf{p}^{\star}_{L_i} = \mathbf{0} \tag{5.39}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}^{\star}_{\text{other}}} = 2(\mathbf{x}^{\star}_{\text{other}} - \hat{\mathbf{x}}_{\text{other}}(k)) = \mathbf{0} \tag{5.40}$$

where $\mathbf{x}_{\text{other}}$ denotes all the state variables except the ones involved in (5.36)-(5.39). Solving (5.36), (5.37), and (5.40) yields the following optimal solutions:

$$\mathbf{p}^{\star}_{R_\kappa} = \hat{\mathbf{p}}_{R_{\kappa|k}} - \frac{1}{2}\left[\boldsymbol{\delta\mu}_{\kappa} + \sum_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{i\kappa}\right] \tag{5.41}$$

$$\mathbf{p}^{\star}_{L_i} = \hat{\mathbf{p}}_{L_{i|k}} + \frac{1}{2}\left[\sum_{j,\mathbf{z}_{ij}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{ij}\right] \tag{5.42}$$

$$\mathbf{x}^{\star}_{\text{other}} = \hat{\mathbf{x}}_{\text{other}}(k) \tag{5.43}$$

where

$$\boldsymbol{\delta\mu}_{\kappa} = \begin{cases} \boldsymbol{\mu}_{\kappa} \ , & \text{if } \kappa = k_m \\ -\boldsymbol{\mu}_{\kappa-1} \ , & \text{if } \kappa = k \\ \boldsymbol{\mu}_{\kappa} - \boldsymbol{\mu}_{\kappa-1} \ , & \text{else} \end{cases}$$

Substituting (5.41)-(5.43) into (5.38) and (5.39) yields the following *linear* equations in terms of the Lagrangian multipliers:

$$\begin{aligned} &\boldsymbol{\Delta\mu}_{\kappa} + \sum_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{i\kappa} - \sum_{i,\mathbf{z}_{i(\kappa+1)}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{i(\kappa+1)} \\ &= 2\left(\hat{\mathbf{p}}_{R_{\kappa|k}} - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{R_{\kappa+1}} - \hat{\mathbf{p}}_{R_{\kappa+1|k}}\right)\boldsymbol{\delta\mu}_{\kappa} + \sum_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{i\kappa} + \sum_{j,\mathbf{z}_{ij}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{ij} \qquad (5.44) \\ &= 2\left(\hat{\mathbf{p}}_{R_{\kappa|k}} - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{L_i} - \hat{\mathbf{p}}_{L_{i|k}}\right) \qquad (5.45) \end{aligned}$$

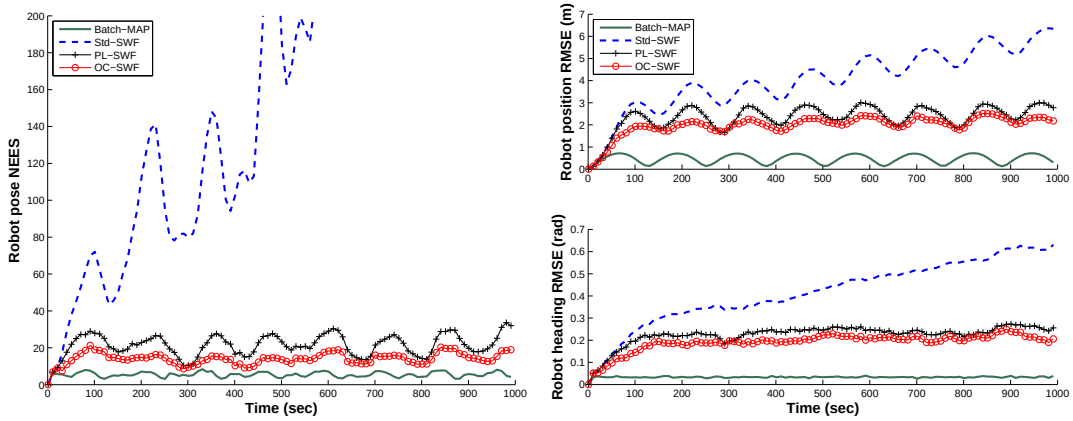
where

$$\boldsymbol{\Delta\mu}_\kappa = \begin{cases} 2\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_{\kappa+1} \ , & \text{if } \kappa = k_m \\ 2\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_{\kappa-1} \ , & \text{if } \kappa = k-1 \\ -\boldsymbol{\mu}_{\kappa-1} \ , & \text{if } \kappa = k \\ 2\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_{\kappa-1} - \boldsymbol{\mu}_{\kappa+1} \ , & \text{else} \end{cases}$$

In order to determine the Lagrangian multipliers, $\boldsymbol{\mu}_\kappa$ and $\boldsymbol{\lambda}_{ij}$, we stack equations (5.44)-(5.45) for all the measurements (constraints) into matrix-vector form and solve the resulting linear system. Once the Lagrangian multipliers are specified, the *optimal* linearization points can be obtained based on (5.41)-(5.43). Subsequently, the Jacobian and Hessian matrices are computed using the optimal linearization points, and then the standard Gauss-Newton steps are carried out (see Section 5.2.3). It should be pointed out that, as compared to the standard SWF and the PL-SWF, the OC-SWF only requires an additional computational overhead of linearly solving for the Lagrangian multipliers, which in general is cubic in the number of active proprioceptive and exteroceptive measurements.

5.5 Simulation results

A series of Monte-Carlo simulations were conducted under different conditions, in order to validate the capability of the proposed OC-SWF to improve estimation performance. As before, the metrics used to evaluate the estimator's performance were RMSE and NEES [14]. In simulation tests presented in this section, we conducted 50 Monte-Carlo simulations, and compared four different estimators: (1) the batch-MAP estimator, (2) the standard SWF, (3) the PL-SWF [38], and (4) the proposed OC-SWF. In the simulation setup, a robot with a simple 3-wheel (2 active and 1 caster) kinematic model moves on a planar surface, at a constant velocity of $v = 0.5$ m/sec. The two active wheels are equipped with encoders, which measure their revolutions and provide measurements of velocity (i.e., right and left wheel velocities, v_r and v_l, respectively), with standard deviation equal to $\sigma = 1\%v$ for each wheel. These measurements are used to obtain linear and rotational velocity measurements for the robot, which are given by $v = \frac{v_r+v_l}{2}$ and $\omega = \frac{v_r-v_l}{a}$, where $a = 0.5$ m is the distance between the active wheels. The standard deviation of the linear and rotational velocity measurement noise is thus equal to $\sigma_v = \frac{\sigma}{\sqrt{2}}$ and $\sigma_\omega = \frac{\sqrt{2}\sigma}{a}$, respectively. We considered a SLAM scenario where a robot



(a) NEES of robot pose

(b) RMSE of robot pose

Figure 5.1: Monte-Carlo simulation results. It is clear that both the PL-SWF and OC-SWF perform significantly better than the standard SWF, in terms of both consistency (NEES) and accuracy (RMSE). Note also that the OC-SWF attains better performance than the PL-SWF.

moves along a circular trajectory of total length of about 500 m, and measures bearing angles to landmarks that lie within its sensing range of 10 m. There are 50 landmarks in total which are randomly generated along the robot trajectory. This can arise, for example, in the case in which a robot moves inside corridors and tracks its position and corners (landmarks) using a monocular camera. At each time step, approximately 10 landmarks are visible. In the SWFs we chose to maintain a sliding window comprising 20 robot poses and at most 10 active landmarks. To ensure a fair comparison among the SWF algorithms, all three of them process the same data and maintain the same states in their windows. In this simulation, the landmarks to be marginalized are chosen such that at least two "old" landmarks always remain in the window, to ensure that the uncertainty does not continuously increase. The batch-MAP estimator processes all measurements, and is used as the benchmark.

For the results presented here, we considered a case with relatively large measurement noise, compared to what is typically encountered in practice, since larger noise levels can lead to larger estimation errors, and thus less accurate linearization, which will make the effects of inconsistency more apparent. Specifically, the standard deviation of the bearing measurement noise was set to 10 deg. Fig. 5.1 shows the results for the robot pose based on the compared estimators, while Table 5.1 depicts the average NEES and RMSE for the landmark positions (averaged over all the landmarks). First notice that as expected, the batch-MAP estimator attains

Table 5.1: Landmark position estimation performance

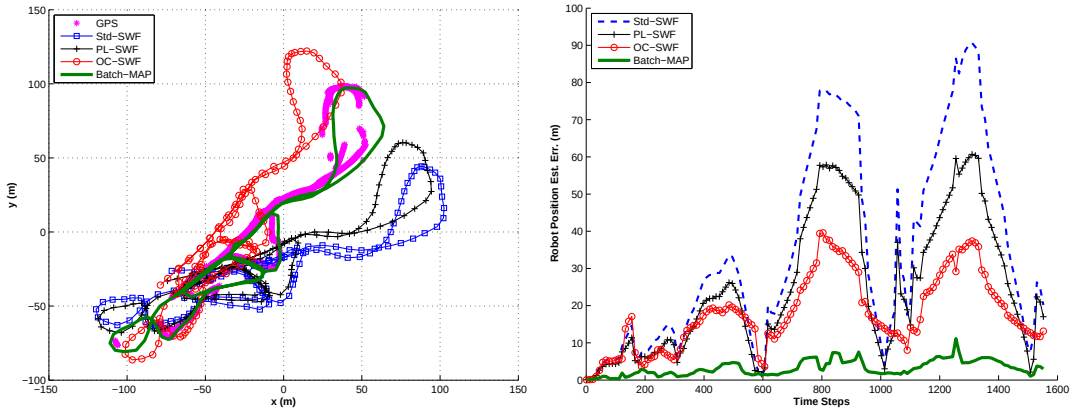
Batch-MAP	Std-SWF	PL-SWF	OC-SWF
RMSE for Landmark Position (m)			
0.5184	2.7449	2.6713	2.6235
NEES for Landmark Position			
3.7769	42.1306	12.3615	9.5719

the best performance, since it utilizes all the available information, while the SWFs discard the measurements belonging to the inactive measurement set, $\mathcal{Z}_{\mathrm{M}}$, due to marginalization. More importantly, the two observability-constrained smoothers (i.e., PL-SWF and OC-SWF) perform substantially better than the standard SWF, in terms of both consistency (NEES) and accuracy (RMSE). This is attributed to the fact that the appropriate parameter observability properties are preserved in the proposed observability-based smoothing framework. We also note that the OC-SWF achieves better performance than the PL-SWF. This is due to the fact that when the noise is large, the prior estimates used as linearization points in the PL-SWF are inaccurate (i.e., the linearization errors become significant), which degrades the estimator's performance. In contrast, the OC-SWF employs, by construction, the optimal linearization points and thus yields better estimation accuracy.

5.6 Experimental results

To experimentally validate the performance of the OC-SWF, the estimator was tested on the original Victoria Park dataset courtesy of Nebot and Guivant.[3] The experimental platform was a 4-wheeled vehicle equipped with a kinematic GPS, a laser sensor, and wheel encoders. The GPS system was used to provide ground truth for the robot position. Wheel encoders were used to provide odometric measurements, and propagation was carried out using the Ackerman model. In this particular application, since the most common feature in the environment were trees, the profiles of trees were extracted from the laser data, the centers of the trunks were then

[3] It is available at: `http://www-personal.acfr.usyd.edu.au/nebot/victoria_park.htm`. Note that, to ensure the comparison to the batch MAP estimator, we here considered the first half of the dataset.



(a) (b)

Figure 5.2: Experimental results: (a) The robot trajectory estimates as compared to the GPS data, and (b) estimation errors of robot position. It is clear that the OC-SWF performs more accurately than the standard SWF and the PL-SWF.

used as the point landmarks, and distance and bearing measurements to them were used for estimation [56].

In this test, we compared the same four estimators as in the preceding simulation: (1) the batch-MAP estimator, (2) the standard SWF, (3) the PL-SWF [38], and (4) the proposed OC-SWF. Since in this experiment, both true landmark positions and true robot orientations were unavailable, we only compared the robot position estimation performance, which is shown in Fig. 5.2. Specifically, Fig. 5.2(a) depicts the trajectory estimates produced by the four estimators as compared to the GPS ground truth, while Fig. 5.2(b) shows the estimation errors of the robot position over time. Note that since the GPS satellite signal was not always available, we computed the estimation errors only at the times when the GPS was available. As evident from Fig. 5.2, the OC-SWF performs significantly better than the standard SWF and the PL-SWF [38]. These results, along with those of the simulations presented in the previous section, show that it is essential for an estimator to ensure appropriate observability properties in order to improve its performance.

5.7 Summary

In this chapter, we addressed the consistency issue of the standard SWF. Even though the SWF is an appealing smoothing algorithm well-suited for real-time applications where the effects of nonlinearity of the measurements are significant, it can suffer from inconsistency. In particular, due to marginalization, the standard SWF uses different estimates for the same states as linearization points when computing the Hessian matrix, which results in its Hessian having a nullspace of lower dimension than the batch-MAP estimator. This implies that the standard SWF acquires spurious information and thus may become inconsistent. To address this issue, we have introduced an observability-based smoothing framework, which extends the methods presented in Chapter 2 for EKFs to the case of the SWF. Specifically, we select the linearization points at which the Hessian is evaluated, so as to ensure that the nullspace of the Hessian is of the same dimension as that of the batch-MAP estimator, while minimizing the linearization errors. Both simulation and experimental results have shown that the proposed OC-SWF performs better than the standard SWF as well as the PL-SWF [38], in terms of both accuracy and consistency.

Chapter 6

Towards General Nonlinear Systems with Partial-State Measurements

In this chapter, we study filter consistency for a broad class of nonlinear systems, i.e., observable nonlinear systems with partial-state measurements from different sources (sensors). In particular, we discover that despite the observability of such a general system, the standard EKF often become inconsistent due to the fact that it acquires spurious information from the measurements of each source. To address this issue, we adapt the observability-based methodology presented in the previous chapters, and develop new EKF algorithms to improve consistency by enforcing the filter to gain information from each source's measurements only along the correct directions of the state space. Part of this chapter has been published in [73].

6.1 Introduction

Nonlinear filtering problems arise in numerous science and engineering fields, such as economics [34], statistical signal processing [91], radar tracking [17], and navigation and guidance systems [158]. These problems consist of estimating the state of a nonlinear stochastic system from noisy measurements. Although the EKF has been successfully applied to many nonlinear filtering problems, if the nonlinearities are significant, its variants are often used to improve performance. For example, the IEKF [14] iterates the filter update till convergence, by iteratively relinearizing the measurement function. Alternatively, the UKF [84] deterministically samples

the nonlinear function around the current state estimate, thus improving the linear approximation by linear regression. Nonlinear filtering problems are challenging for a number of reasons, and one particular difficulty is the inconsistency issue. In particular, no provably consistent filter can be constructed for a nonlinear system, and the consistency of every filter has to be evaluated experimentally. As defined in [14], a state filter is consistent if the estimation errors are zero-mean and have covariance smaller or equal to the one calculated by the filter. Consistency is one of the primary criteria for evaluating the performance of any filter; if a filter is inconsistent, then its estimation accuracy is unknown, which in turn makes the filter unreliable. In effect, significant empirical evidence shows that the standard EKF becomes inconsistent in nonlinear filtering problems such as robot localization [12, 27, 72, 75, 79, 85] (also see Chapters 2 and 3). The lack of understanding filter consistency and of algorithms for improving consistency is clearly a significant limitation.

In this chapter, we revisit the problem of filter consistency for a broad class of discrete-time nonlinear systems, by examining the directions of state space along which information is available from measurements of each source (sensor). Based on this analysis, we propose a novel methodology to improve consistency by ensuring that the filter acquires information from each source's measurements only along the correct directions of the state space.

In particular, the Fisher information matrix (FIM) [14] for given measurements encapsulates all available information about the entire state of a stochastic system. By marginalizing all but the initial state, we obtain the corresponding FIM that contains all information available in the measurements for determining the initial state. Studying the FIM's structure reveals the directions along which information is (un)available from the measurements. These can be exploited in the design of nonlinear estimation algorithms, i.e., enforcing estimators to gain information from measurements only along correct directions. Moreover, we show that the FIM of the initial state can be factorized in terms of the observability matrix of the corresponding deterministic system, and that these two matrices have same rank properties. Based on this key finding, in order to ensure consistent estimation, we impose the constraint of acquiring information along the correct directions on a novel decomposition (according to the different measurement sources) of the observability matrix, instead of the FIM. To this end, we introduce two different EKF algorithms that compute the appropriate filter Jacobians, either directly (i.e., by projecting the best-available Jacobians onto the information-available subspace) or indirectly (i.e., by first finding the optimal linearization points for computing the Jacobians). As a result, only

information actually available from each source's measurements is gained, which substantially improves the estimation consistency and accuracy, as opposed to the standard EKF.

6.2 Methodology

We consider a general discrete-time nonlinear system of the following form:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \tag{6.1}$$

$$\mathbf{z}_{i,k} = \mathbf{h}(\mathbf{x}_k, \mathbf{s}_{i,k}) + \mathbf{v}_{i,k}\ ,\ \ i \in \{1, \ldots, s\} \tag{6.2}$$

where $\mathbf{x}_k \in \mathcal{R}^n$ denotes the state of the system, $\mathbf{u}_k \in \mathcal{R}^n$ is the control input, and $\mathbf{w}_k \in \mathcal{R}^n$ is zero-mean white Gaussian process noise, i.e., $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$. $\mathbf{z}_{i,k} \in \mathcal{R}^m$ is the measurement taken from the i-th ($i \in \{1, \ldots, s\}$) measurement source (e.g., sensor), and is generally (although not necessarily) of *lower* dimension than the state vector, i.e., $m < n$, which is the case of partial-state measurements we consider in this work. The parameter $\mathbf{s}_{i,k}$ denotes the known parameters of the i-th measurement source, such as the sensor's location or a binary indicator of the availability of the i-th measurement. The random variable $\mathbf{v}_{i,k} \in \mathcal{R}^m$ is zero-mean white Gaussian measurement noise, i.e., $\mathbf{v}_{i,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{i,k})$.

We employ the EKF to recursively compute the state estimate and error covariance. Specifically, we linearize the nonlinear system at the linearization points, $\mathbf{x}^\star_{k|k-1}$ and $\mathbf{x}^\star_{k|k}$ (i.e., the linearization points before and after the update at time-step k) [see (6.1), (6.2)] and obtain the following linearized error-state system:

$$\tilde{\mathbf{x}}_{k+1|k} = \mathbf{\Phi}_k \tilde{\mathbf{x}}_{k|k} + \mathbf{w}_k \tag{6.3}$$

$$\tilde{\mathbf{z}}_{i,k|k-1} = \mathbf{H}_{i,k} \tilde{\mathbf{x}}_{k|k-1} + \mathbf{v}_{i,k}\ ,\ \ i \in \{1, \ldots, s\} \tag{6.4}$$

where

$$\mathbf{\Phi}_k = \nabla_{\mathbf{x}_k} \mathbf{f}\Big|_{\{\mathbf{x}^\star_{k|k}, \mathbf{u}_k\}}\ ,\ \ \mathbf{H}_{i,k} = \nabla_{\mathbf{x}_k} \mathbf{h}\Big|_{\{\mathbf{x}^\star_{k|k-1}\}} \tag{6.5}$$

The standard choice of linearization point is the current (and thus best) state estimate, which, however, as will be shown is not necessarily the best choice. Once the propagation and measurement Jacobians are computed, we propagate and update the state estimate and covariance,

respectively, as follows [14]:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k) \tag{6.6}$$

$$\mathbf{P}_{k+1|k} = \mathbf{\Phi}_k \mathbf{P}_{k|k} \mathbf{\Phi}_k^T + \mathbf{Q}_k \tag{6.7}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{r}_k \tag{6.8}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T \tag{6.9}$$

where $\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_{i,k}^T \mathbf{S}_k^{-1}$ is the Kalman gain, $\mathbf{r}_k = \mathbf{z}_{i,k} - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{s}_{i,k})$ is the measurement residual, and $\mathbf{S}_k = \mathbf{H}_{i,k}\mathbf{P}_{k|k-1}\mathbf{H}_{i,k}^T + \mathbf{R}_{i,k}$ is the corresponding residual covariance.

6.2.1 Observability and Fisher information

Since the EKF is constructed based on the linearized system [see (6.3) and (6.4)], it is important to study the observability properties of the corresponding deterministic system (i.e., noise free). Observability examines whether the information provided by the available measurements is sufficient for estimating the initial state without ambiguity. In particular, the observability matrix [30, 112] for the linearized system (6.3)-(6.4) during the time interval $[0, k]$ is defined by [see (2.22)]:

$$\mathbf{M} = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \mathbf{\Phi}_0 \\ \vdots \\ \mathbf{H}_k \mathbf{\Phi}_{k-1} \cdots \mathbf{\Phi}_0 \end{bmatrix} \tag{6.10}$$

If the system is observable, then the corresponding observability matrix $\mathbf{M}$ is full-rank.

The FIM [14] is closely related to the system observability and precisely describes the information available in the measurements. Thus, by studying its properties, we can also gain insight about the directions in the state space along which information is actually available. To this end, we examine the structure of the Hessian (information) matrix of the corresponding batch-MAP estimation over the time interval $[0, k]$, which is known to be optimal [91]. In what follows, we show that the FIM of the initial state $\mathbf{x}_0$ (obtained by marginalization) has the same properties as the observability matrix, which motivates us to instead examine the observability matrix in our analysis.

As discussed in Section 5.2.3, the optimal batch-MAP estimator utilizes all available information to estimate the *entire* state trajectory that is formed by stacking all states in the time

interval $[0, k]$:

$$\mathbf{x}_{0:k} = \begin{bmatrix} \mathbf{x}_0^T & \mathbf{x}_1^T & \cdots & \mathbf{x}_k^T \end{bmatrix}^T \tag{6.11}$$

Specifically, the batch-MAP estimator seeks to determine the entire state-space trajectory estimate $\hat{\mathbf{x}}_{0:k|k}$ by maximizing the following posterior pdf (assuming *no* prior is available):

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k}) \propto \prod_{\kappa=0}^{k-1} p(\mathbf{x}_{\kappa+1}|\mathbf{x}_\kappa) \prod_{\kappa=0}^{k} p(\mathbf{z}_{i,\kappa}|\mathbf{x}_\kappa) \tag{6.12}$$

where $\mathbf{z}_{0:k}$ denotes all the sensor measurements in the time interval $[0, k]$. In the above expression, we have employed the assumption of independent state and measurement noise and the Markovian property of the system dynamics [see (6.1) and (6.2), respectively]. Moreover, using the assumption of Gaussian noise, the above posterior pdf (6.12) can be written as:

$$\begin{aligned} &p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k}) \propto \\ &\prod_{\kappa=0}^{k-1} \frac{1}{\sqrt{|2\pi\mathbf{Q}_\kappa|}} \exp\left(-\frac{1}{2}||\mathbf{x}_{\kappa+1} - \mathbf{f}(\mathbf{x}_\kappa, \mathbf{u}_\kappa)||^2_{\mathbf{Q}_\kappa}\right) \times \\ &\prod_{\kappa=0}^{k} \frac{1}{\sqrt{|2\pi\mathbf{R}_{i,\kappa}|}} \exp\left(-\frac{1}{2}||\mathbf{z}_{i,\kappa} - \mathbf{h}(\mathbf{x}_\kappa, \mathbf{s}_{i,\kappa})||^2_{\mathbf{R}_{i,\kappa}}\right) \end{aligned} \tag{6.13}$$

Due to the monotonicity of the negative logarithm, the maximization of (6.13) is equivalent to the minimization of the following cost function under mild assumptions:

$$c(\mathbf{x}_{0:k}) = \sum_{\kappa=0}^{k-1} \frac{1}{2}||\mathbf{x}_{\kappa+1} - \mathbf{f}(\mathbf{x}_\kappa, \mathbf{u}_\kappa)||^2_{\mathbf{Q}_\kappa} + \sum_{\kappa=0}^{k} \frac{1}{2}||\mathbf{z}_{i,\kappa} - \mathbf{h}(\mathbf{x}_\kappa, \mathbf{s}_{i,\kappa})||^2_{\mathbf{R}_{i,\kappa}} \tag{6.14}$$

The Hessian (information) matrix is computed as [see (5.19)]:

$$\mathbf{A} = \sum_{\kappa=0}^{k-1} \mathcal{F}_\kappa^T \mathbf{Q}_\kappa^{-1} \mathcal{F}_\kappa + \sum_{\kappa=0}^{k} \mathcal{H}_\kappa^T \mathbf{R}_\kappa^{-1} \mathcal{H}_\kappa \tag{6.15}$$

with

$$\mathcal{F}_\kappa = \begin{bmatrix} \mathbf{0} & \cdots & -\mathbf{\Phi}_\kappa & \mathbf{I}_n & \cdots & \mathbf{0} \end{bmatrix} \tag{6.16}$$

$$\mathcal{H}_\kappa = \begin{bmatrix} \mathbf{0} & \cdots & -\mathbf{H}_\kappa & \cdots & \mathbf{0} \end{bmatrix} \tag{6.17}$$

where $\mathbf{H}_\kappa = \mathbf{H}_{i,\kappa}$ and $\mathbf{R}_\kappa = \mathbf{R}_{i,\kappa}$, if the i-th source provides the measurement at time-step κ, i.e., $\mathbf{z}_\kappa = \mathbf{z}_{i,\kappa}$ (note that hereafter we will use these notations interchangeably). It is important

to note that due to the sparse structure of $\mathcal{F}_\kappa$ and $\mathcal{H}_\kappa$ [see (6.16) and (6.17)], the FIM (6.15) has banded structure:

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{\Phi}_0^T\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0 + \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0 & -\boldsymbol{\Phi}_0^T\mathbf{Q}_0^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0 & \mathbf{Q}_0^{-1} + \boldsymbol{\Phi}_1^T\mathbf{Q}_1^{-1}\boldsymbol{\Phi}_1 + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1 & -\boldsymbol{\Phi}_1^T\mathbf{Q}_1^{-1} & \cdots & \mathbf{0} \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \cdots & -\mathbf{Q}_{k-2}^{-1}\boldsymbol{\Phi}_{k-2} & \mathbf{Q}_{k-2}^{-1} + \boldsymbol{\Phi}_{k-1}^T\mathbf{Q}_{k-1}^{-1}\boldsymbol{\Phi}_{k-1} + \mathbf{H}_{k-1}^T\mathbf{R}_{k-1}^{-1}\mathbf{H}_{k-1} & -\boldsymbol{\Phi}_{k-1}^T\mathbf{Q}_{k-1}^{-1} \\ \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{Q}_{k-1}^{-1}\boldsymbol{\Phi}_{k-1} & \mathbf{Q}_{k-1}^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k \end{bmatrix} \tag{6.18}$$

We now show that the Schur complement of the full FIM with respect to the initial state $\mathbf{x}_0$ (i.e., the information matrix of $\mathbf{x}_0$, denoted by $\mathbf{A}_0$), has the following relation to the observability matrix $\mathbf{M}$:

Lemma 6.2.1. *The FIM of the initial state $\mathbf{x}_0$, i.e., the corresponding Schur complement of the full FIM, can be factorized as:*

$$\mathbf{A}_0 = \mathbf{M}^T\boldsymbol{\Sigma}\mathbf{M} \tag{6.19}$$

where $\mathbf{M}$ is the observability matrix and $\boldsymbol{\Sigma}$ is a nonsingular (full-rank) real symmetric block-diagonal matrix.

Proof. See Appendix D.1. □

From this lemma as well as the linear algebra theory [49], studying the FIM of the initial state is equivalent to examining the observability matrix. Therefore, the FIM essentially can be seen as the "observability gramian" for the corresponding stochastic system. Note also that the stochastic system (6.1)-(6.2) is observable if and only if the corresponding deterministic system is observable. In the following, we will exploit this result and decompose the observability matrix in a novel way (i.e., based on the measurement sources), which inspires the proposed approaches for improving filter consistency.

6.2.2 Algorithms

Due to the additive property of the measurement information, we decompose the FIM according to the measurements originated from each of the s sources [see (6.15)]:

$$\begin{aligned}\mathbf{A} &= \sum_{\kappa=0}^{k-1} \mathcal{F}_\kappa^T \mathbf{Q}_\kappa^{-1} \mathcal{F}_\kappa + \sum_{i=1}^{s} \sum_{\kappa=0}^{k} \mathcal{H}_{i,\kappa}^T \mathbf{R}_{i,\kappa}^{-1} \mathcal{H}_{i,\kappa} \\ &= \sum_{i=1}^{s} \underbrace{\left(\sum_{\kappa=0}^{k-1} \mathcal{F}_\kappa^T {\mathbf{Q}'_\kappa}^{-1} \mathcal{F}_\kappa + \sum_{\kappa=0}^{k} \mathcal{H}_{i,\kappa}^T \mathbf{R}_{i,\kappa}^{-1} \mathcal{H}_{i,\kappa} \right)}_{\mathbf{A}_i} \end{aligned} \tag{6.20}$$

where $\mathbf{Q}'_\kappa \triangleq s\mathbf{Q}_\kappa$ denotes the inflated state-noise covariance for the each of the s sources, used in order to compensate for the decomposition. Hence, $\mathbf{A}_i$ is the full FIM constructed using measurements *only* from the i-th source. Based on Lemma 6.2.1, the corresponding FIM of the initial state $\mathbf{A}_{0_i}$ can be written as:

$$\mathbf{A}_{0_i} = \mathbf{M}_i^T \mathbf{\Sigma} \mathbf{M}_i \tag{6.21}$$

It is important to note that in (6.21), $\mathbf{M}_i$ is the "observability matrix" which is constructed using the measurements only from the i-th source, *but* padded with zeros for the measurements from the other sources, in order to match the dimension of the full observability matrix $\mathbf{M}$ [e.g., see (6.23)]. This immediately results in $\mathbf{M}_i^T \mathbf{M}_j = \mathbf{0}$ for $i \neq j$. Note also that we directly use $\mathbf{\Sigma}$ in (6.21), since zeros in $\mathbf{M}_i$ will cancel out the corresponding submatrices in $\mathbf{\Sigma}$ to the measurements from the j-th source ($j \neq i$) [see (D.14)]. Therefore, this result (6.21) leads to the following decomposition of the observability matrix:

Lemma 6.2.2. *The observability matrix is decomposed as:*

$$\mathbf{M} = \sum_{i=1}^{s} \mathbf{M}_i \tag{6.22}$$

Proof. Using the fact that $\mathbf{M}_i^T \mathbf{M}_j = \mathbf{0}$ for $i \neq j$, we have [see Lemma 6.2.1, (6.20) and (6.21)]:

$$\mathbf{M}^T \mathbf{\Sigma} \mathbf{M} = \mathbf{A}_0 = \sum_{i=1}^{s} \mathbf{A}_{0_i} = \sum_{i=1}^{s} \mathbf{M}_i^T \mathbf{\Sigma} \mathbf{M}_i \;\Rightarrow\; \mathbf{M} = \sum_{i=1}^{s} \mathbf{M}_i$$

□

We thus see that based on the decomposition of the FIM of the initial state according to the measurement sources, the observability matrix can be *accordingly* decomposed. For instance, if the i-th source provides measurement intermittently at even time steps only, then the i-th decomposition of the observability matrix, $\mathbf{M}_i$, assumes the following form (by assuming k is even):

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{H}_{i,0} \\ \mathbf{0} \\ \vdots \\ \mathbf{H}_{i,k-2}\mathbf{\Phi}_{k-3}\cdots\mathbf{\Phi}_0 \\ \mathbf{0} \\ \mathbf{H}_{i,k}\mathbf{\Phi}_{k-1}\cdots\mathbf{\Phi}_0 \end{bmatrix} \tag{6.23}$$

It is interesting to note that in many cases (e.g., see Section 6.3) the decomposition of the observability matrix (6.23), $\mathbf{M}_i$, is *rank-deficient*, although the observability matrix (6.10), $\mathbf{M}$, is *full-rank*, i.e., the linearized system (6.3)-(6.4) is observable. The right nullspace of the matrix, $\mathbf{M}_i$, and thus the decomposition of the FIM (6.21), $\mathbf{A}_{0_i}$, dictates the directions of the state space along which no information is available from the measurements of the i-th source. If these directions are incorrect, the filter acquires spurious information from the i-th source's measurement, and hence is expected to become inconsistent. Therefore, to ensure consistent estimation, the filter should have $\mathbf{M}_i$, and hence $\mathbf{A}_{0_i}$, of correct nullspace, for $i = 1, \ldots, s$, so that no nonexistent information is gained from the measurements available from each source. To this end, in computing the filter Jacobians at each time step, we explicitly enforce the following constraint on the decompositions of the observability matrix, i.e., each $\mathbf{M}_i$ has correct nullspace denoted by $\mathbf{N}_i$ [see (6.23)]:

$$\mathbf{M}_i\mathbf{N}_i = \mathbf{0} \Leftrightarrow \begin{cases} \mathbf{H}_{i,0}\mathbf{N}_i = \mathbf{0} & , \text{ if } \kappa = 0 \\ \mathbf{H}_{i,\kappa}\mathbf{\Phi}_{\kappa-1}\cdots\mathbf{\Phi}_0\mathbf{N}_i = \mathbf{0} & , \text{ if } \kappa > 0 \end{cases} \tag{6.24}$$

In particular, $\mathbf{N}_i$ is a design choice which defines the desired nullspace for the i-th measurement source, and one practical choice will be the nullspace of the first measurement Jacobian $\mathbf{H}_{i,0}$, i.e., $\mathbf{H}_{i,0}\mathbf{N}_i = \mathbf{0}$. Once $\mathbf{N}_i$ has been selected, the next design decision is to compute the filter Jacobians appropriately, so that (6.24) is satisfied.

We first propose to compute the Jacobians *indirectly*, i.e. to find optimal linearization points that minimize the linearization errors of the points $\mathbf{x}^\star_{k|k}$ and $\mathbf{x}^\star_{k+1|k}$ used in computing the filter

Jacobians, $\mathbf{\Phi}_k$ and $\mathbf{H}_{k+1}$, at time-step $k+1$, subject to the constraint that ensures each $\mathbf{M}_i$ to have the correct nullspace (6.24). Similarly to the OC-EKF2 in Section 2.5.2, this can be formulated as the following constrained minimization problem:

$$\min_{\mathbf{x}^\star_{k|k},\mathbf{x}^\star_{k+1|k}} \int ||\mathbf{x}^\star_k - \mathbf{x}_k||^2 p(\mathbf{x}_k|z_{0:k})d\mathbf{x}_k + \int ||\mathbf{x}^\star_{k+1|k} - \mathbf{x}_{k+1}||^2 p(\mathbf{x}_{k+1}|z_{0:k})d\mathbf{x}_{k+1} \quad (6.25)$$

$$\text{subject to } \mathbf{H}_{i,k}\mathbf{\Phi}_{k-1}\cdots\mathbf{\Phi}_0\mathbf{N}_i = \mathbf{0}\ ,\ \forall i = 1,\ldots,s \quad (6.26)$$

In general it is intractable to solve this problem analytically. However, when $p(\mathbf{x}_k|z_{0:k})$ and $p(\mathbf{x}_{k+1}|z_{0:k})$ are Gaussian which is the assumption employed in the EKF, this problem can be simplified based on the following lemma, and then solved analytically by using Lagrangian multipliers:

Lemma 6.2.3. *When $p(\mathbf{x}_k|z_{0:k})$ and $p(\mathbf{x}_{k+1}|z_{0:k})$ are Gaussian, the constrained minimization problem* (6.25) *and* (6.26) *is equivalent to the following problem:*

$$\min_{\mathbf{x}^\star_{k|k},\mathbf{x}^\star_{k+1|k}} ||\mathbf{x}^\star_{k|k} - \hat{\mathbf{x}}_{k|k}||^2 + ||\mathbf{x}^\star_{k+1|k} - \hat{\mathbf{x}}_{k+1|k}||^2 \quad (6.27)$$

$$\text{subject to } \mathbf{H}_{i,k}\mathbf{\Phi}_{k-1}\cdots\mathbf{\Phi}_0\mathbf{N}_i = \mathbf{0}\ ,\ \forall i = 1,\ldots,s \quad (6.28)$$

Proof. Analogous to the proof of Lemma 2.5.3. □

Alternatively, we can compute the desired filter Jacobians *directly*. Similarly to the OC-EKF3 in Section 2.5.3, we compute the propagation Jacobian $\mathbf{\Phi}_\kappa$ ($\kappa = 0,\ldots,k-1$) in the same way as in the standard EKF, while enforcing the information constraint (6.24) for computing the measurement Jacobian:

$$\min_{\mathbf{H}_{i,k}} ||\mathbf{H}_{i,k} - \mathbf{H}_o||_F^2 \quad (6.29)$$

$$\text{subject to } \mathbf{H}_{i,k}\mathbf{\Phi}_{k-1}\cdots\mathbf{\Phi}_0\mathbf{N}_i = \mathbf{0}\ ,\ \forall i = 1,\ldots,s \quad (6.30)$$

In the above expression, $\mathbf{H}_o$ ideally is the measurement Jacobian computed using the true states, which, however, is not realizable in any practice. Hence, we employ the latest, and thus the best, state estimates for computing this Jacobian as in the standard EKF, i.e., $\mathbf{H}_o = \mathbf{H}_o(\hat{\mathbf{x}}_{k|k-1})$. The optimal solution to the above problem (6.29)-(6.30) is obtained in *closed form* by application of the following lemma:

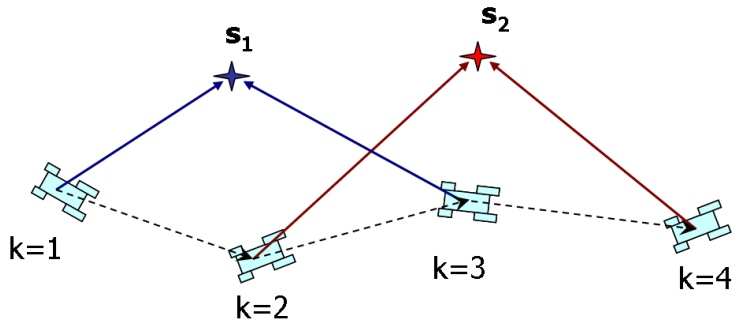


Figure 6.1: Illustration of the application of the two-radar target tracking: A target (robot) moves on a plane and two radars, s_1 and s_2, alternate between measuring distance to the target. For example, at time-step $k = 1$, the first radar, s_1, measures distance to the target; at time-step $k = 2$, the second radar, s_2, measures distance to the target; at time-step $k = 3$, s_1 measures distance again; and so on so forth.

Lemma 6.2.4. *The optimal solution to the constrained minimization problem* (6.29)-(6.30) *is:*

$$\mathbf{H}_{i,k} = \mathbf{H}_o \left(\mathbf{I}_n - \mathbf{U}_i(\mathbf{U}_i^T\mathbf{U}_i)^{-1}\mathbf{U}_i^T\right) \tag{6.31}$$

where $\mathbf{U}_i = \mathbf{\Phi}_{k-1}\cdots\mathbf{\Phi}_0\mathbf{N}_i$.

Proof. Analogous to the proof of Lemma 2.5.4. □

Note that $\mathbf{U}_i$ in (6.31) is the propagated nullspace of the i-th source at time-step k, and $\left(\mathbf{I}_n - \mathbf{U}_i(\mathbf{U}_i^T\mathbf{U}_i)^{-1}\mathbf{U}_i^T\right)$ is the subspace orthogonal to $\mathbf{U}_i$, i.e., the subspace at time-step k where information is available. Hence, as seen from (6.31), $\mathbf{H}_{i,k}$ is the *projection* of the best-available measurement Jacobian onto the information-available subspace.

6.3 Example: Two-radar target tracking

In order to verify the preceding analysis and validate the proposed methodology, in this section, we consider a particular application of two radars tracking a target. Consider a target (robot) that moves on a plane and two radars alternatively provide distance measurements to the target (see Fig. 6.1). Using such intermittent distance measurements as well as odometry measurements, we employ the EKF to estimate the target's pose (position and orientation) in a global frame of reference, denoted by $\mathbf{x}_k = \begin{bmatrix}\mathbf{p}_k^T & \phi_k\end{bmatrix}^T = \begin{bmatrix}x_k & y_k & \phi_k\end{bmatrix}^T$. In what follows, we describe the motion and measurement models of this system in the context of the standard EKF.

In the propagation step, the target (robot) is assumed to be equipped with an odometer that provides measurements processed to obtain an estimate of the pose change between two consecutive time steps, and then employed in the EKF to propagate the state estimate. The EKF propagation equations are identical to (2.2)-(2.3), i.e.,

$$\hat{\mathbf{p}}_{k+1|k} = \hat{\mathbf{p}}_{k|k} + \mathbf{C}(\hat{\phi}_{k|k})^{k}\hat{\mathbf{p}}_{k+1} \tag{6.32}$$

$$\hat{\phi}_{k+1|k} = \hat{\phi}_{k|k} + {}^{k}\hat{\phi}_{k+1} \tag{6.33}$$

where $\mathbf{C}(\cdot)$ denotes the 2×2 rotation matrix, and ${}^{k}\hat{\mathbf{x}}_{k+1} = [{}^{k}\hat{\mathbf{p}}_{k+1}^{T} \;\; {}^{k}\hat{\phi}_{k+1}]^{T}$ is the odometry-based estimate of the target's motion between time-steps k and $k+1$. This estimate is corrupted by zero-mean white Gaussian noise $\mathbf{w}_k = {}^{k}\mathbf{x}_{k+1} - {}^{k}\hat{\mathbf{x}}_{k+1}$, with covariance matrix $\mathbf{Q}_k$. The linearized error-state propagation can be derived in analogy to (2.6), i.e.,

$$\tilde{\mathbf{x}}_{k+1|k} = \mathbf{\Phi}_k\tilde{\mathbf{x}}_{k|k} + \mathbf{G}_k\mathbf{w}_k \tag{6.34}$$

where the state and noise Jacobians are given by

$$\mathbf{\Phi}_k = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{k+1|k} - \hat{\mathbf{p}}_{k|k}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{6.35}$$

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{C}(\hat{\phi}_{k|k}) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{6.36}$$

The distance measurement provided by the i-th radar at time-step $k+1$ is given by:

$$z_{i,k+1} = ||\mathbf{p}_{k+1} - \mathbf{p}_{S_i}|| + v_{i,k+1} = \sqrt{(x_{k+1} - x_{S_i})^2 + (y_{k+1} - y_{S_i})^2} + v_{i,k+1}\ ,\ i = 1, 2 \tag{6.37}$$

where $\mathbf{p}_{S_i} \triangleq [x_{S_i} \; y_{S_i}]^T$ is the known position of the i-th radar expressed in the global frame of reference, and $v_{i,k+1}$ is zero-mean white Gaussian measurement noise, with variance $\sigma_{i,k+1}^2$, i.e., $v_{i,k+1} \sim \mathcal{N}(0, \sigma_{i,k+1}^2)$. Due to the nonlinearity of this measurement function, it is linearized for the use of EKF, which is given by:

$$\tilde{z}_{i,k+1} \simeq \mathbf{H}_{i,k+1}\tilde{\mathbf{x}}_{k+1|k} + v_{i,k+1} \tag{6.38}$$

where the measurement Jacobian is computed as:

$$\mathbf{H}_{i,k+1} = \begin{bmatrix} \frac{(\hat{\mathbf{p}}_{k+1|k} - \mathbf{p}_{S_i})^T}{||\hat{\mathbf{p}}_{k+1|k} - \mathbf{p}_{S_i}||} & 0 \end{bmatrix} = \underbrace{\frac{(\hat{\mathbf{p}}_{k+1|k} - \mathbf{p}_{S_i})^T}{||\hat{\mathbf{p}}_{k+1|k} - \mathbf{p}_{S_i}||}}_{\hat{\boldsymbol{\alpha}}_{i,k+1}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{k+1|k} - \mathbf{p}_{S_i}) \end{bmatrix} \tag{6.39}$$

6.3.1 Observability analysis

We now examine the observability matrix (and thus the information matrix), and show that in the standard EKF the decompositions of the observability matrix with respect to different radars have different nullspace than the ideal case where the true states are used in computing filter Jacobians and which is expected to have correct observability properties.

Ideal EKF linearized system

To facilitate the ensuing analysis, we begin with the *ideal* case of a single radar providing distance measurements. Proceeding analogously to Section 2.4.2, the Jacobians of the ideal EKF are evaluated using the *true* values of the state variables, i.e., $\mathbf{x}_{k|k}^{\star} = \mathbf{x}_{k|k-1}^{\star} = \mathbf{x}_k$, for all k. Note that all matrices evaluated using the true state values are denoted by the symbol " $\breve{}$ ". In this case, by noting that [see (6.35) and (6.39)]

$$\breve{\mathbf{H}}_\kappa \breve{\mathbf{\Phi}}_{\kappa-1} \cdots \breve{\mathbf{\Phi}}_0 = \breve{\boldsymbol{\alpha}}_\kappa \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_0 - \mathbf{p}_S) \end{bmatrix} \tag{6.40}$$

the observability matrix is computed as [see (6.10)]:

$$\breve{\mathbf{M}} = \mathbf{Diag}\left(\breve{\boldsymbol{\alpha}}_0, \breve{\boldsymbol{\alpha}}_1, \cdots, \breve{\boldsymbol{\alpha}}_k\right) \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_0 - \mathbf{p}_S) \\ \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_0 - \mathbf{p}_S) \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_0 - \mathbf{p}_S) \end{bmatrix} \tag{6.41}$$

Using the theorem of the rank of the matrix product [113], we can show that $\mathrm{rank}(\breve{\mathbf{M}}) = 2$ and $\mathrm{null}(\breve{\mathbf{M}}) = \begin{bmatrix} \mathbf{J}(\mathbf{p}_0 - \mathbf{p}_S) \\ 1 \end{bmatrix}$ (see Lemma 2.4.3). This implies that the distance measurements of single radar provide information only about *two* d.o.f., which agrees with our intuition.

Now we extend this analysis to the ideal case where *two* radars alternatively provide distance measurements to the target. Specifically, by proceeding similarly to Section 2.4.2, we can

compute the observability matrix of the ideal-EKF linearized system in this case as follows:

$$\begin{aligned}\breve{\mathbf{M}} &= \underbrace{\mathbf{Diag}\left(\breve{\boldsymbol{\alpha}}_{1,0}, \breve{\boldsymbol{\alpha}}_{2,1}, \cdots, \breve{\boldsymbol{\alpha}}_{1,k-1}, \breve{\boldsymbol{\alpha}}_{2,k}\right)}_{\breve{\mathbf{D}}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_0 - \mathbf{p}_{S_1}) \\ \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_1 - \mathbf{p}_{S_2}) \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_0 - \mathbf{p}_{S_1}) \\ \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_1 - \mathbf{p}_{S_2}) \end{bmatrix} \\ &= \underbrace{\breve{\mathbf{D}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_0 - \mathbf{p}_{S_1}) \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_0 - \mathbf{p}_{S_1}) \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \end{bmatrix}}_{\breve{\mathbf{M}}_1} + \underbrace{\breve{\mathbf{D}} \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\ \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_1 - \mathbf{p}_{S_2}) \\ \vdots & \vdots \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\ \mathbf{I}_2 & \mathbf{J}(\mathbf{p}_1 - \mathbf{p}_{S_2}) \end{bmatrix}}_{\breve{\mathbf{M}}_2}\end{aligned} \tag{6.42}$$

where $\breve{\mathbf{M}}_1$ and $\breve{\mathbf{M}}_2$ are the decompositions of the observability matrix $\breve{\mathbf{M}}$, with respect to the first and second radar, respectively. It is not difficult to see that $\mathrm{rank}(\breve{\mathbf{M}}_1) = \mathrm{rank}(\breve{\mathbf{M}}_2) = 2$, which agrees with the preceding result in the single-radar case, even though the observability matrix is full-rank, i.e., $\mathrm{rank}(\breve{\mathbf{M}}) = 3$, and thus the ideal-EKF linearized system is observable.

Standard EKF linearized system

However, the ideal EKF is not realizable in practice since the true states are generally not available. Therefore, we now consider the standard EKF which computes the Jacobians using the current state estimates, and show that the preceding results do not hold for the standard EKF linearized system. Specifically, the observability matrix for the two-radar scenario under

consideration can be computed as [see (6.10)]:

$$\begin{aligned}\mathbf{M} &= \underbrace{\mathbf{Diag}\left(\hat{\boldsymbol{\alpha}}_{1,0}, \hat{\boldsymbol{\alpha}}_{2,1}, \cdots, \hat{\boldsymbol{\alpha}}_{1,k-1}, \hat{\boldsymbol{\alpha}}_{2,k}\right)}_{\hat{\mathbf{D}}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{0|0} - \mathbf{p}_{S_1}) \\ \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{1|0} - \mathbf{p}_{S_2}) \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{k-1|k-1} + \sum_{\kappa=1}^{k-2} \boldsymbol{\Delta}\hat{\mathbf{p}}_\kappa - \mathbf{p}_{S_1}) \\ \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{k|k} + \sum_{\kappa=1}^{k-1} \boldsymbol{\Delta}\hat{\mathbf{p}}_\kappa - \mathbf{p}_{S_2}) \end{bmatrix} \\ &= \underbrace{\hat{\mathbf{D}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{0|0} - \mathbf{p}_{S_1}) \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{k-1|k-1} + \sum_{\kappa=1}^{k-2} \boldsymbol{\Delta}\hat{\mathbf{p}}_\kappa - \mathbf{p}_{S_1}) \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \end{bmatrix}}_{\mathbf{M}_1} + \underbrace{\hat{\mathbf{D}} \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\ \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{1|0} - \mathbf{p}_{S_2}) \\ \vdots & \vdots \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\ \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{k|k} + \sum_{\kappa=1}^{k-1} \boldsymbol{\Delta}\hat{\mathbf{p}}_\kappa - \mathbf{p}_{S_2}) \end{bmatrix}}_{\mathbf{M}_2} \end{aligned} \tag{6.43}$$

where $\boldsymbol{\Delta}\hat{\mathbf{p}}_\kappa \triangleq \hat{\mathbf{p}}_{\kappa|\kappa} - \hat{\mathbf{p}}_{\kappa|\kappa-1}$ is the correction in the target position due to the EKF update at time-step κ, and in general does not vanish. As a result, the decompositions of the observability matrix becomes full-rank, i.e., $\mathrm{rank}(\mathbf{M}_1) = \mathrm{rank}(\mathbf{M}_2) = 3$, although the observability matrix is still full-rank, $\mathrm{rank}(\mathbf{M}) = 3$. This implies that the standard EKF acquires nonexistent information along one direction of the state space from each radar's measurements, which may lead to inconsistency and thus confirms our preceding analysis. Next, we apply the algorithms presented in Section 6.2.2 to this system so as to improve EKF consistency.

6.3.2 Application of the algorithms

In particular, we choose the desired nullspace of the decompositions of the observability matrix for the two radars as follows (i.e., using the corresponding first state estimates when the two sensors provide their first measurements):

$$\mathbf{N}_1 = \begin{bmatrix} \mathbf{J}(\hat{\mathbf{p}}_{0|0} - \mathbf{p}_{S_1}) \\ 1 \end{bmatrix} \tag{6.44}$$

$$\mathbf{N}_2 = \begin{bmatrix} \mathbf{J}(\hat{\mathbf{p}}_{1|0} - \mathbf{p}_{S_2}) \\ 1 \end{bmatrix} \tag{6.45}$$

We first describe the *indirect* algorithm for computing EKF Jacobians, which finds optimal linearization points by solving the problem (6.27)-(6.28). In this case, the constraint (6.28) can

be simplified as [see (6.43)]:

$$\begin{aligned} & \mathbf{H}_{i,k}\mathbf{\Phi}_{k-1}\cdots\mathbf{\Phi}_0\mathbf{N}_i = \mathbf{0} \\ \Leftrightarrow \quad & \hat{\boldsymbol{\alpha}}_{i,k}\left[\mathbf{I}_2 \quad \mathbf{J}\left(\mathbf{p}^\star_{0|0}+\textstyle\sum_{\kappa=1}^{k}\boldsymbol{\Delta}\mathbf{p}^\star_\kappa-\mathbf{p}_{S_i}\right)\right]\mathbf{N}_i = \mathbf{0} \\ & \Leftrightarrow \quad \mathbf{p}^\star_{k|k} = \mathbf{p}^\star_{k|k-1}-\sum_{\kappa=1}^{k}\boldsymbol{\Delta}\mathbf{p}^\star_\kappa \end{aligned} \tag{6.46}$$

where we have employed the definition similar to $\boldsymbol{\Delta}\hat{\mathbf{p}}_\kappa$, i.e., $\boldsymbol{\Delta}\mathbf{p}^\star_\kappa \triangleq \mathbf{p}^\star_{\kappa|\kappa}-\mathbf{p}^\star_{\kappa|\kappa-1}$. Similarly to the OC-EKF2 (see Section 2.5.2), using the method of Lagrangian multipliers, we can analytically solve for the optimal solution to the problem (6.27)-(6.28):

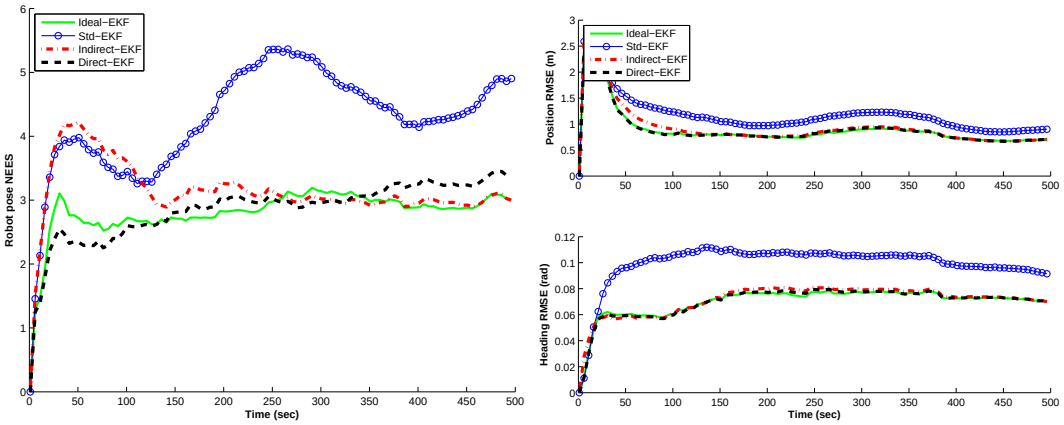
$$\mathbf{p}^\star_{k|k}=\mathbf{p}^\star_{k|k-1}=\hat{\mathbf{p}}_{k|k-1}\ ,\ \phi^\star_{k|k}=\hat{\phi}_{k|k}\ ,\ \phi^\star_{k|k-1}=\hat{\phi}_{k|k-1} \tag{6.47}$$

Alternatively, we can use the *direct* algorithm for computing EKF Jacobians. Specifically, we directly apply the optimal solution of the measurement Jacobian (6.31), i.e., projecting the best-available measurement Jacobian onto the information-available directions (see Lemma 6.2.4), while computing the propagation Jacobians in the same way as the standard EKF [see (6.35)].

It is important to point out that in the both proposed EKF algorithms, once the filter Jacobians are computed, the state estimates and covariance are propagated and updated in the same way as in the standard EKF. Note also that the proposed EKFs are causal and realizable in practice, since they do not use any information about the future or true states.

6.3.3 Numerical results

To demonstrate the capability of the proposed algorithms to improve filter consistency, we conducted 100 Monte-Carlo simulations under various conditions, and as before, employed the RMSE and NEES [14] as the metrics to evaluate the filters' performance. In this numerical simulation test, a target (robot) with a simple differential drive model moved on a planar surface, at a constant velocity of $v = 0.25$ m/sec. The two-drive wheels were equipped with encoders, which measure their revolutions and provide measurements of velocity (i.e., right and left wheel velocities, v_r and v_l, respectively), with standard deviation equal to $\sigma = 1\%v$ for each wheel. These measurements were used to obtain the linear and rotational velocity measurements for the target, which are given by $v = \frac{v_r+v_l}{2}$ and $\omega = \frac{v_r-v_l}{a}$, where $a = 0.5$ m is the



(a) Avg. NEES of target pose

(b) Avg. RMSE of target pose

Figure 6.2: Monte-Carlo results of two-radar tracking. In this simulation, a target moves on a circular trajectory and two radars with known positions alternate between providing distance measurements to the target. Note that in these plots, the RMSE values of the ideal EKF and the two proposed EKFs are very close, which makes the corresponding lines difficult to distinguish.

distance between the active wheels. The standard deviations of the linear and rotational velocity measurement noise were thus equal to $\sigma_v = \frac{\sigma}{\sqrt{2}}$ and $\sigma_\omega = \frac{\sqrt{2}\sigma}{a}$, respectively. Two radars with known positions alternatively provide distance measurements to the target. The standard deviation of the distance-measurement noise was equal to 10% of the radar-to-target distance. It should be noted that the sensor-noise levels selected for the simulations are larger than what is typically encountered in practice. This was done on purpose in order to make the effects of inconsistency more apparent, since larger noise leads to larger estimation errors, and thus less accurate linearization.

Fig. 6.2 shows the Monte-Carlo results of the average NEES and RMSE for the robot (target) pose. It becomes clear that the proposed EKFs (i.e., the indirect and direct EKFs) perform much better than the standard EKF, and very close to the benchmark, the ideal EKF, in terms of both consistency (NEES) and accuracy (RMSE). This is attributed to the fact that the proposed EKFs acquire the information only along the correct directions of the state space from *each* radar's measurements.

6.4 Summary

In this chapter, we have studied the filter inconsistency issue in discrete-time nonlinear systems where only partial-state measurements are available. We showed that despite the system observability, the linearized filters such as the EKF still can become inconsistent. To understand the causes of the inconsistency, we examined the FIM of the initial state (by marginalizing all but the initial state) and showed that it is closely related to the observability matrix. Moreover, we proposed a novel decomposition of the observability matrix of the linearized system with respect to different sources of measurements, and proved that when using the standard EKF, each decomposition of the observability matrix has higher rank than that of the ideal case. This implies that the filter gain spurious information from the measurements of each source, which leads to inconsistency. To address this issue, we proposed to compute the filter Jacobians in such a way that ensures that each decomposition of the observability matrix has nullspace of correct dimension. We applied the proposed algorithms to the problem of two-radar target tracking, and demonstrated the superior performance of the proposed filters over the standard approach.

Chapter 7

A Bank of MAP Estimators for Target Tracking

In this chapter, we study the consistency of nonlinear estimators from a perspective different than the system observability used in the previous chapters, i.e., finding and tracking multiple modes of the posterior pdf, and present a general framework to improve consistency for estimation problems with polynomial (nonlinear) measurement functions. The key idea of our approach is to analytically select and track state hypotheses (the modes of the posterior pdf). We apply our proposed approach to the particular problem of target tracking, i.e., estimating the kinematic state of a moving target using only range or bearing measurements from a single mobile sensor (robot) whose position and orientation are known. Parts of this chapter have been published in [76, 77].

7.1 Introduction

Nonlinear estimation problems such as target tracking are often addressed using linearized estimators (e.g., the EKF [14, 17]). These estimators suffer from linearization errors and the inability to track multimodal pdfs, which often arise in nonlinear estimation problems. Several methods have been proposed to reduce linearization errors. For example, the IEKF [14] iterates the filter update till convergence, by relinearizing the measurement function at each iteration; The UKF [84] deterministically samples the nonlinear function around the state estimate, thus improving the linear approximation. However, any (explicit or implicit) linearization-based

filtering approach marginalizes all but the current state, and hence is unable to refine past linearization points. In contrast, a batch-MAP estimator [91] computes the estimates for the states at all time steps using all available measurements. This allows continuous relinearization of the entire state trajectory, which greatly reduces linearization errors. However, just as the EKF and its variants, the batch-MAP estimator can only track one, of the potentially many, modes of the posterior pdf. Only a few estimators, such as the multi-hypothesis EKF (MHEKF) [99] and the PF [8, 41, 58], are specifically designed to treat multimodal distributions by simultaneously tracking a set of different state estimates. However, in most cases these hypotheses are generated randomly, thus wasting a considerable portion of the computational resources.

In this chapter, we present a *parametric linearized* estimation framework that provides both relinearization and multi-hypothesis tracking, together with a highly efficient hypothesis generation scheme. Ideally, the optimal approach to the batch-MAP estimation problem would be to compute all modes of the posterior pdf, thus ensuring a globally optimal estimate. However, as our analysis will show later, this approach is computationally intractable due to the growing size of the state vector. We therefore relax the problem, and optimize only for the current state at each time step, treating the state history of each hypothesis as a constant prior. We first convert the nonlinear cost function of this subproblem into polynomial or rational form, and subsequently employ algebraic geometry techniques [33] to analytically compute all local minima and thus all modes of the pdf. Each mode is used to initialize a new MAP estimator in the bank, thus allowing to track the most probable hypotheses of the state trajectory, and in turn greatly improving the accuracy and consistency of the MAP estimate. At the same time, we achieve low, resource-adaptive computational cost through pruning and marginalization. The former controls the exponential growth of hypotheses, while the latter limits the size of the state vector. We successfully apply the proposed bank of MAP approach to both range-only and bearing-only target tracking. We stress that apart from the particular application of target tracking treated in this work, the proposed framework is applicable to a broad class of nonlinear estimation problems in robotics and computer vision that can be expressed in (or converted into) polynomial form.

7.2 Related work

The problem of target tracking has been studied for decades and many different estimators of both batch and recursive types have been proposed in the literature [14, 17]. Among these algorithms, the EKF is one of the most widely used methods. However, due to the fact that the EKF is unable to refine the past linearization points when new measurements become available and thus can result in large linearization errors, it gives unsatisfactory performance. This has given rise to refinements of the EKF developed specifically for bearing-only tracking, e.g., the modified polar coordinates EKF [1] and the shifted Rayleigh filter [31]. However, these EKF variants can only track a single mode (more precisely, mean) of the posterior pdf of the target state and thus suffer from the same problem as the EKF, i.e., they can potentially track an inaccurate mode of the pdf and hence become inconsistent or even diverge.

To mitigate the aforementioned issue, a MHEKF was proposed specifically for bearing-only tracking in [99] to track multiple hypotheses of the target state. The MHEKF makes an assumption about the minimum and maximum distances between the sensor and target and partitions this range interval to a number of subintervals, each representing a hypothesis regarding the true range of the target. A bank of independently operating range-parameterized EKFs are thus created, each designed for one of the hypotheses and receiving the same bearing measurement. The MHEKF determines a fixed number of EKFs at the first available measurement, and this idea was extended in [122] so that the filter bank can dynamically change its size at each time step based on the current measurement likelihood. Since no filter in the MHEKF can guarantee computing the globally optimal estimate (due to the multimodal nature of the distribution as well as its inability to relinearize the nonlinear measurement function), this approach can also become inconsistent and diverge. Note that this method assumes prior knowledge about the range interval, while this might not always be available in real applications of bearing-only tracking (e.g., using cameras). More importantly, this approach does not provide a measurable criterion about how many partitions are needed in the assumed range interval and where to choose them. In contrast, our proposed bank of MAP estimators selects most probable hypotheses of the target trajectory based on local optimality at each time step.

Considerable attention has recently been paid to the PF for both bearing-only and range-only target tracking [8, 28, 50, 57, 58, 134], because of its capability of solving nonlinear estimation

problems with multimodal pdfs. In the standard (bootstrap) PF, each particle represents a hypothesis of the target state, weighted by its measurement likelihood. If the particles sample the state space sufficiently, the PF will converge to the true distribution. However, the particles are usually initialized randomly, and if far from a mode of the pdf, their weights can decay quickly and lead to particle depletion and thus inconsistency, even if a resampling scheme is employed. This is due to the fact that the very few surviving particles may not be sufficient to represent the underlying multimodal pdf. Therefore, in order to converge to good estimates, the PF requires to use a large number of particles, thus exacerbating its computational demands. In contrast, the proposed estimator analytically computes all modes of the posterior pdf at the current time step and efficiently focuses the available computational resources on the most probable hypotheses of the state.

7.3 Problem formulation

Consider a single sensor (robot) moving in a plane and estimating the state (position, velocity, etc.) of a moving target, by processing the available range or bearing measurements. In this work, we study the case of *global* tracking, i.e., the position of the target is expressed with respect to a fixed (global) frame of reference, instead of a relative *sensor-centered* one. We hereafter assume that the pose (position and orientation) of the tracking sensor is known with high accuracy in the global frame of reference (e.g., from GPS and compass measurements). The state vector of the target at time-step k is defined as a vector of dimension $2N$, where $N-1$ is the highest order of time derivative of the target position described by a known stochastic target motion model, and can include components such as position, velocity, acceleration, etc.:

$$\mathbf{x}_k = \begin{bmatrix} x_{T_k} & y_{T_k} & \dot{x}_{T_k} & \dot{y}_{T_k} & \ddot{x}_{T_k} & \ddot{y}_{T_k} & \cdots \end{bmatrix}^T =: \begin{bmatrix} \mathbf{p}_{T_k}^T & \mathbf{d}_{T_k}^T \end{bmatrix}^T \tag{7.1}$$

where $\mathbf{p}_{T_k} \triangleq \begin{bmatrix} x_{T_k} & y_{T_k} \end{bmatrix}^T$ is the target position, and $\mathbf{d}_{T_k} \triangleq \begin{bmatrix} \dot{x}_{T_k} & \dot{y}_{T_k} & \ddot{x}_{T_k} & \ddot{y}_{T_k} & \cdots \end{bmatrix}^T$ denotes all the higher-order time derivatives of the target position.

In the following, we present the target stochastic motion model and the sensor measurement model that will be used throughout this chapter. Subsequently, we describe the batch-MAP formulation of target tracking, which is similar to that of SLAM presented in Section 5.2.3.

7.3.1 Motion model

We consider the case where the target moves randomly but assume that the stochastic model describing the motion of the target (e.g., constant acceleration or constant velocity [14]) is known. In particular, the discrete-time state propagation equation is generically given by the following linear form:

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{w}_{k-1} \tag{7.2}$$

where $\mathbf{w}_{k-1}$ is zero-mean white Gaussian noise with covariance $\mathbf{Q}_{k-1}$. The state transition matrix, $\mathbf{\Phi}_{k-1}$, and the process noise Jacobian, $\mathbf{G}_{k-1}$, that appear in the preceding expression depend on the motion model used [14]. We will make no further assumptions on these matrices other than that their values are known.

7.3.2 Measurement model

In this work, we are interested in the case in which a single sensor measures its distance or bearing angle to the target. The corresponding measurement equations are described below.

Range-only measurement The range-only measurement at time-step k is given by:

$$z_k = \sqrt{(x_{T_k} - x_{S_k})^2 + (y_{T_k} - y_{S_k})^2} + n_{\rho_k} \tag{7.3}$$

$$\triangleq h_\rho(\mathbf{x}_k) + n_{\rho_k} \tag{7.4}$$

where $\mathbf{x}_{S_k} \triangleq [\mathbf{p}_{S_k}^T \ \phi_{S_k}]^T \triangleq [x_{S_k} \ y_{S_k} \ \phi_{S_k}]^T$ is the known sensor pose expressed in the global frame of reference, and n_{ρ_k} is zero-mean white Gaussian measurement noise, with variance $\sigma_{\rho_k}^2$, i.e., $n_{\rho_k} \sim \mathcal{N}(0, \sigma_{\rho_k}^2)$.

Bearing-only measurement Similarly, the bearing measurement at time-step k is given by:

$$z_k = \operatorname{atan2}\left((y_{T_k} - y_{S_k}), (x_{T_k} - x_{S_k})\right) - \phi_{S_k} + n_{\theta_k} \tag{7.5}$$

$$\triangleq h_\theta(\mathbf{x}_k) + n_{\theta_k} \tag{7.6}$$

where n_{θ_k} is zero-mean white Gaussian measurement noise, with variance $\sigma_{\theta_k}^2$, i.e., $n_{\theta_k} \sim \mathcal{N}(0, \sigma_{\theta_k}^2)$.

7.3.3 Batch-MAP optimization

Similar to the SLAM batch-MAP formulation (see Section 5.2.3), the batch-MAP estimator utilizes all available information to estimate the *entire* target trajectory that is given by stacking all states in the time interval $[0, k]$ [see (7.1)]:

$$\mathbf{x}_{0:k} = \begin{bmatrix} \mathbf{x}_0^T & \mathbf{x}_1^T & \cdots & \mathbf{x}_k^T \end{bmatrix}^T \tag{7.7}$$

Specifically, the batch-MAP estimator seeks to determine the entire state-space trajectory estimate $\hat{\mathbf{x}}_{0:k|k}$ by maximizing the posterior pdf which is equivalent to minimizing the following cost function [see (5.14)]:

$$c(\mathbf{x}_{0:k}) = \frac{1}{2}||\mathbf{x}_0 - \hat{\mathbf{x}}_{0|0}||^2_{\mathbf{P}_{0|0}} + \sum_{\kappa=1}^{k} \frac{1}{2}||\mathbf{x}_\kappa - \mathbf{\Phi}_{\kappa-1}\mathbf{x}_{\kappa-1}||^2_{\mathbf{Q}'_{\kappa-1}} + \sum_{\kappa=1}^{k} \frac{1}{2}||z_\kappa - h(\mathbf{x}_\kappa)||^2_{\sigma^2_\kappa} \tag{7.8}$$

where $\mathbf{Q}'_{\kappa-1} \triangleq \mathbf{G}_{\kappa-1}\mathbf{Q}_{\kappa-1}\mathbf{G}^T_{\kappa-1}$.

A standard approach for minimizing (7.8) is to employ Newton-Raphson iterative minimization [163], which relies on the Jacobian and Hessian matrices [see (5.15)]. Hence we first examine their structure that will be useful for the ensuing analysis. Specifically, at the ℓ-th iteration, the Jacobian $\mathbf{b}^{(\ell)}$ can be obtained as:

$$\begin{aligned} \mathbf{b}^{(\ell)} = \; & \mathbf{\Pi}^T \mathbf{P}^{-1}_{0|0} \left(\hat{\mathbf{x}}^{(\ell)}_{0|k} - \hat{\mathbf{x}}_{0|0} \right) + \\ & \sum_{\kappa=1}^{k} \mathcal{F}^{(\ell)T}_{\kappa-1} \mathbf{Q}'^{-1}_{\kappa-1} \left(\hat{\mathbf{x}}^{(\ell)}_{\kappa|k} - \mathbf{\Phi}_{\kappa-1} \hat{\mathbf{x}}^{(\ell)}_{\kappa-1|k} \right) + \sum_{\kappa=1}^{k} \sigma_\kappa^{-2} \mathcal{H}^{(\ell)T}_\kappa \left(z_\kappa - h(\hat{\mathbf{x}}^{(\ell)}_{\kappa|k}) \right) \end{aligned} \tag{7.9}$$

where $\mathbf{\Pi} \triangleq \begin{bmatrix} \mathbf{I}_{2N} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$ is used to adjust the dimension of the $2N$-dimensional prior estimate to the dimension of the entire state $\mathbf{x}_{0:k}$. In the above expression, $\mathcal{F}^{(\ell)}_{\kappa-1}$ and $\mathcal{H}^{(\ell)}_\kappa$, are the Jacobians of the motion and measurement models [see (7.2), and (7.4) or (7.6), respectively], with respect to the entire state $\mathbf{x}_{0:k}$, evaluated at $\hat{\mathbf{x}}^{(\ell)}_{0:k|k}$. It is important to note that both the target motion model and the measurement function involve only a few states, i.e., the target motion only depends on two consecutive states, while the measurement only depends on the target position where it is observed. Thus, $\mathcal{F}_{\kappa-1}$ and $\mathcal{H}_\kappa$ have the following sparse structure (for concise notations, the iteration index (ℓ) is dropped here):

$$\mathcal{F}_{\kappa-1} = \begin{bmatrix} \mathbf{0}_{2N\times 2N} & \cdots & -\mathbf{\Phi}_{\kappa-1} & \mathbf{I}_{2N} & \cdots & \mathbf{0}_{2N\times 2N} \end{bmatrix} \tag{7.10}$$

$$\mathcal{H}_\kappa = \begin{bmatrix} \mathbf{0}_{1\times 2N} & \cdots & -\mathbf{H}_\kappa & \cdots & \mathbf{0}_{1\times 2N} \end{bmatrix} \tag{7.11}$$

where $\mathbf{H}_\kappa$ is the measurement Jacobian matrix at time-step κ, given by [see (7.4) and (7.6)]:

$$\mathbf{H}_\kappa = \begin{cases} \left[\frac{(\hat{\mathbf{p}}_{T_\kappa|k} - \mathbf{p}_{S_\kappa})^T}{||\hat{\mathbf{p}}_{T_\kappa|k} - \mathbf{p}_{S_\kappa}||} \;\; \mathbf{0}_{1\times(2N-2)} \right], \text{if } z_\kappa \text{ is range}-\text{only} \\ \\ \left[\frac{(\hat{\mathbf{p}}_{T_\kappa|k} - \mathbf{p}_{S_\kappa})^T \mathbf{J}^T}{||\hat{\mathbf{p}}_{T_\kappa|k} - \mathbf{p}_{S_\kappa}||^2} \;\; \mathbf{0}_{1\times(2N-2)} \right], \text{if } z_\kappa \text{ is bearing}-\text{only} \end{cases} \tag{7.12}$$

On the other hand, the Hessian matrix, $\mathbf{A}$, is approximated in the Gauss-Newton method by:

$$\mathbf{A} \simeq \boldsymbol{\Pi}^T \mathbf{P}_{0|0}^{-1} \boldsymbol{\Pi} + \sum_{\kappa=1}^{k} \mathcal{F}_{\kappa-1} \mathbf{Q}'^{-1}_{\kappa-1} \mathcal{F}_{\kappa-1} + \sum_{\kappa=1}^{k} \sigma_\kappa^{-2} \mathcal{H}_\kappa^T \mathcal{H}_\kappa \tag{7.13}$$

which is a good approximation for small-residual problems [163].

Note that the Hessian $\mathbf{A}$ has dimension $2N(k+1) \times 2N(k+1)$ [see (7.1) and (7.7)]. However, due to the sparse structure of the matrices $\mathcal{H}_\kappa$ and $\mathcal{F}_{\kappa-1}$, the matrix $\mathbf{A}$ is also sparse, and more importantly, it has a banded structure with upper and lower bandwidth of $4N$ (due to the Markov motion model and the range or bearing measurement only depending on the target position where it is observed). We can exploit this sparse banded structure to reduce the computational complexity of solving (5.20) to $\mathcal{O}(N^3 k)$, instead of $\mathcal{O}\left(N^3 k^3\right)$ [49].

7.4 Incrementally solving the batch-MAP optimization problem

We know that iterative algorithms such as Gauss-Newton are only able to converge to one local minimum, while the nonlinear batch-MAP problem of minimizing (7.8) potentially has multiple local minima. In order to guarantee global optimality, ideally we would like to analytically compute all the stationary points of the batch-MAP problem. Unfortunately, in general, it is computationally intractable to do so. In this section, we present an incremental (approximate) solution to the batch-MAP problem by intelligently generating multiple high-quality estimates used as initial guesses for an iterative algorithm. Specifically, we relax the problem by fixing the past state estimates and analytically solving a one-step minimization problem for the current state estimate, at every time step when a new measurement becomes available. This analytic optimization is carried out by converting the nonlinear cost function into polynomial or rational form which is then solved using algebraic geometry techniques. We then use the analytically-computed local minima corresponding to the current state along with the fixed past

state estimates as highly accurate initial guesses, which are refined through the iterative algorithm used to solve the batch-MAP problem.

7.4.1 Relaxation of the batch-MAP problem

As will become clear, by transforming the nonlinear measurement function into polynomial or rational form, we can convert the KKT optimality conditions of the batch-MAP problem into a polynomial system (see Section 7.4.2). The number of variables increases linearly with respect to the time horizon k. However, since the complexity of solving multivariate polynomial systems is exponential in the number of variables [45], it is, in general, computationally intractable to solve the batch-MAP problem analytically.

For this reason, we relax the batch-MAP problem and solve it incrementally. In particular, at time-step k, by fixing the past state estimates $\hat{\mathbf{x}}_{0:k-1|k-1}$ (i.e., assuming they are optimal), we approximate the cost function (7.8) as follows:

$$c(\mathbf{x}_{0:k}) \simeq c(\hat{\mathbf{x}}_{0:k-1|k-1}) + \frac{1}{2}||\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}||^2_{\mathbf{P}_{k|k-1}} + \frac{1}{2}||z_k - h(\mathbf{x}_k)||^2_{\sigma_k^2} \tag{7.14}$$

where $\mathcal{N}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$ is the prior pdf for the current new state $\mathbf{x}_k$, and is computed based on the linear motion model (7.2) as follows:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \tag{7.15}$$

$$\mathbf{P}_{k|k-1} = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{\Phi}_{k-1}^T + \mathbf{Q}'_{k-1} \tag{7.16}$$

Now the relaxed batch-MAP problem of minimizing (7.14) becomes equivalent to solving the following one-step minimization problem incrementally for the new state estimate:

$$\min_{\mathbf{x}_k} \left[\frac{1}{2}||\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}||^2_{\mathbf{P}_{k|k-1}} + \frac{1}{2}||z_k - h(\mathbf{x}_k)||^2_{\sigma_k^2}\right] \tag{7.17}$$

Once we find all the local minima of (7.17), we use them along with the past state estimates as accurate initial guesses in the proposed bank of MAP estimators (see Section 7.5), while in the proposed AGS-PF, they are employed to guide sampling particles (see Section 8.2). Therefore, in what follows we describe the analytic approach to determining all the local minima in detail.

7.4.2 Analytic determination of local minima

Observing that both relative range and bearing measurements depend only on the target position [see (7.4) and (7.6)], we can decouple the target position $\mathbf{p}_{T_k}$ and the remaining states $\mathbf{d}_{T_k}$

in solving (7.17), so as to simplify the ensuing derivations. Specifically, using the following partitioning of the information matrix, $\mathbf{P}_{k|k-1}^{-1} \triangleq \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{pp}_{k|k-1}} & \mathbf{\Sigma}_{\mathbf{pd}_{k|k-1}} \\ \mathbf{\Sigma}_{\mathbf{dp}_{k|k-1}} & \mathbf{\Sigma}_{\mathbf{dd}_{k|k-1}} \end{bmatrix}$, the cost function of (7.17) can be expanded as:

$$c(\mathbf{x}_k) = \frac{1}{2}(\mathbf{p}_{T_k} - \hat{\mathbf{p}}_{T_{k|k-1}})^T \mathbf{\Sigma}_{\mathbf{pp}_{k|k-1}} (\mathbf{p}_{T_k} - \hat{\mathbf{p}}_{T_{k|k-1}}) + \frac{1}{2}(\mathbf{d}_{T_k} - \hat{\mathbf{d}}_{T_{k|k-1}})^T \mathbf{\Sigma}_{\mathbf{dd}_{k|k-1}} (\mathbf{d}_{T_k} - \hat{\mathbf{d}}_{T_{k|k-1}}) + (\mathbf{p}_{T_k} - \hat{\mathbf{p}}_{T_{k|k-1}})^T \mathbf{\Sigma}_{\mathbf{pd}_{k|k-1}} (\mathbf{d}_{T_k} - \hat{\mathbf{d}}_{T_{k|k-1}}) + \frac{1}{2\sigma_k^2} \left(z_k - h(\mathbf{p}_{T_k})\right)^2 \tag{7.18}$$

We note that

$$\min_{\mathbf{p}_{T_k}, \mathbf{d}_{T_k}} c(\mathbf{p}_{T_k}, \mathbf{d}_{T_k}) \;=\; \min_{\mathbf{p}_{T_k}} \left(\min_{\mathbf{d}_{T_k}} c(\mathbf{p}_{T_k}, \mathbf{d}_{T_k}) \right)$$

Thus, we first solve for $\mathbf{d}_{T_k}$ based on its optimality condition, i.e., by setting the gradient of (7.18) with respect to $\mathbf{d}_{T_k}$ to zero, and obtain:

$$\mathbf{d}_{T_k} = \hat{\mathbf{d}}_{T_{k|k-1}} - \mathbf{\Sigma}_{\mathbf{dd}_{k|k-1}}^{-1} \mathbf{\Sigma}_{\mathbf{dp}_{k|k-1}} (\mathbf{p}_{T_k} - \hat{\mathbf{p}}_{T_{k|k-1}}) \tag{7.19}$$

Substitution of (7.19) into (7.18) yields:

$$c(\mathbf{p}_{T_k}) = \frac{1}{2}(\mathbf{p}_{T_k} - \hat{\mathbf{p}}_{T_{k|k-1}})^T \mathbf{P}_{\mathbf{pp}_{k|k-1}}^{-1} (\mathbf{p}_{T_k} - \hat{\mathbf{p}}_{T_{k|k-1}}) + \frac{1}{2\sigma_k^2} \left(z_k - h(\mathbf{p}_{T_k})\right)^2 \tag{7.20}$$

where $\mathbf{P}_{\mathbf{pp}_{k|k-1}}$ is the covariance matrix corresponding to the target position, obtained by partitioning the covariance matrix as $\mathbf{P}_{k|k-1} \triangleq \begin{bmatrix} \mathbf{P}_{\mathbf{pp}_{k|k-1}} & \mathbf{P}_{\mathbf{pd}_{k|k-1}} \\ \mathbf{P}_{\mathbf{dp}_{k|k-1}} & \mathbf{P}_{\mathbf{dd}_{k|k-1}} \end{bmatrix}$. In the above expression (7.20), we have employed the following identity:

$$\mathbf{P}_{\mathbf{pp}_{k|k-1}}^{-1} = \mathbf{\Sigma}_{\mathbf{pp}_{k|k-1}} - \mathbf{\Sigma}_{\mathbf{pd}_{k|k-1}} \mathbf{\Sigma}_{\mathbf{dd}_{k|k-1}}^{-1} \mathbf{\Sigma}_{\mathbf{dp}_{k|k-1}} \tag{7.21}$$

which follows from the block matrix inversion lemma [49].

We thus see that solving (7.17) becomes equivalent to minimizing (7.20). It is important to note that the size of the nonlinear problem has dramatically decreased from $2N$ for (7.17) to a constant size of 2 for minimizing (7.20). Moreover, the analytic solution for the target position is independent of its higher-order time derivatives, regardless of the stochastic target motion model. In the following we present our algebraic geometry approaches for minimizing (7.20) analytically, in the cases of range-only tracking and bearing-only tracking, respectively.

Range-only tracking

In the case of range-only target tracking, i.e., $h(\cdot) = h_\rho(\cdot)$ [see (7.4)], by introducing a new variable $\rho = h_\rho(\mathbf{p}_{T_k})$, the problem of minimizing (7.20) is equivalent to the following constrained minimization problem:

$$\min_{\mathbf{p}_{T_k},\, \rho} \left[\frac{1}{2}(\mathbf{p}_{T_k} - \hat{\mathbf{p}}_{T_{k|k-1}})^T \mathbf{P}^{-1}_{\mathbf{pp}_{k|k-1}} (\mathbf{p}_{T_k} - \hat{\mathbf{p}}_{T_{k|k-1}}) + \frac{1}{2\sigma^2_{\rho_k}} (z_k - \rho)^2 \right] \tag{7.22}$$

$$\text{subject to } \rho^2 = (x_{S_k} - x_{T_k})^2 + (y_{S_k} - y_{T_k})^2 \,,\ \rho \geq 0 \tag{7.23}$$

which can be solved by employing the method of Lagrange multipliers [16]. Specifically, without loss of generality, by assuming $\mathbf{P}^{-1}_{\mathbf{pp}_{k|k-1}} = \mathbf{Diag}(s_1, s_2)$, the Lagrangian function can be constructed as follows:[1]

$$\begin{aligned} \mathcal{L}(x_{T_k}, y_{T_k}, \rho, \lambda) = {} & \frac{s_1}{2}(x_{T_k} - \hat{x}_{T_{k|k-1}})^2 + \frac{s_2}{2}(y_{T_k} - \hat{y}_{T_{k|k-1}})^2 \\ & + \frac{(z_k - \rho)^2}{2\sigma^2_{\rho_k}} + \lambda \left(\rho^2 - (x_{S_k} - x_{T_k})^2 - (y_{S_k} - y_{T_k})^2 \right) \end{aligned} \tag{7.24}$$

where λ is the Lagrangian multiplier. Setting the derivatives of $\mathcal{L}(\cdot)$ with respect to the four optimization variables to zero, and performing simple algebraic manipulations, we have:

$$\frac{\partial \mathcal{L}}{\partial x_{T_k}} = 0 \Rightarrow x_{T_k} = \frac{s_1 \hat{x}_{T_{k|k-1}} - 2\lambda x_{S_k}}{s_1 - 2\lambda} \tag{7.25}$$

$$\frac{\partial \mathcal{L}}{\partial y_{T_k}} = 0 \Rightarrow y_{T_k} = \frac{s_2 \hat{y}_{T_{k|k-1}} - 2\lambda y_{S_k}}{s_2 - 2\lambda} \tag{7.26}$$

$$\frac{\partial \mathcal{L}}{\partial \rho} = 0 \Rightarrow \rho = \frac{z_k}{1 + 2\sigma^2_{\rho_k}\lambda} \tag{7.27}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \Rightarrow 0 = \rho^2 - (x_{S_k} - x_{T_k})^2 - (y_{S_k} - y_{T_k})^2 \tag{7.28}$$

[1] We can always diagonalize $\mathbf{P}^{-1}_{\mathbf{pp}}$ by applying a 2D rotational transformation, which does not affect distance measurements. Moreover, we here temporarily omit the positivity constraint on ρ, which will be used later for determining feasible solutions.

Therefore, by substituting (7.25)-(7.27) into (7.28) and multiplying both sides of (7.28) with $(1+2\sigma_{\rho_k}^2\lambda)^2(s_1-2\lambda)^2(s_2-2\lambda)^2$, we obtain a fourth-order univariate polynomial in λ:[2]

$$0 = f(\lambda) = \sum_{i=0}^{4} a_i \lambda^i \tag{7.29}$$

where a_i, $i = 0, 1, \ldots, 4$, are the coefficients expressed in terms of the known quantities s_1, s_2, z_k, σ_{ρ_k}, $\hat{x}_{T_{k|k-1}}$, $\hat{y}_{T_{k|k-1}}$, x_{S_k}, and y_{S_k}. Since $f(\lambda)$ is quartic, its roots can be found in closed form [76]. Although there exist 4 solutions for λ and thus 4 solutions for x_{T_k}, y_{T_k} and ρ, as they depend injectively on λ [see (7.25)-(7.27)], we only need to consider the pairs (x_{T_k}, y_{T_k}) that correspond to real solutions for λ and to a nonnegative ρ [see (7.23)]. Moreover, since some of these solutions could be local maxima and/or saddle points, the second-order derivative test [16] is employed to extract the minima. Finally, once we determine all the local minima for the target position, we compute the corresponding estimates for the higher-order position derivatives via (7.19).

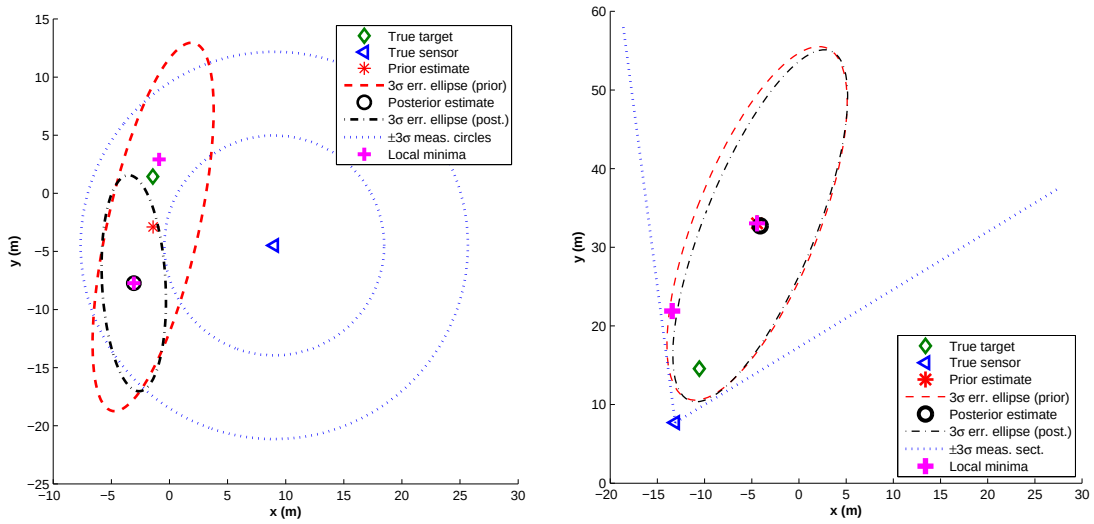
Since the maximum number of local minima for the problem (7.17) will significantly impact the computational complexity of our proposed algorithms, we seek a tighter upper bound for it. In particular, based on the finite dimensional Mountain Pass Theorem (MPT) (see Theorem 5.2 in [81]), we can show the following lemma:

Lemma 7.4.1. *There are at most 2 local minima for the problem of minimizing* (7.22).

Proof. See Appendix E.1. □

Thus, we see from this lemma that the total number of local minima for the one-step MAP problem (7.17) for range-only target tracking, in worst case, can grow exponentially over time, in an order of 2^k, instead of 4^k. Fig. 7.1(a) shows a typical example where two local minima for the current state occur while the MAP estimate erroneously converges to a local minimum with larger error.

[2] It is important to note that if any of the denominators of (7.25)-(7.27) becomes zero while the corresponding numerator is nonzero, the target is at the infinity position, and moreover the cost of (7.23) also becomes infinity and hence attains the global maximum, which is not interesting to us. On the other hand, there exists the degenerate case where both the numerator and denominator of (7.25) or (7.26) become zeros (i.e., $\frac{0}{0}$), which can be avoided through an appropriate coordinate transformation.



(a) Range-only tracking (b) Bearing-only tracking

Figure 7.1: Illustrative problem for single time step target tracking: The crosses indicate the locations of the analytically-computed local minima. It is clear that the MAP estimate initialized with the prior estimate converges to the local minimum with larger error with respect to ground truth. Note that in bearing-only tracking, the MAP estimate is computed based on the original (not inferred) measurements, and the approximation (7.32) used in the inferred measurements introduces a slight offset in the analytic local minima.

Bearing-only tracking

We now consider the problem of minimizing (7.20) in the case of bearing-only tracking, i.e., $h(\cdot) = h_\theta(\cdot)$ [see (7.6)]. In order to use an algebraic geometry approach, we create an inferred measurement that has *rational* form. Specifically, after moving the sensor orientation term to the left hand side of (7.5), and applying the tangent function on both sides, we obtain the following transformed measurement:

$$\check{z}_k \triangleq \tan(z_k + \phi_{S_k}) = \tan\left(\operatorname{atan2}\left((y_{T_k} - y_{S_k}), (x_{T_k} - x_{S_k})\right) + n_{\theta_k}\right) \tag{7.30}$$

By denoting $\xi_k \triangleq \operatorname{atan2}\left((y_{T_k} - y_{S_k}), (x_{T_k} - x_{S_k})\right)$, considering $z_k + \phi_{S_k} \in (-\pi, \pi]$, and following the standard formulas to compute the pdf of functions of random variables [148], the

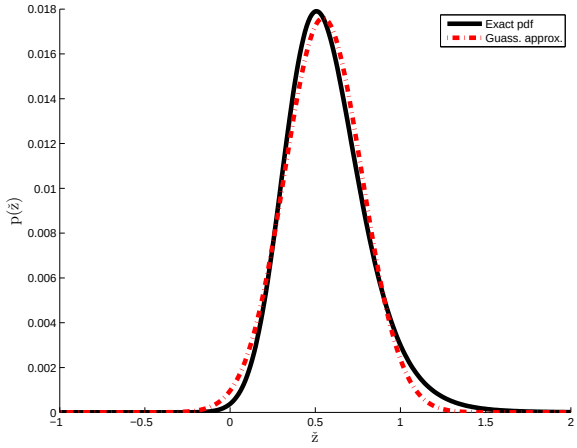


Figure 7.2: An example of approximating the pdf of transformed measurements by a Gaussian pdf. In this case, $\xi = 0.5$ and $\sigma_\theta = 10$ deg. In addition, the Kullback-Leibler divergence (KLD) between these two pdfs is only 0.0447, which indicates the difference between the two distributions is small.

likelihood distribution of the transformed measurement is given by:

$$p(\check{z}_k|\mathbf{x}_k) = \begin{cases} \frac{\mathcal{N}(\tan^{-1}(\check{z}_k);\xi_k,\sigma^2_{\theta_k})+\mathcal{N}(\tan^{-1}(\check{z}_k)-\pi;\xi_k,\sigma^2_{\theta_k})}{1+\check{z}_k^2}, & \text{if } \check{z}_k \geq 0 \\ \frac{\mathcal{N}(\tan^{-1}(\check{z}_k);\xi_k,\sigma^2_{\theta_k})+\mathcal{N}(\tan^{-1}(\check{z}_k)+\pi;\xi_k,\sigma^2_{\theta_k})}{1+\check{z}_k^2}, & \text{if } \check{z}_k < 0 \end{cases} \tag{7.31}$$

Clearly, $p(\check{z}_k|\mathbf{x}_k)$ is not Gaussian (which results from the tangent of a Gaussian random variable), but it can be well approximated by a Gaussian pdf by matching the first- and second-order moments. This is done by linearizing (7.30) around the expected value of the noise, i.e.,

$$\check{z}_k \simeq \frac{y_{T_k} - y_{S_k}}{x_{T_k} - x_{S_k}} + \bar{n}_k \triangleq \bar{z}_k \tag{7.32}$$

where $\bar{n}_k \triangleq \sec^2(z_k + \phi_{S_k})n_{\theta_k}$, is zero-mean white Gaussian noise with variance $\bar{\sigma}_k^2 \triangleq \sec^4(z_k + \phi_{S_k})\sigma^2_{\theta_k}$, i.e., $\bar{n}_k \sim \mathcal{N}(0, \bar{\sigma}_k^2)$. We term this approximation (7.32) an *inferred* measurement which is in the desired rational form. As illustrated in Fig. 7.2, this approximation is reasonably accurate, particularly for scenarios of high signal-to-noise ratios. Moreover, the local minimum of (7.20) attained based on the inferred measurement is very close to that using the corresponding original bearing measurement. This can be seen from Fig. 7.1(b), where one of the analytic local minima computed using the inferred measurement almost coincides with the MAP estimate for the current state that instead uses the original bearing measurement. This

further confirms that the inferred measurement is a reasonably good approximation to the original bearing measurement in solving (7.20).[3] Moreover, the inferred measurement is used only for finding hypotheses of the trajectory, not for estimating the state.

In what follows we use the inferred bearing measurement (7.32) [instead of (7.6)] to compute the analytic solutions for minimizing (7.20). In particular, with $\mathbf{P}^{-1}_{\mathbf{pp}_{k|k-1}} \triangleq \begin{bmatrix} s_1 & s_3 \\ s_3 & s_2 \end{bmatrix}$, (7.20) can be written as:

$$c(x_{T_k}, y_{T_k}) = \frac{1}{2}\Big(s_1(x_{T_k} - \hat{x}_{T_{k|k-1}})^2 + s_2(y_{T_k} - \hat{y}_{T_{k|k-1}})^2 + 2s_3(x_{T_k} - \hat{x}_{T_{k|k-1}})(y_{T_k} - \hat{y}_{T_{k|k-1}})\Big) + \frac{1}{2\bar{\sigma}_k^2}\left(\bar{z}_k - \frac{y_{T_k} - y_{S_k}}{x_{T_k} - x_{S_k}}\right)^2 \tag{7.33}$$

Based on the optimality conditions, i.e., setting the derivatives of $c(x_{T_k}, y_{T_k})$ with respect to the two optimization variables to zero, and performing simple algebraic manipulations, we have:

$$\begin{aligned} \frac{\partial c}{\partial x_{T_k}} &= s_1(x_{T_k} - \hat{x}_{T_{k|k-1}}) + s_3(y_{T_k} - \hat{y}_{T_{k|k-1}}) + \frac{1}{\bar{\sigma}_k^2}\left[\frac{\bar{z}_k(y_{T_k} - y_{S_k})}{(x_{T_k} - x_{S_k})^2} - \frac{(y_{T_k} - y_{S_k})^2}{(x_{T_k} - x_{S_k})^3}\right] = 0 \\ \Rightarrow\ & s_1(x_{T_k} - \hat{x}_{T_{k|k-1}})(x_{T_k} - x_{S_k})^3 + s_3(x_{T_k} - x_{S_k})^3(y_{T_k} - \hat{y}_{T_{k|k-1}}) + \\ & \frac{1}{\bar{\sigma}_k^2}\left[\bar{z}_k(x_{T_k} - x_{S_k})(y_{T_k} - y_{S_k}) - (y_{T_k} - y_{S_k})^2\right] = 0 \end{aligned} \tag{7.34}$$

$$\begin{aligned} \frac{\partial c}{\partial y_{T_k}} &= s_2(y_{T_k} - \hat{y}_{T_{k|k-1}}) + s_3(x_{T_k} - \hat{x}_{T_{k|k-1}}) - \frac{1}{\bar{\sigma}_k^2}\left[\frac{\bar{z}_k}{x_{T_k} - x_{S_k}} - \frac{(y_{T_k} - y_{S_k})}{(x_{T_k} - x_{S_k})^2}\right] = 0 \\ \Rightarrow\ & s_2(x_{T_k} - x_{S_k})^2(y_{T_k} - \hat{y}_{T_{k|k-1}}) + s_3(x_{T_k} - x_{S_k})^2(x_{T_k} - \hat{x}_{T_{k|k-1}}) - \\ & \frac{1}{\bar{\sigma}_k^2}\left[\bar{z}_k(x_{T_k} - x_{S_k}) - (y_{T_k} - y_{S_k})\right] = 0 \end{aligned} \tag{7.35}$$

From (7.35), we can compute y_{T_k} in terms of x_{T_k} as follows:

$$y_{T_k} = \frac{-\bar{\sigma}_k^2 s_3(x_{T_k} - x_{S_k})^2(x_{T_k} - \hat{x}_{T_{k|k-1}}) + \bar{z}_k(x_{T_k} - x_{S_k}) + \bar{\sigma}_k^2 s_2(x_{T_k} - x_{S_k})^2 \hat{y}_{T_{k|k-1}} + y_{S_k}}{1 + \bar{\sigma}_k^2 s_2(x_{T_k} - x_{S_k})^2} \tag{7.36}$$

[3] The inferred measurement model (7.32) does not consider the special case of $x_{T_k} = x_{S_k}$, which however has low probability of occurrence in practice.

Substitution of (7.36) into (7.34) yields a rational equation, whose denominator is always non-zero. Thus, we only need to consider the numerator which is an eighth-order univariate polynomial in x_{T_k}:

$$0 = f(x_{T_k}) = \sum_{i=0}^{8} a_i x_{T_k}^i \tag{7.37}$$

where a_i, $i = 0, 1, \ldots, 8$, are the coefficients expressed in terms of the known quantities, $\bar{z}_k$, $\bar{\sigma}_k$, s_1, s_2, s_3, $\hat{x}_{T_{k|k-1}}$, $\hat{y}_{T_{k|k-1}}$, x_{S_k}, and y_{S_k} [77]. The roots of $f(x_{T_k})$ can be found from the eigenvalues of the corresponding 8×8 companion matrix [44]. Although there exist 8 solutions for x_{T_k} and thus 8 solutions for y_{T_k}, as it depends injectively on x_{T_k} [see (7.36)], we only need to consider the pairs (x_{T_k}, y_{T_k}) that correspond to real eigenvalues of the companion matrix. Following the same reasoning as in the case of range-only tracking, since some of these solutions could be local maxima and/or saddle points, the second-order derivative test [16] is employed to extract the minima. Finally, once we determine all the local minima for the target position, we compute the corresponding estimates for the higher-order position derivatives via (7.19). Moreover, the following lemma provides a tighter upper bound for the maximum number of local minima for the case of bearing-only tracking.

Lemma 7.4.2. *There are at most 7 local minima for* (7.33).

Proof. According to the MPT (see Appendix E.1 and Theorem 5.2 in [81]), for a coercive $\mathcal{C}^1$ function, there exists a third critical point which is not a local minimum between any two strict local minima. It can be verified that the cost function (7.33) $\in \mathcal{C}^1(\mathbb{R}^2 \backslash \{x_{T_k} = x_{S_k}\})$ is coercive, and therefore at least one of the 8 critical points cannot be a local minimum, leaving a maximum number of 7 local minima. □

Note that due to its rational form, the inferred bearing measurement (7.32) is symmetric with respect to the sensor, while the original bearing measurement (7.5) is different in different quadrants. This can result in more local minima of (7.33) than those of (7.20). To discard the spurious local minima resulting from the symmetry of the inferred measurement, we can employ the Mahalanobis distance test [14]. As a result, we have never observed more than 4 local minima in practice.

7.5 A bank of MAP estimators

As discussed in the preceding section, due to the nonlinearity of range and bearing measurements, the incremental one-step MAP problem (7.17), and thus the original batch (multi-step) MAP problem (7.8), may have multiple local minima that correspond to the modes of the posterior pdf. Any iterative algorithm (e.g., Gauss-Newton) used in the batch-MAP estimator only converges to the global optimum and hence the true MAP estimate, if the initial estimate $\hat{\mathbf{x}}_{0:k|k}^{(0)}$ is within its region of attraction. However, in general, there exists no systematic method for determining an initial estimate that can always ensure convergence to the global optimum. As a result, the standard batch-MAP estimator when used for target tracking can become inconsistent and even diverge if no good initial estimate is provided. This is confirmed by the simulation and experimental results presented in Sections 7.6 and 7.7.

To mitigate the aforementioned issue, in this section, we propose a general *linearized* estimation framework for tracking multiple local minima (modes). Within this framework, we develop a bank of MAP estimators for the particular problem of target tracking. The key idea of this approach is to use the analytically-computed local minima at each time step (see Section 7.4.2) as guidance to find and track the most probable hypotheses of the target trajectory, thus improving estimation performance. Specifically, at time-step $k-1$, based on (7.15) and (7.16), we first propagate the current state estimate corresponding to the i-th solution and its covariance matrix, $\hat{\mathbf{x}}_{k-1|k-1}^{[i]}$ and $\mathbf{P}_{k-1|k-1}^{[i]}$, $i = 1, 2, \ldots, m$ (m is the number of estimators in the bank at time-step $k-1$). Then, once a new measurement becomes available, the propagated state estimate and covariance, $\hat{\mathbf{x}}_{k|k-1}^{[i]}$ and $\mathbf{P}_{k|k-1}^{[i]}$, are used as the prior in (7.17). Next, we use the algebraic-geometry methods presented in Section 7.4.2 to determine all the local minima of (7.17) analytically, denoted by $\mathbf{x}_k^{[j]}$, $1 \leq j \leq \alpha m$ (see Lemma 7.4.1 and 7.4.2, $\alpha = 2$ and 7, respectively). Finally, for each of these solutions, we employ the Gauss-Newton approach that uses the latest estimates of the trajectory corresponding to this solution as the initial value and all the available original measurements, to refine the entire state estimates $\hat{\mathbf{x}}_{0:k|k}^{[j]}$ up to current time-step k [see (7.8)].

This procedure incrementally evolves over time, and at every time step, generates at most αm trajectory estimates. In the end, we will have multiple candidates of the MAP estimate, among which the one with the least cost is selected as the best estimate for the global optimum (and thus for the true state). Algorithm 4 outlines the main steps of the proposed algorithm.

Algorithm 4 A Bank of MAP Estimators for Target Tracking

1: At each time-step k:
2: Propagate the current target state estimate and covariance via (7.15) and (7.16).
3: Analytically determine all the local minima of (7.17).
4: For each of the local minima, refine the corresponding entire state (trajectory) estimates and covariance, by employing the Gauss-Newton approach that uses the latest state estimates corresponding to this solution as the initial guess, and also compute the MAP cost (7.8).
5: In the end, select the estimate in the bank with the least cost as the resulting MAP estimate.

7.5.1 Computational cost reduction

In the worst case, the total number of analytic solutions, and thus MAP estimators in the bank, grows exponentially with time. In addition, as the target continuously moves, the size of the state vector $\mathbf{x}_{0:k}^{[i]}$ of each MAP estimator increases linearly with time. In order to make the algorithm suitable for real-time applications, in what follows, we present an effective pruning scheme, as well as the process of marginalization of old, matured states, to reduce the computational cost of the proposed algorithm.

Pruning scheme

In practice, the number of physically different trajectory hypotheses is significantly lower than the exponential number of hypotheses generated by the estimator, since many different initial guesses reside within the same basin of attraction. Additionally, we observe that in general, if two MAP estimators in the bank have similar costs, the trajectory estimates are also close. Therefore, we first aggregate the trajectory estimates of which the corresponding costs are equal within a tolerance, and retain one representative trajectory of each such group while discarding the others. In addition, we also employ the K-means algorithm [42] to cluster the remaining estimated trajectories into two groups based on their costs and remove the (outlier) group which has larger costs. These two steps, aggregation and clustering, are repeated, until the number of MAP estimators in the bank is within the threshold denoted by m_{max}.[4]

[4] Simulation results have shown that the aggregation is so effective that most of the time there is no need to perform the clustering.

Marginalization of old states

To further reduce the computational complexity, we also employ a marginalization process that removes the old, matured states from the state vector of each batch-MAP estimator in the bank (see Section 5.3 and [38,163]). In particular, suppose that marginalization of old states is carried out at time-step k_o, when all the measurements during the time interval $[0, k_o]$ are available. Then, as the robot keeps moving and collects new measurements in the time interval $[k_o+1, k]$, the MAP estimation takes place at time-step k. To facilitate our derivations, we first define the notations that we will be using. The old states that are marginalized out at time-step k_o are denoted by $\mathbf{x_M} \triangleq \mathbf{x}_{0:k_m}$. The remaining states that stay active in the sliding window after marginalization are denoted by $\mathbf{x_R} \triangleq \mathbf{x}_{k_m+1:k_o}$. Finally, the new target states that are added into the state vector during $[k_o+1, k]$ are denoted by $\mathbf{x_N} \triangleq \mathbf{x}_{k_o+1:k}$. At time-step k, the sliding window contains the states $\mathbf{x_R}$ and $\mathbf{x_N}$, and the estimator computes the batch-MAP estimate by minimizing a cost function similar to (7.8):

$$c(\mathbf{x}_{0:k}) = c(\mathbf{x_M}, \mathbf{x_R}, \mathbf{x_N}) = c_{\mathbf{M}}(\mathbf{x_M}, \mathbf{x_R}) + c_{\mathbf{N}}(\mathbf{x_R}, \mathbf{x_N}) \tag{7.38}$$

where we have decomposed the cost function into two terms: $c_{\mathbf{M}}(\mathbf{x_M}, \mathbf{x_R})$ that contains all quadratic terms that involve states in $\mathbf{x_M}$ only, as well as terms involving the last state in $\mathbf{x_M}$ and the first state in $\mathbf{x_R}$; and $c_{\mathbf{N}}(\mathbf{x_R}, \mathbf{x_N})$ that contains all quadratic terms that involve states in $\mathbf{x_R}$ only, states in $\mathbf{x_N}$ only, and terms involving the last state in $\mathbf{x_R}$ and the first state in $\mathbf{x_N}$. It is important to note that there is no quadratic term jointly involving states in $\mathbf{x_N}$ and $\mathbf{x_M}$, since the target states marginalized at time-step k_o do not participate in any measurement after that time. Thus, we have:

$$\min_{\mathbf{x_M}, \mathbf{x_R}, \mathbf{x_N}} c(\mathbf{x_M}, \mathbf{x_R}, \mathbf{x_N}) = \min_{\mathbf{x_R}, \mathbf{x_N}} \left(c_{\mathbf{N}}(\mathbf{x_R}, \mathbf{x_N}) + \min_{\mathbf{x_M}} c_{\mathbf{M}}(\mathbf{x_M}, \mathbf{x_R}) \right) \tag{7.39}$$

We then solve for $\mathbf{x}^\star_{\mathbf{M}} = \arg\min_{\mathbf{x_M}} c_{\mathbf{M}}(\mathbf{x_M}, \mathbf{x_R})$ which *only* depends on $\mathbf{x_R}$. This results in an approximately equivalent cost function of (7.39), $c'_{\mathbf{N}}(\mathbf{x_R}, \mathbf{x_N})$, which does *not* depend on $\mathbf{x_M}$ and whose minimization can be carried out by the Gauss-Newton method (see Section 7.3.3). The approximation in this process is introduced due to the fact that $c_{\mathbf{M}}$ is permanently approximated by its second-order Taylor series expansion, and the marginalized states $\mathbf{x_M}$, as well as all the measurements that directly involve them are discarded. Now it becomes clear that due to the marginalization, each batch-MAP estimator in the bank has constant computational requirements, which depend only on the size of the sliding window. This, along with

pruning, results in *constant* computational complexity for the proposed bank of MAP estimators, compared to *linear* for the standard (non-marginalized) batch-MAP estimator (due to the sparse, banded structure of the Hessian matrix) (see Section 7.3.3).

7.6 Simulation results

A series of Monte-Carlo simulations were conducted under various conditions, in order to demonstrate the capability of the proposed algorithm to improve tracking performance. We used RMSE and NEES [14] as the metrics for evaluating estimators' performance. In the following simulation tests, we adopted a zero-acceleration motion model for the target [14]:

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{w}(t) \tag{7.40}$$

where

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\ \mathbf{G} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},\ \mathbf{x}(t) = \begin{bmatrix} x_T(t) \\ y_T(t) \\ \dot{x}_T(t) \\ \dot{y}_T(t) \end{bmatrix}$$

and $\mathbf{w}(t) = \begin{bmatrix} w_x(t) & w_y(t) \end{bmatrix}^T$ is zero-mean white Gaussian noise with covariance $\mathbb{E}\left[\mathbf{w}(t)\mathbf{w}(\tau)^T\right] = q\mathbf{I}_2\delta(t-\tau)$, where $q = 1\ \left(\frac{\mathrm{m}}{\mathrm{sec}^2}\right)^2 \frac{1}{\mathrm{Hz}}$, and $\delta(t-\tau)$ is the Dirac delta function. In the implementation, we discretize this continuous-time system model (7.40) with time step $\Delta t = 0.1$ sec. The initial true target state is $\mathbf{x}_0 = \begin{bmatrix} 0 & 0 & -5 & 5 \end{bmatrix}^T$, while the initial estimate of the target state is randomly generated from a Gaussian pdf, $\mathcal{N}(\mathbf{x}_0, \mathbf{P}_{0|0})$, where $\mathbf{P}_{0|0} = 10^3\mathbf{I}_4$ is the initial covariance of the state estimate. Similar to [47], we chose a circular sensor trajectory with perfectly known poses for this simulation. Fig. 7.3 shows the trajectories of the target and sensor in one typical realization of Monte-Carlo simulations.

For the results presented in this section, we performed 100 Monte-Carlo simulations, and compared four different estimators. During each Monte-Carlo run, all the estimators process the same data, to ensure a fair comparison. The compared estimators are: (1) the standard EKF, (2) the standard batch-MAP estimator that incrementally uses the EKF estimates (i.e., the current EKF estimate along with the MAP estimates of the past states) as the initial value as well as employs the marginalization process as in (4), (3) the sampling importance resampling

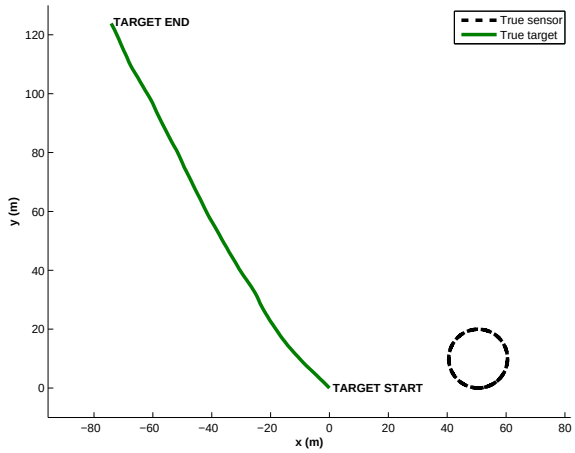
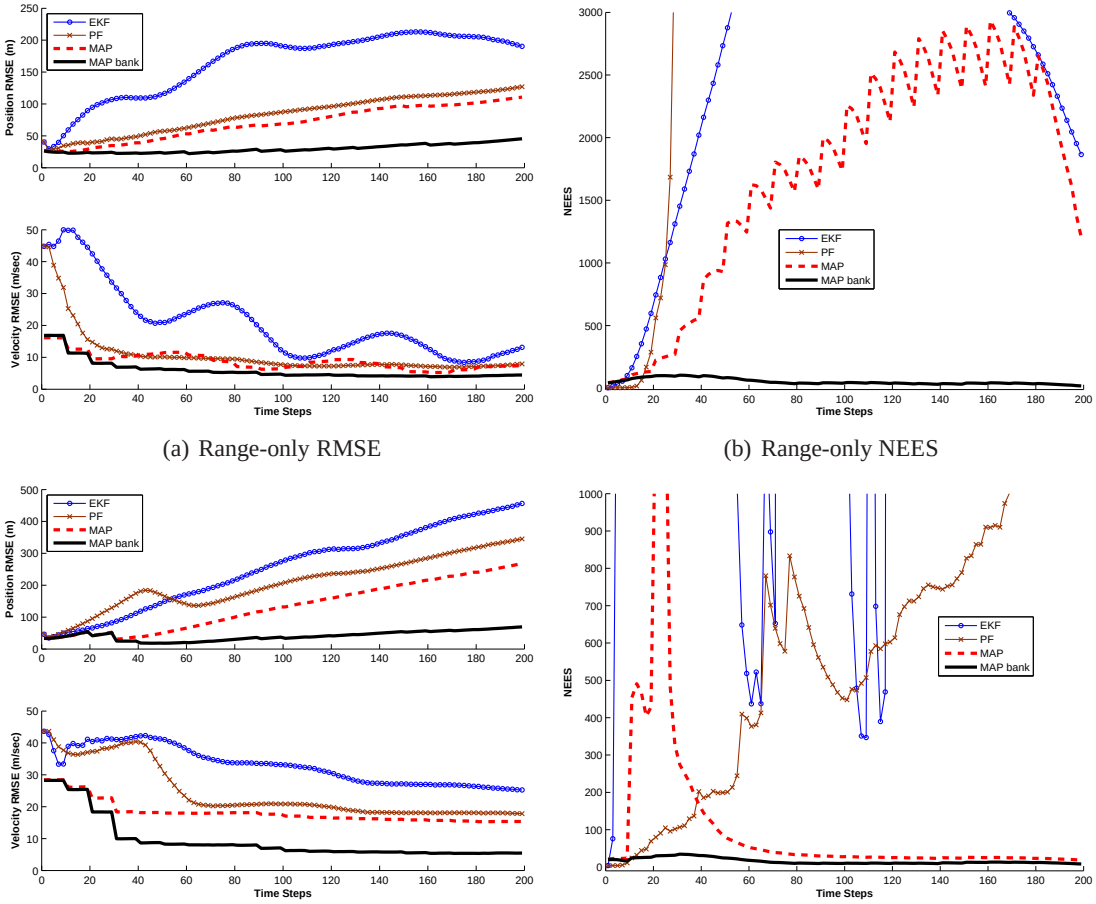


Figure 7.3: The trajectories of the target and sensor obtained from one typical realization of the 100 Monte-Carlo simulations.

(SIR)-PF with 3000 particles [8], and (4) the proposed bank of MAP estimators with pruning ($m_{\max} = 10$) and marginalization (sliding window of 25 time steps). Note that in both MAP estimators, the maximum number of Gauss-Newton iterations allowed was set to 20. In this simulation, we implemented the standard (bootstrap) SIR-PF [50] that uses the prior distribution as the proposal distribution to draw particles and employs systematic resampling at every time step. Moreover, to alleviate the particle depletion problem, we also dithered the sensor noise (i.e., increasing noise covariance). We have examined different resampling schemes such as Ripley's and stratified resampling [58], but found negligible performance difference.

In what follows, we present the comparison results for both range-only and bearing-only target tracking. In the case of range-only tracking, the standard deviation of the distance-measurement noise was equal to 10% of the sensor-to-target distance, while in the case of bearing-only tracking, the standard deviation of the bearing-measurement noise was equal to 10 deg. Notice that the sensor-noise levels selected for these simulations are larger than what is typically encountered in practice. This was done purposefully, since higher noise levels lead to larger estimation errors, which can make the effects of estimator's inconsistency and divergence more apparent.

Specifically, Fig. 7.4 shows the Monte-Carlo results of the four estimators. As evident from this figure, the standard EKF estimates are inaccurate, diverge from the ground truth, and become inconsistent. The standard batch-MAP estimator, incrementally using the EKF



(a) Range-only RMSE

(b) Range-only NEES

(c) Bearing-only RMSE

(d) Bearing-only NEES

Figure 7.4: Target tracking Monte-Carlo results. It is clear that the proposed algorithm performs substantially better than its competitors, in terms of both accuracy (RMSE) and consistency (NEES). Note that for clarity of presentation, only the portions of the NEES lines that are within certain thresholds are plotted.

estimate as the initial guess, has significantly improved performance compared to the EKF, mostly due to the continuous relinearization of the past trajectory. As expected, the PF attains better estimation accuracy than the EKF. This is due to the fact that each particle in the PF essentially represents a hypothesis of the target state, and thus the PF is more likely to converge to the optimal solution. However, it does not always work as well as the standard batch-MAP estimator [see Figs. 7.4(a) and 7.4(c)], in part because it does not allow smoothing the old

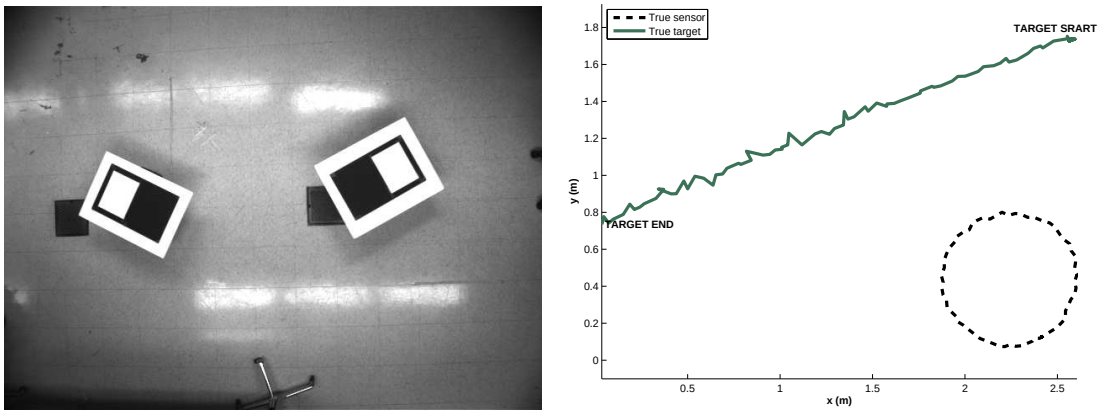
Table 7.1: Computational cost and estimation accuracy

	Runtime (sec)	Position Est. Err. (m)	Velocity Est. Err. (m/sec)
Range-only tracking			
EKF	0.0013	164.5105	20.8169
MAP	0.0764	69.7182	8.7626
PF	0.5036	83.2766	10.6280
Bank of MAP	0.3460	29.8041	5.9631
Bearing-only tracking			
EKF	0.0012	254.8551	32.9720
MAP	0.0628	134.4449	18.1967
PF	0.5590	206.6339	24.3979
Bank of MAP	0.5118	41.6355	9.3550

state estimates using newly available measurements.[5] Note also that the NEES of the PF is not necessarily better than that of the EKF, primarily due to the numerical issue incurred in the simulation that the covariance matrices of the PF computed from particles become ill-conditioned. Most importantly, the bank of MAP estimators performs substantially better than its competitors, in terms of both accuracy (RMSE) and consistency (NEES). This is attributed to the good initial estimates attained through the algebraic methods (see Section 7.4.2).

Finally, using the same simulation setup as described above, we compared the computational requirements of the proposed bank of MAP estimators and its competitors. We counted the CPU running time for a complete update of the EKF, the PF, the standard batch-MAP estimator, and the bank of MAP estimators (including the analytic determination of all local minima, batch-MAP refinement, pruning, and marginalization). Our Matlab implementation running on a Core2 Quad CPU required an average execution time for each estimator shown in Table 7.1. These results were obtained by averaging the CPU running time over all Monte-Carlo runs and over all time steps. As expected, the EKF and the standard MAP estimator which only track single hypothesis of the target trajectory, are much more computationally efficient than both the PF and the proposed bank of MAP estimators which instead track multiple hypotheses of

[5] Although particle-based smoothers exist, their computational requirements are significantly higher [58].



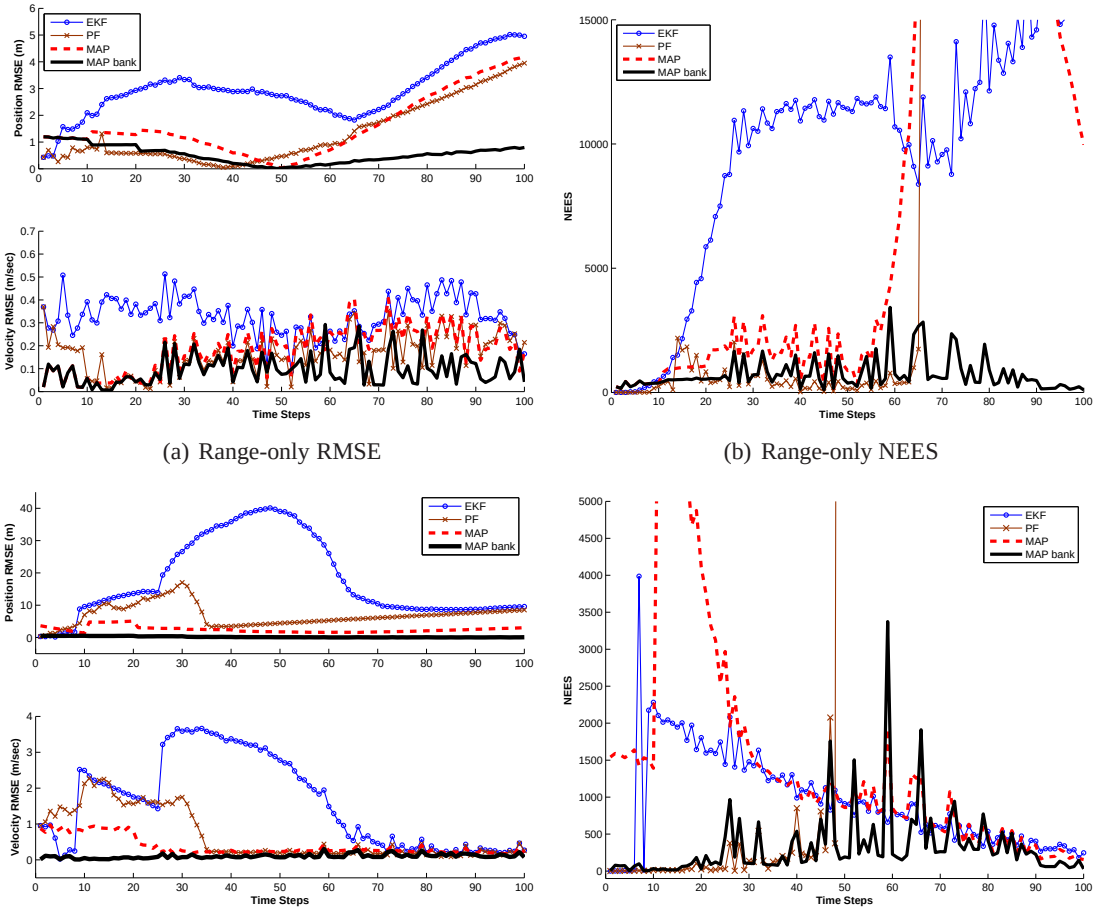
(a) (b)

Figure 7.5: Experimental setup: (a) Calibrated image of two Pioneer III robots (one acts as the target while the other is the sensor) with tracking patterns mounted on top of them. (b) Trajectories of the two robots (target and sensor) that move inside a 4 m $\times$ 2 m arena during the indoor experiment.

the target trajectory. However, their tracking performance is substantially worse than the proposed algorithm. Moreover, as compared to the PF, the proposed bank of MAP estimators not only is less computationally demanding, but also achieves significantly better performance [see Figs. 7.4(a) and 7.4(c), and Table 7.1]. Specifically, in the case of range-only tracking, the bank of MAP estimators achieves on average 60% higher position and 40% higher velocity estimation accuracy compared to the PF, at 30% less computational cost; similarly, in the case of bearing-only tracking, it achieves on average 80% higher position and 60% higher velocity estimation accuracy than the PF with comparable (8% less) computational cost.

7.7 Experimental results

In this section, we present a real-world experiment performed to further validate the proposed algorithm. During the test, two Pioneer-III robots, one acting as the target and the other serving as the sensor, moved in a rectangular area of 4 m $\times$ 2 m, within which the positions of the robots were being tracked by an overhead camera. For this purpose, rectangular tracking patterns were mounted on top of the robots and the vision system was calibrated in order to provide ground-truth measurements of the robots' poses in a global coordinate frame. The standard



(a) Range-only RMSE

(b) Range-only NEES

(c) Bearing-only RMSE

(d) Bearing-only NEES

Figure 7.6: Target tracking experimental results. It is clear that the proposed bank of MAP estimators performs better than its competitors. Note that for clarity of presentation, only the portions of the NEES lines that are within certain thresholds are plotted.

deviation of the noise in these measurements was approximately 0.5 deg for orientation and 0.01 m, along each axis, for position. The target robot was commanded to move along a straight line at a constant velocity of $v = 0.1$ m/sec, and thus a zero-acceleration motion model with $q = 0.01 \left(\frac{\text{m}}{\text{sec}^2}\right)^2 \frac{1}{\text{Hz}}$ was used to describe this motion [see (7.40)], while the sensor robot was operated to move on a circle. Fig. 7.5(a) shows the experimental setup, and Fig. 7.5(b) depicts the trajectories of the target and the sensor.

In this experiment, the initial estimate of the target state was set to be $\hat{\mathbf{x}}_{0|0} =$

$\begin{bmatrix} 1.2 & 0.95 & 0.7 & -0.8 \end{bmatrix}^T$ with covariance $\mathbf{P}_{0|0} = \mathbf{I}_4$. Relative distance and bearing measurements were produced synthetically using the differences in the positions of the target and the sensor, as these were recorded by the overhead camera, with the addition of noise. For the experimental results shown in the following, the distance and bearing measurements were corrupted by zero-mean white Gaussian noise, with standard deviation $\sigma_\rho = 0.1$ m and $\sigma_\theta = 5$ deg, respectively.

The same four estimators as in the previous simulation were implemented, and the comparative results are presented in Fig. 7.6. From the experimental results, it becomes clear that the bank of MAP estimators outperforms the standard EKF, the PF, and the standard MAP estimator, in terms of both accuracy (RMSE) and consistency (NEES). This agrees with the simulation results presented in the preceding section. Most importantly, both the experimental and simulation results confirm the significance of correctly finding and tracking multiple modes of the posterior pdf as well as reduced linearization errors in nonlinear estimation problems.

7.8 Summary

In order to improve consistency, nonlinear estimators should be able to track multimodal pdfs which often occur in nonlinear problems. However, this is not the case for many existing estimators (e.g., the EKF, and the MAP estimator). In this work, we have introduced a general estimation framework, a bank of MAP estimators, that simultaneously allows tracking multiple modes of the posterior pdf, and reduces linearization errors through relinearization of past measurements. We have applied it to the problems of both range-only and bearing-only target tracking. Due to the computational intractability of analytically solving the batch-MAP problem, we have employed a relaxation scheme that keeps past state estimates temporarily constant and incrementally solves a one-step minimization problem for the current state at every time step. This minimization is solved analytically using algebraic geometry methods. The analytically-computed local minima are then used to find accurate initial values for the bank of MAP estimators, thus focusing the available resources on tracking the most probable hypotheses of the target trajectory. Additionally, to reduce the computational cost of the proposed algorithm, we have employed hypothesis pruning along with marginalization of old states. Simulation and experimental results have shown that the proposed algorithm significantly outperforms the EKF, the batch-MAP estimator, as well as the PF, in terms of both accuracy and consistency.

Chapter 8

Analytically-Guided Sampling-Based PF for Target Tracking

In this chapter, we adapt the idea of analytically selecting hypotheses presented in the previous chapter to PFs. Within the PF framework, one critical design choice that greatly affects the filter's performance is the selection of the proposal distribution from which particles are drawn. We hence advocate the proposal distribution to be a Gaussian-mixture-based approximation of the posterior pdf after taking into account the most recent measurement. The novelty of our approach is that each Gaussian in the mixture is determined *analytically* to match the modes of the underlying unknown posterior pdf. As a result, particles are sampled along the *most probable* regions of the state space, hence reducing the probability of particle depletion. Part of this chapter has been published in [69].

8.1 Introduction

Particle filtering has become an increasingly popular *nonparametric* nonlinear estimation approach used in a wide range of applications such as target tracking [8, 39, 41, 58, 134]. A particle filter (PF) seeks to approximate the posterior pdf by a set of random samples (particles) and updates its estimate recursively in time. Within the sequential importance sampling (SIS) framework, one critical step is to design an appropriate *proposal distribution* (or importance density), which is used to draw particles for the next time step. Clearly, from the Bayesian filtering perspective, the best choice of the proposal distribution is the posterior pdf itself, which,

however, in general is intractable to evaluate analytically. In this chapter, we focus on formally designing a proposal distribution that better approximates the posterior pdf so as to improve the PF's performance.

Since, in general, it is also difficult to sample from the *optimal* proposal distribution, which minimizes the variances of the particles' weights conditioned on the trajectory and all available measurements [40], many, often *ad hoc*, choices of the proposal distribution are described in the literature. Typically, the prior pdf is used, which results in the standard (bootstrap) PF weighted by the measurement likelihood [50]. If, however, the prior is uninformative, the generated particles may not be able to sample the state space sufficiently. Specifically, when far from a mode of the posterior pdf, the weights of the particles decay quickly and lead to particle depletion (i.e., only a few particles have significant weights) [8]. As a result, the very few surviving particles are unable to appropriately represent the underlying posterior pdf, which may cause filter *inconsistency*.[1] In general, in order to converge to meaningful estimates, the standard PF requires using a large number of particles, and thus has high processing requirements. In cases where the posterior pdf is closer than the prior to the measurement likelihood, then using the likelihood, instead of the prior, as the proposal distribution often improves performance [8]. Alternatively, a Gaussian proposal distribution can be obtained by local linearization [40], based on which the unscented PF (UPF) [166] was introduced. The UPF employs the UKF or the EKF to generate the proposal distribution that takes into account the latest measurements and thus better approximates the posterior pdf. Similar ideas were also exploited in [115, 139, 168]. However, often due to the multimodal nature of the posterior pdf, the particles sampled from the UKF/EKF posterior pdf do not necessarily capture all the true posterior modes, which may degrade the UPF's performance.

The closest to the work presented in this chapter is the Gaussian sum PF (GSPF) [98] – which essentially is a bank of Gaussian PFs (GPFs) [97] – has been derived based on the concept of the Gaussian sum filter (GSF) [3]. Specifically, by assuming that the prior pdf can be represented as the sum of Gaussian distributions, the GSPF updates each distribution using the particles that are sampled, for example, from the corresponding prior pdf and weighted by the measurement likelihood. However, the GSPF does not provide a measurable criterion about how many Gaussian distributions are needed and most importantly, where to choose them. These

[1] Consistency is one of the primary criteria for evaluating the performance of an estimator. As defined in [14], an estimator is *consistent* if the estimation errors are zero-mean and have covariance smaller or equal to the one calculated by the estimator.

critical issues are addressed in the Analytically-Guided-Sampling (AGS)-PF introduced in this chapter.

In particular, the proposed AGS-PF efficiently utilizes the available computational resources by employing a small number of particles drawn from the *most probable* hypotheses about the estimated state, i.e., the AGS-PF samples the most likely regions of the state space. The key idea behind our approach is to employ a Gaussian mixture to approximate the posterior pdf, where each Gaussian is determined *analytically* and corresponds to a mode of the posterior pdf. Specifically, we first formulate and convert the nonlinear cost function of the MAP optimization problem for the current state into *polynomial* form, and then employ algebraic-geometry techniques [33] to analytically compute all the modes of the posterior pdf. Subsequently, we use a Gaussian mixture as the proposal distribution to approximate the posterior distribution. Each Gaussian component matches one mode of the posterior pdf, and its covariance is computed as the inverse of the Hessian matrix of the MAP problem. This analytically-determined proposal distribution provides a better approximation to the posterior pdf, because it not only takes into account the current measurement but also matches all the modes of the posterior pdf. Therefore, the particles drawn from this proposal distribution sample the most probable regions of the state space. Simulation and experimental results demonstrate that the AGS-PF significantly improves the performance in the cases of range-only and bearing-only target tracking. We stress that apart from the particular application of target tracking treated here, the proposed analytically-guided sampling scheme is applicable to a broad class of nonlinear estimation problems in robotics and computer vision that can be expressed in (or converted into) polynomial form.

8.2 Analytically-guided sampling-based particle filtering

In this section, we present a novel analytically-guided sampling scheme that consists of a Gaussian-mixture-based proposal distribution whose modes are determined analytically to match those of the posterior pdf. As a result, the new AGS-PF effectively focuses the available computational resources on the most probable regions of the state space. In what follows, we begin with a brief overview of the generic PF, and then describe our novel sampling scheme which can readily be integrated into the PF framework.

8.2.1 Particle filtering

A PF seeks to approximate the posterior distribution of the entire state trajectory, $p(\mathbf{x}_{0:k}|z_{0:k})$, sequentially in time, using a set of M weighted samples (particles), $\{\mathbf{x}_{0:k}^{[j]}\}_{j=1}^{M}$, where $\mathbf{x}_{0:k}^{T} \triangleq \begin{bmatrix}\mathbf{x}_0^T & \cdots & \mathbf{x}_k^T\end{bmatrix}$ denotes all the states up to time-step k, and $z_{0:k}$ denotes all the measurements in the time interval $[0, k]$. To this end, relying on sequential importance resampling (SIR), the PF generally requires three sequential steps to update its estimate (see [8,41,58], and Algorithm 5): Firstly, it draws particles for the next time step from a proposal distribution, $\pi(\mathbf{x}_{0:k}|z_{0:k})$, which is a critical design choice. Secondly, it assigns a weight to each particle in order to account for the fact that the proposal distribution is usually different from the true posterior pdf. Lastly, it performs resampling to multiply (or discard) particles with high (or low) weights.

As mentioned before, one of the main challenges in PFs is designing an appropriate proposal distribution. Even though numerous choices can be made, one typically requires that the proposal distribution has the following form in order to be amenable to recursive computation [41]:

$$\pi(\mathbf{x}_{0:k}|z_{0:k}) = \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k})\pi(\mathbf{x}_{0:k-1}|z_{0:k-1}) \tag{8.1}$$

It is well known that the curse of dimensionality can quickly make the particles too sparse to represent the posterior pdf (i.e., particle depletion). In practice, it is common to approximate the proposal distribution $\pi(\mathbf{x}_{0:k}|z_{0:k})$, by fixing the past trajectory $\mathbf{x}_{0:k-1}$ and only sampling the current state $\mathbf{x}_k$, i.e., using $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k})$. It has been proven in [40] that the *optimal* proposal distribution for the current state, with respect to minimizing the variance of the particles' weights, is in the form of a conditional pdf conditioned on the past trajectory and all the measurements:

$$\pi_{\text{opt}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k}) = p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k}) \tag{8.2}$$

Based on (8.1) as well as the common assumptions that the motion model is a Markov process and that the measurements are conditionally independent given the states, the (unnormalized) importance weight of the j-th particle is computed recursively as follows [41]:

$$w_k^{[j]} = \frac{p(\mathbf{x}_{0:k}^{[j]}|z_{0:k})}{\pi(\mathbf{x}_{0:k}^{[j]}|z_{0:k})} \propto w_{k-1}^{[j]} \frac{p(\mathbf{x}_k^{[j]}|\mathbf{x}_{k-1}^{[j]})p(z_k|\mathbf{x}_k^{[j]})}{\pi(\mathbf{x}_k^{[j]}|\mathbf{x}_{0:k-1}^{[j]}, z_{0:k})} \tag{8.3}$$

To summarize, a generic PF is outlined in Algorithm 5.

Algorithm 5 A generic particle filtering algorithm

1: **loop**
2: Draw particles $\{\mathbf{x}_k^{[j]}\}_{j=1}^M \sim \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k})$
3: Compute weights via (8.3), and normalize weights
4: Resample particles based on weights
5: **end loop**

In general, the optimal proposal distribution (8.2) is not available analytically or in a suitable form for efficient sampling, and one may choose (infinitely many) other possible distributions to approximate it. As mentioned before, a common choice is to sample from the state-transition prior distribution (motion model), i.e., $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$. In this case the weight is simply proportional to the measurement likelihood, $p(z_k|\mathbf{x}_k)$ [see (8.3)]. However, such a choice may easily lead to filter inconsistency (see Sections 8.4 and 8.5). To address this issue, in the following, we design an analytically-determined proposal distribution by taking the current measurement into account as well as matching all the modes of the posterior pdf.

8.2.2 Analytically-guided sampling scheme

Our choice of Gaussian-mixture-based proposal distribution is motivated by the following Gaussian sum theorem (see Theorem 4.1 of [3], p.214):

Theorem 8.2.1. *For a measurement model with additive Gaussian noise [e.g., see* (7.4)*], i.e.,* $z_k = h(\mathbf{x}_k) + v_k$*, where* $v_k \sim \mathcal{N}(0, \sigma_k^2)$*, and a prior pdf given by* $p(\mathbf{x}_k|z_{0:k-1}) = \sum_{i=1}^m \alpha_i \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}^{(i)}, \mathbf{P}_{k|k-1}^{(i)}\right)$*, the posterior pdf* $p(\mathbf{x}_k|z_{0:k})$ *approaches the Gaussian sum* $\sum_{i=1}^m \alpha_i' \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{(i)}, \mathbf{P}_{k|k}^{(i)}\right)$ *uniformly in* $\mathbf{x}_k$ *and* z_k *as* $\mathbf{P}_{k|k-1}^{(i)} \to 0$ *for* $i = 1, \ldots, m$*, where*

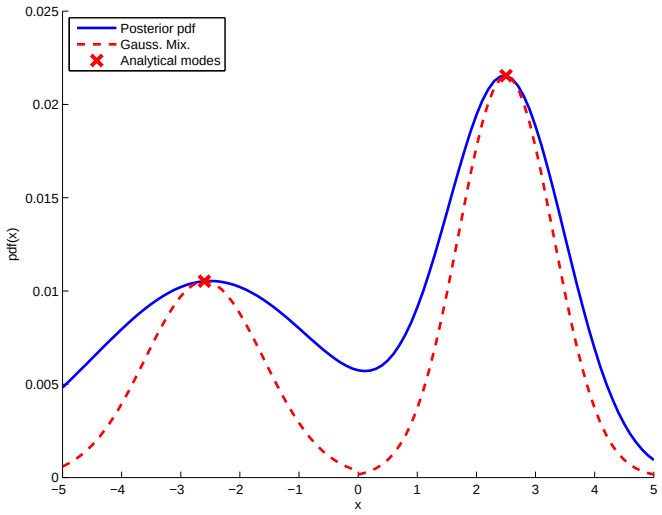


Figure 8.1: Illustration of the proposed analytically-determined proposal distribution that uses a Gaussian mixture to approximate the posterior distribution. The modes of the posterior pdf, which are analytically computed, are also used as the modes of the proposal distribution. Note that for visualization the plotted Gaussians of the mixture are scaled so that their modes also coincide along y-axis with those of the posterior pdf.

the mean, covariance and weight are computed as follows:

$$\hat{\mathbf{x}}_{k|k}^{(i)} = \hat{\mathbf{x}}_{k|k-1}^{(i)} + \mathbf{K}_k^{(i)} \left(z_k - h(\hat{\mathbf{x}}_{k|k-1}^{(i)}) \right) \tag{8.4}$$

$$\mathbf{P}_{k|k}^{(i)} = \mathbf{P}_{k|k-1}^{(i)} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)} \mathbf{P}_{k|k-1}^{(i)} \tag{8.5}$$

$$\mathbf{K}_k^{(i)} = \mathbf{P}_{k|k-1}^{(i)} \mathbf{H}_k^{(i)^T} \left(\mathbf{H}_k^{(i)} \mathbf{P}_{k|k-1}^{(i)} \mathbf{H}_k^{(i)^T} + \sigma_k^2 \right)^{-1} \tag{8.6}$$

$$\mathbf{H}_k^{(i)} = \nabla_{\mathbf{x}_k} h \Big|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}^{(i)}} \tag{8.7}$$

$$\alpha_i' = \frac{\alpha_i \beta_i}{\sum_{i=1}^{m} \alpha_i \beta_i} \tag{8.8}$$

$$\beta_i = \mathcal{N} \left(z_k; h(\hat{\mathbf{x}}_{k|k-1}^{(i)}), \mathbf{H}_k^{(i)} \mathbf{P}_{k|k-1}^{(i)} \mathbf{H}_k^{(i)^T} + \sigma_k^2 \right) \tag{8.9}$$

Based on this theorem, under mild assumptions, the Gaussian mixture can provide a good approximation to the posterior pdf. Hence, we propose to use a Gaussian mixture as the proposal distribution in the AGS-PF, while the novelty here is that we employ tools from algebraic geometry to analytically determine the modes of the posterior pdf, which are then used as the modes of the proposal distribution (see Fig. 8.1).

In particular, given that the prior pdf in the PF is approximated by a set of particles [see (8.10)], $\{\mathbf{x}_{k\ominus}^{[j]}, w_{k\ominus}^{[j]}\}_{j=1}^{M}$, we find a Gaussian mixture to approximate this distribution, which can be achieved, e.g., by clustering the M particles into m groups and then fitting a Gaussian to each group.

$$p(\mathbf{x}_k|z_{0:k-1}) \simeq \sum_{j=1}^{M} w_{k\ominus}^{[j]} \delta(\mathbf{x}_k - \mathbf{x}_{k\ominus}^{[j]}) \tag{8.10}$$

$$\simeq \sum_{i=1}^{m} \alpha_i \underbrace{\mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}^{(i)}, \mathbf{P}_{k|k-1}^{(i)}\right)}_{p^{(i)}(\mathbf{x}_k|z_{0:k-1})} \tag{8.11}$$

where $\alpha_i = \sum_{\mathbf{x}_{k\ominus}^{[j]} \in \text{group } i} w_{k\ominus}^{[j]}$, and $\delta(\cdot)$ is the Dirac delta function. Using Bayes' rule and the above Gaussian mixture approximation of the prior pdf (8.11), the posterior pdf can be approximated as follows:

$$\begin{aligned} p(\mathbf{x}_k|z_{0:k}) &\propto p(z_k|\mathbf{x}_k) p(\mathbf{x}_k|z_{0:k-1}) \\ &\simeq \sum_{i=1}^{m} \alpha_i \underbrace{p(z_k|\mathbf{x}_k) p^{(i)}(\mathbf{x}_k|z_{0:k-1})}_{\propto p^{(i)}(\mathbf{x}_k|z_{0:k})} \\ &\simeq \sum_{i=1}^{m} \sum_{\ell=1}^{n_i} \alpha'_{i_\ell} \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{(i_\ell)}, \mathbf{P}_{k|k}^{(i_\ell)}\right) \end{aligned} \tag{8.12}$$

It is important to point out that due to the nonlinearity of the measurement model, for each Gaussian distribution in the prior mixture, $p^{(i)}(\mathbf{x}_k|z_{0:k-1})$, the corresponding posterior pdf, $p^{(i)}(\mathbf{x}_k|z_{0:k})$, very often is a multi-modal, rather than a unimodal, distribution. To take this fact into account, in (8.12) we use n_i Gaussian distributions to approximate the i-th posterior pdf, $p^{(i)}(\mathbf{x}_k|z_{0:k})$, whose weights, α'_{i_ℓ}, are computed based on (8.8) but with appropriate normalization, i.e., $\alpha'_{i_\ell} = \frac{\alpha_i \beta_i}{\sum_{i=1}^{m} n_i \alpha_i \beta_i}$. Note that n_i is *analytically determined* as the number of modes of the i-th posterior pdf, rather than arbitrarily chosen as in most Gaussian sum filters such as the GSPF [98]. Here the index i_ℓ denotes the ℓ-th Gaussian of the i-th posterior pdf. Thus, a mixture of $n = \sum_{i=1}^{m} n_i$ Gaussian distributions is used to approximate the posterior pdf, which is then used as the proposal distribution in the proposed AGS-PF.

We now aim to analytically compute all the modes, $\hat{\mathbf{x}}_{k|k}^{(i_\ell)}$, ($\forall i = 1, \ldots, m$ and $\forall \ell = 1, \ldots, n_i$), of the i-th posterior pdf, $p^{(i)}(\mathbf{x}_k|z_{0:k})$, i.e., to solve the following one-time-step

MAP problem analytically:

$$\max_{\mathbf{x}_k} p^{(i)}(\mathbf{x}_k|z_{0:k}) \propto p^{(i)}(\mathbf{x}_k|z_{0:k-1})p(z_k|\mathbf{x}_k) \tag{8.13}$$

Exploiting the Gaussianity of the measurement noise along with the Gaussian approximation of the i-th element, $p^{(i)}(\mathbf{x}_k|z_{0:k-1})$, of the approximate prior (8.11), the MAP problem (8.13) is equivalent to the following nonlinear least-squares problem [76]:

$$\min_{\mathbf{x}_k} \frac{1}{2}||\mathbf{x}_k - \hat{\mathbf{x}}^{(i)}_{k|k-1}||^2_{\mathbf{P}^{(i)}_{k|k-1}} + \frac{1}{2}||z_k - h(\mathbf{x}_k)||^2_{\sigma_k^2} \tag{8.14}$$

where we have employed the notation $||\mathbf{a}||^2_{\mathbf{M}} \triangleq \mathbf{a}^T\mathbf{M}^{-1}\mathbf{a}$. Note that for a broad class of nonlinear estimation problems arising in robotics and computer vision, we can transform or convert (8.14) into polynomial form and then solve for all the local minima (corresponding to all the modes) analytically using algebraic geometry techniques [33].

Once we analytically find all the modes, $\hat{\mathbf{x}}^{(i_\ell)}_{k|k}$, of the i_ℓ-th Gaussian component of the posterior pdf (proposal distribution), we compute the corresponding covariance, $\mathbf{P}^{(i_\ell)}_{k|k}$, from the inversion of the Hessian matrix of (8.14) as follows:

$$\mathbf{P}^{(i_\ell)}_{k|k} = \left(\mathbf{P}^{(i)^{-1}}_{k|k-1} + \sigma_k^{-2}\mathbf{H}^{(i_\ell)^T}_k \mathbf{H}^{(i_\ell)}_k\right)^{-1} \tag{8.15}$$

where the measurement Jacobian, $\mathbf{H}^{(i_\ell)}_k$, is evaluated at the i_ℓ-th analytically-computed mode, $\hat{\mathbf{x}}^{(i_\ell)}_{k|k}$ [see (8.7)]. Based on the matrix inversion lemma [49], it is not difficult to see that (8.15) is precisely the standard EKF covariance update equation [14] [also see (8.5)].

8.3 AGS-PF for target tracking

We now apply the AGS-PF presented in the previous section to the particular problem of target tracking, to illustrate in detail the key idea of *analytically* determining the proposal distribution for sampling particles. In particular, in order to find the proposal distribution (Gaussian mixture), we first determine all the modes of the posterior pdf (proposal distribution), $\hat{\mathbf{x}}^{(i_\ell)}_{k|k}$, by employing our analytic approaches presented in the previous chapter (see Section 7.4.2) to analytically solve the problem (8.14). On the other hand, in order to obtain the covariance, $\mathbf{P}^{(i_\ell)}_{k|k}$, of the i_ℓ-th Gaussian component of the proposal distribution (8.15), we compute the measurement Jacobian, $\mathbf{H}^{(i_\ell)}_k$, as in (7.12), using the found modes, $\hat{\mathbf{x}}^{(i_\ell)}_{k|k}$.

Since the analytically-determined proposal distribution derived in the previous section not only takes into account the current measurement but also matches all the modes of the posterior pdf, it provides a better approximation to the posterior pdf, and thus the AGS-PF is expected to perform better than the standard PF as well as the GSPF. Specifically, at time-step $k-1$, we cluster the particles into m groups, e.g., using the K-means algorithm [42]. Here, m is a design choice selected based on the available computational resources. We compute the sample mean and covariance of the i-th group ($i = 1, \ldots, m$), and approximate this group by a Gaussian, $\mathcal{N}^{(i)} \triangleq \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}^{(i)}_{k-1|k-1}, \mathbf{P}^{(i)}_{k-1|k-1})$, $i = 1, \ldots, m$. Then, based on the *linear* motion model (7.2), we propagate each Gaussian to obtain the prior, $\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}^{(i)}_{k|k-1}, \mathbf{P}^{(i)}_{k|k-1})$ [see (7.15) and (7.16)]. When a new measurement becomes available, we analytically compute all the modes of the posterior pdf for each of the m Gaussians $\mathcal{N}^{(i)}$, by solving (8.14) (see Section 7.4.2). Once all the modes of the posterior pdf (and thus the proposal distribution) are determined, we compute the corresponding covariance based on (8.15) for each Gaussian component of the proposal distribution. Finally, after all the Gaussian components (modes and covariances) are specified, we use them as a proposal distribution to draw particles, $\{\mathbf{x}^{[j]}_k\}_{j=1}^{M_{i\ell}} \sim \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}^{(i_\ell)}_{k|k}, \mathbf{P}^{(i_\ell)}_{k|k})$, where $M_{i\ell}$ is the number of particles drawn from the i_ℓ-th Gaussian. For simplicity, $M_{i\ell}$ is set equal to the number of the particles originating from the i-th group after clustering, i.e., $M_{i\ell} = M_i$, though a more adaptive scheme (e.g., based on the particles' weights) may be used. In summary, the main steps of the AGS-PF target tracking are outlined in Algorithm 6.

Algorithm 6 Analytically-Guided-Sampling (AGS)-PF for Target Tracking

Require: Initialize particles by sampling from $p(\mathbf{x}_0)$

1: **loop**
2: Cluster particles into m groups using K-means, and fit a Gaussian to each group
3: Propagate each of the m Gaussians (means and covariances) to obtain the priors via (7.15) and (7.16)
4: For each group, given a new measurement, analytically determine the proposal distribution as a Gaussian mixture [see Sections 7.4.2 and (8.15)]
5: Draw particles from the analytically-determined proposal distribution (Gaussian mixture)
6: Compute weights via (8.3), prune particles, and normalize weights
7: Resample particles based on weights
8: **end loop**

Note that as seen from Section 7.4.2, we may find multiple modes of the posterior pdf for each group at each time step. Hence, the number of Gaussian components of the proposal distribution may be larger than the original number of clusters. This can result in an unbounded growth of particles over time in the worst case, since more and more particles may be generated for each group in the subsequent time steps. In order to keep the number of particles constant and reduce the computational cost of the AGS-PF, at every time step we *prune* out particles with low weights (see Algorithm 6). As a result, the AGS-PF can use substantially *fewer* particles while achieving significantly *better* performance than the standard PF (see Sections 8.4 and 8.5). This is attributed to the fact that the particles drawn from the analytically-determined proposal distribution that captures all the modes of the posterior pdf, are *effectively* sampled from the *most probable* regions of the state space.

8.4 Simulation results

A series of Monte-Carlo simulations were conducted under various conditions, in order to validate the capability of the proposed AGS-PF to improve tracking performance, using the performance evaluation metrics of NEES and RMSE [14]. In this simulation, we performed 100 Monte-Carlo simulations and compared three different PFs: (1) the standard (bootstrap) PF with $\mathbf{1000}$ particles, which uses the prior as the proposal distribution to draw particles from and employs the systematic resampling strategy [50]; (2) the GSPF [98] using the same number of particles as the standard PF and $m = 10$ Gaussians to represent the underlying distributions, which are initialized by clustering the initial particles into m groups and then fitting a Gaussian to each group; and (3) the proposed AGS-PF with $\mathbf{500}$ particles, which uses the analytically-determined proposal distribution as well as systematic resampling. In the AGS-PF, at each time step, we clustered the particles into $m = 10$ groups, equal to the number of Gaussians used in the GSPF. Note that in order to validate the effectiveness of the analytically-determined proposal distribution employed by the AGS-PF, significantly fewer particles were used in the AGS-PF, as compared to the standard PF and the GSPF. Despite this fact, as shown below, the AGS-PF attains substantially better performance than both the standard PF and the GSPF.

For the results presented in this section, we adopted a zero-acceleration motion model for

the target [14] [also see (7.40)]:

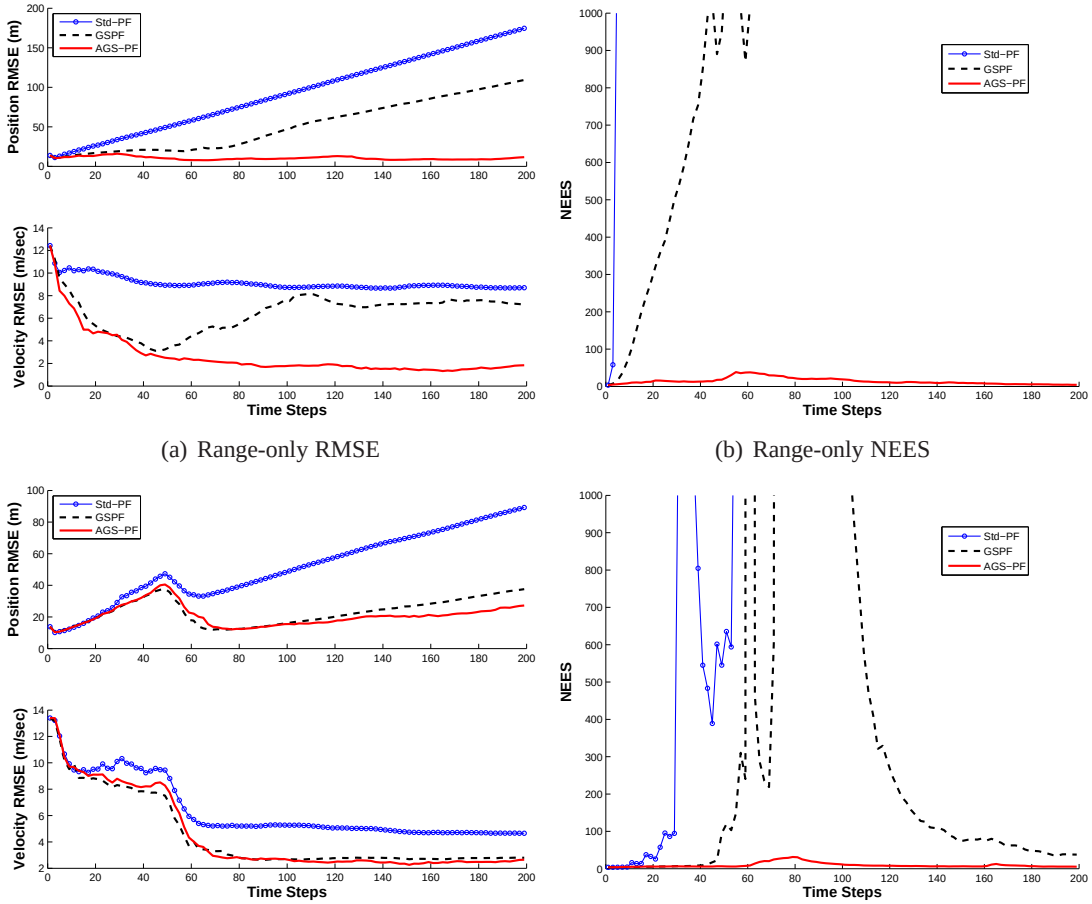
$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{w}(t) \tag{8.16}$$

where

$$\mathbf{x}(t) = \begin{bmatrix} x_T(t) \\ y_T(t) \\ \dot{x}_T(t) \\ \dot{y}_T(t) \end{bmatrix}, \ \mathbf{F} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ \mathbf{G} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and $\mathbf{w}(t) = \begin{bmatrix} w_x(t) & w_y(t) \end{bmatrix}^T$ is zero-mean, white Gaussian noise with covariance $\mathbb{E}\left[\mathbf{w}(t)\mathbf{w}(\tau)^T\right] = q\mathbf{I}_2\delta(t-\tau)$, where $q = 1\left(\frac{\mathrm{m}}{\mathrm{sec}^2}\right)^2\frac{1}{\mathrm{Hz}}$, and $\delta(t-\tau)$ is the Dirac-delta function. In our implementation, we discretize this continuous-time system model (8.16) with time step $\Delta t = 0.1$ sec. The initial true target state is $\mathbf{x}_0 = \begin{bmatrix} 0 & 0 & -5 & 5 \end{bmatrix}^T$, while the initial estimate of the target state is randomly generated from a Gaussian pdf, $\mathcal{N}(\mathbf{x}_0, \mathbf{P}_{0|0})$, where $\mathbf{P}_{0|0} = 100\mathbf{I}_4$ is the initial covariance of the state estimate. Similarly to [47], we chose a circular sensor trajectory with perfectly known sensor positions for the simulations. Fig. 7.3 shows the trajectories of the target and sensor in one typical realization of the Monte-Carlo simulations. The standard deviation of the distance-measurement noise was equal to $\sigma_\rho = 0.5$ m, while the standard deviation of the bearing-measurement noise was set to $\sigma_\theta = 3$ deg.

Fig. 8.2 shows the Monte-Carlo results of the three PFs. It is clear that the standard PF provides inaccurate estimates which are diverging from the ground truth and become inconsistent. As explained before, the poor NEES performance of the standard PF is primarily due to the ill conditioning of the covariances computed from particles whose weights are small or particles that do not span all directions of the state space. As expected, the GSPF performs more accurately than the standard PF. Most importantly, the AGS-PF performs better than both the standard PF and the GSPF, in terms of accuracy (RMSE) and consistency (NEES). This is attributed to the analytically-determined proposal distribution which matches all the modes of the posterior pdf while taking into account the most recent measurements. It is interesting to note that the superior performance of the AGS-PF over the GSPF in bearing-only tracking, is not as pronounced as that in range-only tracking [see Figs. 8.2(a) and 8.2(c)]. This is due to the approximation incurred in the AGS-PF using the inferred, instead of original, bearing measurements (7.32) to determine the Gaussian-mixture-based proposal distribution, which, however, is not the case in range-only tracking.



(a) Range-only RMSE

(b) Range-only NEES

(c) Bearing-only RMSE

(d) Bearing-only NEES

Figure 8.2: Target tracking Monte-Carlo results: The proposed AGS-PF algorithm performs substantially better than its competitors, in terms of both accuracy (RMSE) and consistency (NEES). Note that for clarity of presentation, only the portions of the NEES lines that are within a certain threshold are plotted.

Finally, using the same simulation setup as described above, we compared the computational requirements of the proposed AGS-PF and its competitors by measuring the CPU runtime for a complete update of all filters. Our Matlab implementations running on a Core i7 CPU of 2.67 GHz required an average execution time for each filter shown in Table 8.1. These results were obtained by averaging the CPU running time over all Monte-Carlo simulations and over all time steps. As compared to the standard PF and the GSPF, the proposed AGS-PF is not only computationally more efficient by using fewer particles but also achieves significantly

Table 8.1: Computational cost and estimation accuracy

	Runtime (sec)	Pos. Est. Err. (m)	Vel. Est. Err. (m/sec)
Range-only tracking			
Std-PF	0.2273	92.3522	9.1060
GSPF	0.1801	51.4123	6.5045
AGS-PF	0.1813	10.3799	2.5885
Bearing-only tracking			
Std-PF	0.2757	51.5319	6.3499
GSPF	0.2283	23.2511	4.4451
AGS-PF	0.1956	20.7163	4.4377

better tracking performance [see Figs. 8.2(a) and 8.2(c), and Table 8.1]. Specifically, in the case of range-only tracking, as compared to the standard PF and the GSPF, the AGS-PF attains on average 89% and 80% reduction in position estimation error, 71% and 60% reduction in velocity estimation error, while at 20% lower and same order of computational cost, respectively. Similarly, for bearing-only tracking, it achieves on average 60% higher position and 30% higher velocity estimation accuracy than the standard PF at 30% less computational cost; while attaining compatible estimation accuracy as the GSPF at 15% less computational cost.

8.5 Experimental results

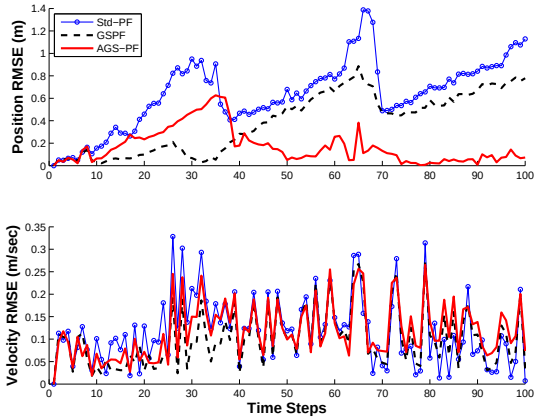
In this section, we conducted the same real-world experiment as in Section 7.7 to further validate the proposed AGS-PF. Specifically, during the test, two Pioneer-III robots, one acting as the target and the other serving as the sensor, moved in a rectangular area of 4 m $\times$ 2 m, within which the positions of the robots were tracked by an overhead camera. For this purpose, rectangular tracking patterns were mounted on top of the robots and the vision system was calibrated in order to provide ground-truth measurements of the robots' poses in a global coordinate frame. The standard deviation of the noise in these measurements was approximately 0.5 deg for orientation and 0.01 m, along each axis, for position. The target robot drove along a straight line at a constant velocity of $v = 0.1$ m/sec, and thus a zero-acceleration motion model with $q = 0.05 \left(\frac{\text{m}}{\text{sec}^2}\right)^2 \frac{1}{\text{Hz}}$ was used to describe this motion [see (8.16)], while the sensor robot moved

on a circle. Fig. 7.5(a) shows the experimental setup, and Fig. 7.5(b) depicts the trajectories of the target and the sensor. In this experiment, the initial estimate of the target state was set to $\hat{\mathbf{x}}_{0|0} = \begin{bmatrix} 2.5940 & 1.7374 & 0.0003 & -0.0001 \end{bmatrix}^T$ with covariance $\mathbf{P}_{0|0} = \mathbf{I}_4$. Relative distance and bearing measurements were produced synthetically using the differences in the true positions of the target and the sensor, as these were recorded by the overhead camera, with the addition of noise. For the results shown in this section, the distance and bearing measurements were corrupted by zero-mean white Gaussian noise, with standard deviation $\sigma_\rho = 0.1$ m and $\sigma_\theta = 2$ deg, respectively.

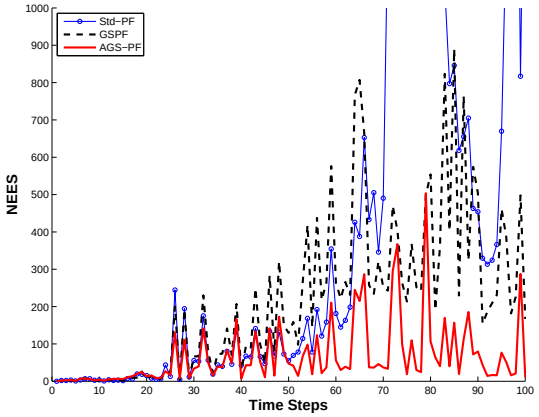
The same three PFs (i.e., the standard PF, the GSPF, and the proposed AGS-PF) as in the preceding simulation were implemented, and the comparative results obtained from this single-run experiment are presented in Fig. 8.3. From the experimental results, it becomes clear that the proposed AGS-PF outperforms the standard PF and the GSPF, in terms of both accuracy (RMSE) and consistency (NEES), which agrees with the simulation results presented in the previous section.

8.6 Summary

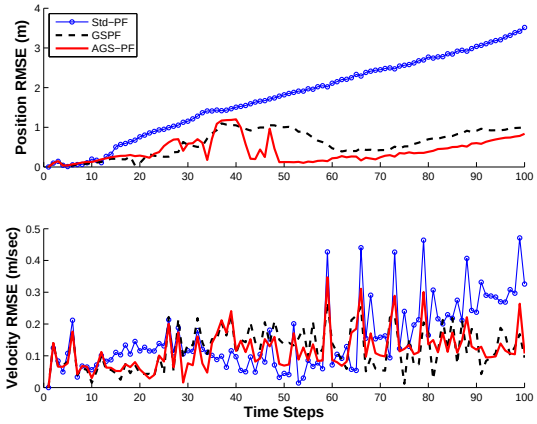
In this chapter, we have introduced a new AGS-PF, which uses a Gaussian mixture as the proposal distribution, each Gaussian corresponding to one of the analytically-computed modes of the posterior pdf. Using such proposal distribution, the AGS-PF draws its particles within the most probable regions of the state space. As a result, as compared to the standard PF and the GSPF, the AGS-PF attains better performance while requiring fewer computational resources. We applied this algorithm to the particular problems of range-only and bearing-only target tracking. Simulation and experimental results have demonstrated that the proposed approach outperforms the standard PF and the GSPF, in terms of accuracy, consistency and efficiency.



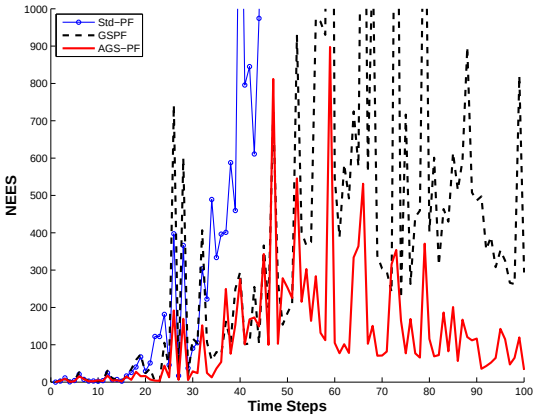
(a) Range-only RMSE



(b) Range-only NEES



(c) Bearing-only RMSE



(d) Bearing-only NEES

Figure 8.3: Target tracking experimental results: The proposed AGS-PF algorithm performs better than its competitors, in terms of both accuracy (RMSE) and consistency (NEES). Note that for clarity of presentation, only the portions of the NEES lines that are within a certain threshold are plotted.

Chapter 9

Concluding Remarks

9.1 Summary of contributions

The work presented in the preceding chapters has focused on investigating the fundamental causes of estimation inconsistency in the applications of robot localization and target tracking, and providing methodologies for improving estimator consistency. The main contributions of this work can be summarized as follows:

- **Observability-constrained estimators for robot localization**

  In Chapters 2 and 3, we studied in depth the EKF inconsistency of robot localization including SLAM and CL and proved for the first time ever that the standard linearized EKF system has an observable subspace of *higher* dimension than the corresponding nonlinear system. This indicates that the standard EKF gains *spurious* information from the available measurements and erroneously reduces the uncertainty of its estimates, which leads to inconsistency. Based on this key insight, in order to address this problem, we proposed an observability-based methodology. The underlying idea of this approach is computing the EKF propagation and measurement Jacobians so as to ensure that the linearized EKF system model has an observable subspace of *correct* dimension. To achieve this goal, we developed three different Observability-Constrained (OC)-EKFs, which compute the appropriate Jacobians either indirectly (i.e., by first finding appropriate linearization points used for computing Jacobians) or directly (i.e., by projecting the best-available measurement Jacobian onto the observable directions).

Proceeding similarly, we showed in Chapter 4 that the linear-regression-based SLAM system model used by the UKF has the same observability issue, i.e., the linear-regression-based system has observable subspace of higher dimension than the actual nonlinear system. Hence, we adapted the previous observability-based methodology and introduced a novel Observability-Constrained (OC)-UKF for SLAM. In particular, the observability constraints are enforced when constructing the optimal linear regression matrices in the LRKF framework. Moreover, in Chapter 5, we generalized the observability-based methodology from the filtering framework to the smoothing framework and developed an Observability-Constrained (OC)-SWF for SLAM. In this case, due to the marginalization used in the SWF, the standard SWF possesses different parameter observability properties from the optimal batch-MAP estimator. To address this issue, we impose the observability constraint in computing the Jacobian and thus the Hessian matrices, when employing Gauss-Newton to iteratively solve the nonlinear optimization problem.

Besides the *unobservable* systems such as SLAM and CL considered in Chapters 2-5, in Chapter 6, we generalized the observability-based methodology to a broad class of observable systems, i.e., discrete-time nonlinear systems with partial-state measurements. In particular, a novel decomposition of the observability matrix according to the measurement sources (sensors) reveals that the standard EKF acquires nonexistent information from each source's measurements, which degrades the performance (in terms of consistency and accuracy). Therefore, we adapted the observability-based idea and computed the EKF Jacobians so that the filter only acquires information of correct d.o.f. from the measurements of *each* source.

- **Analytically-selected multi-hypothesis target tracking**

  In Chapter 7, we studied the inconsistency issue of range-only and bearing-only target tracking, which are highly nonlinear estimation problems with non-convex cost functions and often with *multiple* local minima (corresponding to the modes of the posterior pdf). In such cases, we found that a standard linearized estimator (e.g., the EKF) becomes inconsistent primarily because it is able to find and track only *one* local minimum, without guarantee of global optimum. To address this issue, we provided a formal methodology for designing linearized estimators that analytically find and track multiple local minima. Within the category of linearized estimators, a bank of MAP estimators was developed.

By converting the nonlinear cost function into polynomial form, the bank of MAP algorithm analytically computes all the modes of the posterior pdf for the current state at each time step, and then uses these modes as high-quality initial estimates for the batch-MAP estimators in the bank to refine the entire trajectory. Furthermore, in Chapter 8 we adapted this analytic idea to nonparametric estimation and developed the Analytically-Guided-Sampling (AGS)-PF. The key idea of the AGS-PF is to employ an analytically-determined Gaussian mixture as proposal distribution which not only takes into account the most recent measurement but also matches all the modes of the posterior (optimal proposal) distribution. As a result, the AGS-PF can efficiently focus its particles on the most probable region in the state space.

With this work, we provide a solid theoretical framework for improving consistency of nonlinear estimators that will enable long-term consistent autonomous navigation even in GPS-denied environments, and will offer significant benefits for robots employed in various practical application domains.

9.2 Future research directions

Building upon the theoretical foundations developed in this work, in the future we will focus on developing efficient state estimation algorithms to enable long-term consistent autonomous navigation under various practical constraints, as well as expanding the same theoretical framework to distributed teams of reconfigurable arrays of networked (mobile) sensors or intelligent embedded systems. Although there are many open interesting questions along these lines, the following two research directions are particularly of importance and interest:

- **Resource-aware consistent vision-aided inertial navigation**

  In Chapters 2, 3, 4 and 5, we have primarily focused on robots navigating in 2D. Even though this assumption is satisfied when robots operate in most man-made environments (e.g., indoors, and on paved roads), an increasing number of applications require robots moving in 3D and thus estimating their 6 d.o.f. poses (positions and orientations). For example, the full 3D pose estimation is necessary for spacecraft [159], unmanned aerial vehicles [103], autonomous underwater vehicles [107], robots operating on rugged outdoor terrain [156], and in some cases, robots moving indoors [119]. Therefore, it will be

of practical importance to extend this work to 3D.

Over the past few decades, inertial navigation systems (INS) have been extensively used for estimating 3D robot pose in GPS-denied areas (e.g., underwater, indoor, in the urban canyon, and on other planets). Most INS rely on an inertial measurement unit (IMU) that measures the 3 d.o.f. rotational velocity and 3 d.o.f. linear acceleration of the robot on which it is rigidly attached. Unfortunately, simple integration of IMU measurements that are corrupted by noise and bias, often results in pose estimates unreliable for navigation purposes. Although high-accuracy IMUs do exist, they remain prohibitively expensive for widespread deployment. For this reason, it is common to aid an INS with an alternative sensor, such as a laser scanner, sonar, radar, or camera, whose measurements can be used to determine the robot motion with respect to the surrounding environments. Of these possible aiding sources, camera is an appealing choice since it is small, light-weight, inexpensive, and passive (energy efficient), while providing rich information. Hence, vision-aided INS (V-INS) is emerging as an important application [120].

Even though state-of-the-art V-INS algorithms can provide accurate pose estimates over short periods of time, they are not ready for long-term deployment in critical scenarios. This is due to certain limitations, including *inconsistent* state estimates and *resource* (sensing and processing) constraints that are not appropriately addressed and often result in short mission duration. Therefore, to address these issues, our future research efforts will focus on: (i) investigation of the fundamental causes of 3D pose estimation inconsistency and approaches to mitigate these problems; and (ii) development of new state estimation algorithms to optimally allocate often limited system resources. An immediate impact of this research will be to advance the current state of the art, by improving the cost efficiency as well as the estimation consistency and accuracy.

- **Estimation and control in mobile sensor (robot) networks**

  Sensor networks – a typical cyber-physical system – are becoming increasingly popular, since they can measure and estimate quantities of interest at spatially distributed locations. Static sensor networks are useful for applications such as habitat monitoring [131] or terrain surveillance [2], while the additional mobility offers a sensor network an even broader spectrum of applications. Teams of mobile sensors (robots), for example, have been used for inspection of nuclear power plants [35], aerial surveillance [15], search

and rescue [152], and underwater or space exploration [23]. Although these application domains are quite dissimilar, a common requirement is to solve a statistical inference problem. For instance, we may seek to determine the poses of sensors (localization), estimate the trajectory of a moving target (tracking), build a map of the area where robots operate (mapping), or find the spatio-temporal parameters of a natural process (learning).

Within the expansive list of inference problems, two large categories are interesting: *model-based* estimation such as EKF, where an analytical model of the process being tracked is available, and *nonparametric* estimation such as PF, where such a model does not exist *a priori*. The latter has become increasingly popular, but its inconsistency is not sufficiently studied, which hence will be one of our future research efforts. In both classes, an important distinction can be made between *centralized* approaches, which assume the presence of a fusion center in the network where most (or all) of the processing takes place; and *decentralized* ones, where no such special node exists, and all the processing takes place in the sensor platforms. The latter is more appealing and will be within our research focus, since requiring a fusion center is not desirable in many cases due to reliability and/or scalability concerns. Furthermore, mobile sensor networks often need to operate under stringent constraints on their available sensing, processing, communication, and power resources. These, however, are not appropriately addressed by most existing inference algorithms and hence make them impracticable. For this reason, our future research efforts will also focus on: (i) performing optimal inference under realistic resource constraints; (ii) studying the effects of the key properties of the sensor system – such as the size of the sensor network, the type and precision of the sensors, the frequency of the observations, and the availability of communication and processing resources – on the attainable estimation accuracy; and (iii) learning optimal motion strategies for each robot in the team so as to aid estimation tasks, and vice versa. This research will promote robot (sensor) coordination for information acquisition, communication, and management, by providing adaptability to changing conditions and increasing the reliability of networks.

As we finish this work, we are excited to find robot systems standing at the brink of widespread field deployment in real-world applications. We hope that our work, which enables consistent autonomous navigation, serves as a solid first step towards this goal.

References

[1] V. Aidala and S. Hammel, "Utilization of modified polar coordinates for bearings-only tracking," *IEEE Transactions on Automatic Control*, vol. 28, no. 3, pp. 283–294, 1983.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Dec. 2002.

[3] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Dover Publications, 2005.

[4] J. Andrade-Cetto and A. Sanfeliu, "The effects of partial observability in SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, Apr. 18–22, 2004, pp. 394–402.

[5] ——, "The effects of partial observability when building fully correlated maps," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 771–777, Aug. 2005.

[6] J. Andrade-Cetto, T. Vidal-Calleja, and A. Sanfeliu, "Unscented transformation of vehicle states in SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, Apr. 2005, pp. 323–328.

[7] S. Arulampalam, M. Clark, and R. Vinter, "Performance of the shifted rayleigh filter in single-sensor bearings-only tracking," in *Proc. of the International Conference on Information Fusion*, Quebec, QC, Canada, Jul. 9–12, 2007.

[8] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[9] A. Bahr, M. R. Walter, and J. J. Leonard, "Consistent cooperative localization," in *Proc. of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12-17, 2009, pp. 3415–3422.

[10] T. Bailey, "Constrained initialisation for bearing-only SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, vol. 2, Taipei, Taiwan, Sep. n14–19, 2003, pp. 1966–1971.

[11] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[12] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 9–15, 2006, pp. 3562–3568.

[13] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm," in *Proc. of the IEEE International Conference on Robotics and Automation*, Orlando, FL, May 15–19, 2006, pp. 424–429.

[14] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. New York: Wiley, 2001.

[15] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306 – 1324, Jul. 2006.

[16] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[17] S. S. Blackman and R. F. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[18] W. L. Brogan, *Modern Control Theory*. Upper Saddle River, NJ: Prentice Hall, 1991.

[19] A. Broggi, M. Bertozzi, A. Fascioli, C. G. L. Bianco, and A. Piazzi, "Visual perception of obstacles and vehicles for platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 164–176, Sep. 2000.

[20] D. Brown, “A solution to the general problem of multiple station analytical stereo triangulation,” Patrick Air Force Base, Florida, RCA-MTP Data Reduction Technical Report, no. 43, 1958.

[21] ——, “Results in geodetic photogrammetry I: The precise determination of the location of Bermuda from photogrametria observations of flares ejected from Juno 11,” Patrick Air Force Base, Florida, RCA-MTP Data Reduction Technical Report, no. 54, 1959.

[22] ——, “The bundle adjustment - progress and prospects,” *International Archives of Photogrammetry*, vol. 21, no. 3, pp. 1–33, 1976.

[23] C. J. Cannell and D. J. Stilwell, “A comparison of two approaches for adaptive sampling of environmental processes using autonomous underwater vehicles,” in *Proc. of the MTS/IEEE OCEANS*, Washington, DC, Dec. 19–23, 2005, pp. 1514–1521.

[24] F. Capezio, F. Mastrogiovanni, A. Sgorbissa, and R. Zaccaria, “Robot-assisted surveillance in large environments,” *Journal of Computing and Information Technology*, vol. 17, no. 1, pp. 95–108, Mar. 2009.

[25] J. Casper and R. R. Murphy, “Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center,” *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, vol. 33, no. 2, pp. 367–385, Jun. 2003.

[26] J. A. Castellanos, R. Martinez-Cantin, J. Tardos, and J. Neira, “Robocentric map joining: Improving the consistency of EKF-SLAM,” *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 21–29, Jan. 2007.

[27] J. A. Castellanos, J. Neira, and J. Tardos, “Limits to the consistency of EKF-based SLAM,” in *Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, Jul. 5–7, 2004, pp. 1244–1249.

[28] V. Cevher, R. Velmurugan, and J. H. McClellan, “A range-only multiple target particle filter tracker,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, May 14–19, 2006.

[29] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway, "Real-time and robust monocular SLAM using predictive multi-resolution descriptors," in *Proc. of the International Symposium on Visual Computing*, Lake Tahoe, NV, Nov. 6–8, 2006, pp. 276–285.

[30] Z. Chen, K. Jiang, and J. Hung, "Local observability matrix and its application to observability analyses," in *Proc. of the 16th Annual Conference of IEEE*, Pacific Grove, CA, Nov. 27–30, 1990, pp. 100–103.

[31] J. M. C. Clark, R. B. Vinter, and M. M. Yaqoob, "Shifted Rayleigh filter: a new algorithm for bearings-only tracking," *IEEE Transactions Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1373–1384, Oct. 2007.

[32] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, New York, 1991.

[33] D. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*. Springer, 2005.

[34] R. Davidson and J. G. MacKinnon, *Estimation and Inference in Econometrics*. Oxford University Press, 1993.

[35] A. Davison and N. Kita, "Active visual localisation for cooperating inspection robots," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, Oct. 31–Nov. 5, 2000, pp. 1709–15.

[36] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006.

[37] Y. Dieudonne, O. Labbani-Igbida, and F. Petit, "Deterministic robot-network localization is hard," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 331–339, Apr. 2010.

[38] T. Dong-Si and A. I. Mourikis, "Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis," in *Proc. of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 5655–5662.

[39] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[40] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, Jul. 2000.

[41] A. Doucet and A. M. Johansen, *A tutorial on particle filtering and smoothing: fifteen years later*. University Press, 2009, ch. 8.2.

[42] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience, 2000.

[43] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[44] A. Edelman and H. Murakami, "Polynomial roots from companion matrix eigenvalues," *Mathematics of Computation*, vol. 64, no. 210, pp. 763–776, 1995.

[45] I. Z. Emiris and A. Rege, "Monomial bases and polynomial system solving," in *Proc. of the international symposium on Symbolic and algebraic computation*, Oxford, UK, Jul. 20–22, 1994, pp. 114–122.

[46] J. K. Erickson, "Living the dream - an overview of the Mars exploration project," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 12–18, Jun. 2006.

[47] A. Farina, "Target tracking with bearing-only measurements," *Signal Processing*, vol. 78, pp. 61–78, Oct. 1999.

[48] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, Jun. 2000.

[49] G. H. Golub and C. F. V. Loan, *Matrix Computations*. The Johns Hopkins University Press, 1996.

[50] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F: Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, Apr. 1993.

[51] D. Goshen-Meskin and I. Bar-Itzhack, "Observability analysis of piece-wise constant systems - Part I: Theory," *IEEE Transactions on Aerospace and Electronics Systems*, vol. 28, no. 4, pp. 1056–1067, Oct. 1992.

[52] S. I. Granshaw, "Bundle adjustment methods in engineering photogrammetry," *The Photogrammetric Record*, vol. 10, no. 56, pp. 181–207, Oct. 1980.

[53] G. Grisetti, R. Kmmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[54] G. Grisetti, R. Kmmerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2d and 3d mapping," in *IEEE International Conference on Robotics and Automation*, Anchorage, AK, May3–8, 2010, pp. 273–278.

[55] G. Grisetti, C. Stachniss, and W. Burgard, "Nonlinear constraint network optimization for efficient map learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 428–439, Sep. 2009.

[56] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map building algorithm for real time implementation," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, Jun. 2001.

[57] F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, Feb. 2002.

[58] F. Gustafsson, "Particle filter theory and practice with positioning applications," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 25, no. 7, pp. 53 –82, Jul. 2010.

[59] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[60] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Transactions on Automatic Control*, vol. 22, no. 5, pp. 728 – 740, Oct. 1977.

[61] J. A. Hesch and S. I. Roumeliotis, "Design and analysis of a portable indoor localization aid for the visually impaired," *International Journal of Robotics Research*, vol. 29, no. 11, pp. 1400–1415, Sep. 2010.

[62] S. Holmes, G. Klein, and D. W. Murray, "A square root unscented kalman filter for visual monoSLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, May 12–17, 2008, pp. 3710–3716.

[63] S. A. Holmes, G. Klein, and D. W. Murray, "An $O(N^2)$ square root unscented kalman filter for visual simultaneous localization and mapping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 1251–1263, Jul. 2009.

[64] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and System*, Lausanne,Switzerland, Sep. 30–Oct. 4, 2002, pp. 434–439.

[65] ——, "Putting the 'i' in 'team': an ego-centric approach to cooperative localization," in *Proc. of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sep. 14-19, 2003, pp. 868–874.

[66] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A first-estimates jacobian EKF for improving SLAM consistency," in *Proc. of the 11th International Symposium on Experimental Robotics*, Athens, Greece, Jul. 14–17, 2008.

[67] ——, "An observability constrained sliding window filter for SLAM," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, Sep. 25–30, 2011, pp. 65–72.

[68] ——, "A quadratic-complexity observability-constrained unscented Kalman filter for SLAM," *IEEE Transactions on Robotics*, 2012, (revised).

[69] G. P. Huang and S. I. Roumeliotis, "Analyticlly-guided-sampling particle filter applied to range-only target tracking," in *Proc. of the IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 6–10, 2013, (to appear).

[70] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter-based SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, May 19–23, 2008, pp. 473–479.

[71] ——, "On the complexity and consistency of UKF-based SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12–17, 2009, pp. 4401–4408.

[72] ——, "Observability-based rules for designing consistent EKF SLAM estimators," *International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, Apr. 2010.

[73] G. P. Huang and S. I. Roumeliotis, "On filter consistency of discrete-time nonlinear systems with partial-state measurements," in *Proc. of the American Control Conference*, Washington, DC, Jun. 17–19, 2013, (to appear).

[74] G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis, "On the consistency of multi-robot cooperative localization," in *Proc. of the Robotics: Science and Systems*, Seattle, WA, Jun. 28–Jul. 1, 2009, pp. 65–72.

[75] ——, "Observability-based consistent EKF estimators for multi-robot cooperative localization," *Autonomous Robots*, vol. 30, no. 1, pp. 99–122, Jan. 2011.

[76] G. P. Huang, K. X. Zhou, N. Trawny, and S. I. Roumeliotis, "A bank of MAP estimators for single-sensor range-only target tracking," in *Proc. of the American Control Conference*, Baltimore, MD, Jun. 30–Jul. 2, 2010, pp. 6974–6980.

[77] ——, "Bearing-only tracking using a bank of MAP estimators," in *Proc. of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 4998–5005.

[78] S. Huang and G. Dissanayake, "Convergence analysis for extended Kalman filter based SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Orlando, FL, May 15–19, 2006, pp. 412–417.

[79] ——, "Convergence and consistency analysis for extended Kalman filter based SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1036–1049, Oct. 2007.

[80] T. L. Huntsberger, A. Trebi-Ollennu, H. Aghazarian, P. S. Schenker, and P. Pirjanian, "Distributed control of multi-robot systems engaged in tightly coupled tasks," *Autonomous Robots*, vol. 17, no. 1, pp. 79–92, Jul. 2004.

[81] Y. Jabri, *The Mountain Pass Theorem: Variants, Generalizations and Some Applications*. Cambridge University Press, 2003.

[82] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[83] J. Jennings, G. Whelan, and W. Evans, "Cooperative search and rescue with a team of mobile robots," in *Proc. of the International Conference on Advanced Robotics*, New Orleans, LA, Jul. 7–9, 1997, pp. 193–200.

[84] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means andcovariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[85] S. Julier and J. K. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in *Proc. of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 21–26, 2001, pp. 4238–4243.

[86] M. Kaess, V. Ila, R. Roberts, and F. Dellaert, "The Bayes tree: An algorithmic foundation for probabilistic robot mapping," in *Proc. of the International Workshop on the Algorithmic Foundations of Robotics*, Singapore, Dec. 2010, pp. 157–173.

[87] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *International Journal of Robotics Research*, vol. 31, pp. 217–236, Feb. 2012.

[88] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[89] M. Kaess, H. Johannsson, D. Rosen, and J. Leonard, "iSAM: Incremental smoothing and mapping," https://svn.csail.mit.edu/isam, accessed Aug. 31, 2012.

[90] N. Karam, F. Chausse, R. Aufrere, and R. Chapuis, "Localization of a group of communicating vehicles by state exchange," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 9-15, 2006, pp. 519–524.

[91] S. Kay, *Fundamentals of Statistical Signal Processing, Vol. I - Estimation Theory*. Prentice Hall, 1993.

[92] J. H. Kim and S. Sukkarieh, "Airborne simultaneous localisation and map building," in *Proc. of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sep. 14–19, 2003, pp. 406–411.

[93] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nara, Japan, Nov. 13–16, 2007.

[94] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1066 –1077, Oct. 2008.

[95] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2D mapping," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 18–22, 2010, pp. 22–29.

[96] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based maps," *International Journal of Robotics Research*, vol. 29, no. 8, pp. 941–957, jul 2010.

[97] J. H. Kotecha and P. M. Djuric, "Gaussian particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, oct 2003.

[98] ——, "Gaussian sum particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2602–2612, oct 2003.

[99] T. Kronhamn, "Bearings-only target motion analysis based on a multihypothesis Kalman filter and adaptive ownship motion control," *IEE Proc. - Radar, Sonar and Navigation*, vol. 145, pp. 247–252, Aug. 1998.

[100] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 3607–3613.

[101] R. Kurazume and S. Hirose, "An experimental study of a cooperative positioning system," *Autonomous Robots*, vol. 8, no. 1, pp. 43–52, Jan. 2000.

[102] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *Proc. of the IEEE International Conference on Robotics and Automation*, Los Alamitos, CA, May 8–13, 1994, pp. 1250 – 1257.

[103] J. W. Langelaan, "State estimation for autonomous flight in cluttered environments," Ph.D. dissertation, Stanford University, Department of Aeronautics and Astronautics, 2006.

[104] J. Langelaan and S. Rock, "Passive GPS-free navigation for small UAVs," in *Proc. of the IEEE Aerospace Conference*, Big Sky, MT, Mar. 5–12, 2005, pp. 1–9.

[105] K. Lee, W. Wijesoma, and J. Guzman, "On the observability and observability analysis of SLAM," in *Proc. of the IEEE/RSJ Intenational Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 9–15, 2006, pp. 3569–3574.

[106] T. Lefebvre, H. Bruyninckx, and J. De Schuller, "Comment on "a new method for the nonlinear transformation of means and covariances in filters and estimators" [and authors' reply]," *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1406–1409, Aug. 2002.

[107] J. J. Leonard, R. J. Rikoski, P. M. Newman, and M. Bosse, "Mapping partially observable features from multiple uncertain vantage points," *International Journal of Robotics Research*, vol. 21, pp. 943–975, Oct. 2002.

[108] A. Martinelli, "Improving the precision on multi robot localization by using a series of filters hierarchically distributed," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, Oct. 29–Nov. 2, 2007, pp. 1053–1058.

[109] A. Martinelli and R. Siegwart, "Observability analysis for mobile robot localization," in *Proc. of the IEEE/RSJ Intenational Conference on Intelligent Robots and Systems*, Edmonton, Canada, Aug. 2005, pp. 1471– 1476.

[110] R. Martinez-Cantin and J. A. Castellanos, "Unscented SLAM for large-scale outdoor environments," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug. 2–6, 2005, pp. 3427–3432.

[111] P. S. Maybeck, *Stochastic Models, Estimation and Control*, ser. Mathematics in Science and Engineering. London: Academic Press, 1982, vol. 141-2.

[112] ——, *Stochastic Models, Estimation, and Control*, ser. Mathematics in Science and Engineering. London: Academic Press, 1979, vol. 141-1.

[113] C. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2001.

[114] J. W. Milnor, *Topology from the Differentiable Viewpoint*. Princeton University Press, 1997.

[115] M. Montemerlo, "FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Jul. 2003.

[116] A. I. Mourikis and S. I. Roumeliotis, "Analytical characterization of the accuracy of SLAM without absolute orientation measurements," in *Proc. of the Robotics: Science and Systems*, Philadelphia, PA, Aug. 16–19, 2006, pp. 215–222.

[117] ——, "Performance analysis of multirobot cooperative localization," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 666–681, Aug. 2006.

[118] ——, "Predicting the accuracy of Cooperative Simultaneous Localization and Mapping (C-SLAM)," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1273–1286, Dec. 2006.

[119] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, D. M. Helmick, and L. Matthies, "Autonomous stair climbing for tracked vehicles," *International Journal of Computer Vision & International Journal of Robotics Research - Joint Special Issue on Vision and Robotics*, vol. 26, no. 7, pp. 737–758, Jul. 2007.

[120] A. Mourikis, N. Trawny, S. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *Robotics, IEEE Transactions on*, vol. 25, no. 2, pp. 264 –280, Apr. 2009.

[121] P. Moutarlier and R. Chatila, "Stochastic multisensory data fusion for mobile robot location and environment modeling," in *Proc. of the Fifth International Symposium of*

Robotics Research, H. Miura and S. Arimoto, Eds., Tokyo, Japan, Aug. 28–31, 1989, pp. 85–94.

[122] D. Mušicki, "Bearings only single-sensor target tracking using gaussian mixtures," *Automatica*, vol. 45, no. 9, pp. 2088–2092, Sep. 2009.

[123] J. Neira and J. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890 – 897, 2001.

[124] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization," in *Proc. of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12–17, 2009, pp. 1402–1409.

[125] P. Newman, "On the structure and solution of the simultaneous localisation and map building problem," Ph.D. dissertation, University of Sydney, Mar. 1999.

[126] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, Jan. 2006.

[127] S. Panzieri, F. Pascucci, and R. Setola, "Multirobot localization using interlaced extended Kalman filter," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 9-15, 2006, pp. 2816–2821.

[128] C. Parker, H. Zhang, and C. R. Kube, "Blind buldozing: Multiple robot nest construction," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, Oct. 27-31 2003, pp. 2010–15.

[129] L. E. Parker, B. Birch, and C. Reardon, "Indoor target intercept using an acoustic sensor network and dual wavefront path planning," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, Oct. 27–31, 2003, pp. 278 – 283.

[130] M. Paskin, "Thin junction tree filters for simultaneous localization and mapping," Ph.D. dissertation, University of California, Berkeley, 2002.

[131] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson, "Analysis of wireless sensor networks for habitat monitoring," in *Wireless sensor networks*, C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Eds. Norwell, MA, USA: Kluwer Academic Publishers, 2004, pp. 399–423.

[132] A. Ranganathan, M. Kaess, and F. Dellaert, "Fast 3d pose estimation with out-of-sequence measurements," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 29– Nov.2, 2007, pp. 2486–2493.

[133] I. M. Rekleitis, G. Dudek, and E. E. Milios, "Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Switzerland, Sep. 30–Oct. 4, 2002, pp. 2690–2695.

[134] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[135] B. Ristic, S. Arulampalam, and J. McCarthy, "Target motion analysis using range-only measurements: algorithms, performance and application to ISAR data," *Signal Processing*, vol. 82, no. 2, pp. 273–296, Feb. 2002.

[136] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, Oct. 2002.

[137] S. I. Roumeliotis and I. M. Rekleitis, "Propagation of uncertainty in cooperative multirobot localization: Analysis and experimental results," *Autonomous Robots*, vol. 17, no. 1, pp. 41–54, Jul. 2004.

[138] W. Rudin, *Principles of Mathematical Analysis*. McGraw-Hill, 1986.

[139] Y. Rui and Y. Chen, "Better proposal distributions: object tracking using unscented particle filter," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2001, pp. 786–793.

[140] S. Se, D. G. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *International Journal of Robotics Research*, vol. 21, no. 8, pp. 735–758, Aug. 2002.

[141] B. Sellner, F. W. Heger, L. M. Hiatt, R. Simmons, and S. Singh, "Coordinated multiagent teams and sliding autonomy for large-scale assembly," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1425–1444, Jul. 2006.

[142] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, Sept./Oct. 2010.

[143] G. Sibley, G. S. Sukhatme, and L. Matthies, "Constant time sliding window filter SLAM as a basis for metric visual perception," in *Proc. IEEE International Conference on Robotics and Automation Workshop*, Roma, Italy, Apr. 10–14, 2007.

[144] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, Dec. 1986.

[145] R. Smith, M. Self, and P. Cheeseman, *Estimating Uncertain Spatial Relationships in Robotics*. Springer-Verlag New York, 1990, pp. 167–193.

[146] T. L. Song, "Observability of target tracking with bearings-only measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1468–1472, Oct. 1996.

[147] ——, "Observability of target tracking with range-only measurements," *IEEE Journal of Oceanic Engineering*, vol. 24, no. 3, pp. 383–387, Jul. 1999.

[148] H. Stark and J. W. Woods, *Probability and Random Processes with Applications to Signal Processing*. Prentice Hall, 2001.

[149] H. Strasdat, J. Montiel, and A. Davison, "Real-time monocular SLAM: Why filter?" in *IEEE International Conference on Robotics and Automation*, Anchorage, AK, May3–8, 2010, pp. 2657–2664.

[150] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, Feb. 2012.

[151] M. Struwe, *Variational Methods: Applications to Nonlinear Partial Differential Equations and Hamiltonian Systems*. Springer, 2008.

[152] H. Sugiyama, T. Tsujioka, and M. Murata, "Collaborative movement of rescue robots for reliable and effective networking in disaster area," in *Proc. of the International Conference on Collaborative Computing: Networking, Applications and Worksharing*, San Jose, CA, Dec. 19–21, 2005.

[153] R. Szeliski and S. B. Kang, "Recovering 3d shape and motion from image streams using nonlinear least squares," *Journal of Visual Communication and Image Representation*, vol. 5, no. 1, pp. 10–28, 1994.

[154] A. D. Tews, G. S. Sukhatme, and M. J. Mataric, "A multi-robot approach to stealthy navigation in the presence of an observer," in *Proc. of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, Apr. 26–May 1, 2004, pp. 2379–2385.

[155] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, MA: The MIT Press, 2005.

[156] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley, the robot that won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, Jun. 2006.

[157] S. Thrun, "Robotic mapping: A survey," in *Exploring artificial intelligence in the new millennium*, G. Lakemeyer and B. Nebel, Eds. San Francisco, CA: Morgan Kaufmann Inc., 2003, pp. 1–35.

[158] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*. IEE, 2005.

[159] N. Trawny, A. I. Mourikis, S. I. Roumeliotis, A. E. Johnson, and J. Montgomery, "Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks," *Journal of Field Robotics*, vol. 24, no. 5, pp. 357–378, May 2007.

[160] N. Trawny and S. I. Roumeliotis, "On the global optimum of planar, range-based robot-to-robot relative pose estimation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Anchorage, AK, May3–8, 2010, pp. 3200–3206.

[161] N. Trawny, X. S. Zhou, K. Zhou, and S. I. Roumeliotis, "Interrobot transformations in 3-D," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 226 –243, Apr. 2010.

[162] N. Trawny, X. S. Zhou, and S. I. Roumeliotis, "3D relative pose estimation from six distances," in *Proc. of the Robotics: Science and Systems*, Seattle, WA, Jun. 28– Jul. 1, 2009.

[163] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment – a modern synthesis," *Lecture Notes in Computer Science*, vol. 1883, pp. 298–375, Jan. 2000.

[164] I. Ulrich and J. Borenstein, "The GuideCane–applying mobile robot technologies to assist the visually impaired," *IEEE Transactions on Systems, Man, and Cybernetics-Part A*, vol. 31, no. 2, pp. 131–136, Mar. 2001.

[165] F. van der Heijden, "Consistency checks for particle filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 140–145, Jan. 2006.

[166] R. van der Merwe, A. Doucet, N. De Freitas, and E. Wan, "The unscented particle filter," in *Proc. of the Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, Dec. 3–8, 2001, pp. 584–590.

[167] T. Vidal-Calleja, M. Bryson, S. Sukkarieh, A. Sanfeliu, and J. Andrade-Cetto, "On the observability of bearing only SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Roma, Italy, Apr. 10–14, 2007, pp. 4114–4118.

[168] F. Wang, Y. Lin, T. Zhang, and J. Liu, "Particle filter with hybrid proposal distribution for nonlinear state estimation," *Journal of Computers*, vol. 6, no. 11, pp. 2491–2501, Nov. 2011.

[169] S. B. Williams, P. Newman, M. Dissanayake, and H. Durrant-Whyte, "Autonomous underwater simultaneous localisation and map building," in *Proc. of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, Apr. 24–28, 2000, pp. 1793–1798.

[170] K. X. Zhou and S. I. Roumeliotis, "Multi-robot active target tracking with distance and bearing observations," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, Oct. 11–15 2009, pp. 2209–2216.

[171] K. Zhou and S. Roumeliotis, "Optimal motion strategies for range-only constrained multisensor target tracking," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1168 –1185, Oct. 2008.

[172] ——, "Multirobot active target tracking with combinations of relative observations," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 678 –695, Aug. 2011.

[173] X. S. Zhou and S. I. Roumeliotis, "Robot-to-robot relative pose estimation from range measurements," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1379–1393, Dec. 2008.

Appendix A

Appendices for Chapter 2

A.1 Unicycle model

If the unicycle model is used, and we employ the approximation that the velocity and heading are constant during each propagation interval, we obtain ${}^{R_k}\hat{\mathbf{x}}_{R_{k+1}} = [v_{m_k}\delta t \;\; 0 \;\; \omega_{m_k}\delta t]^T$, where $\mathbf{u}_{m_k} = [v_{m_k} \;\; \omega_{m_k}]^T$ are the linear and rotational velocity measurements, respectively, and δt is the sampling period. Substitution in (2.2)-(2.3) yields the familiar robot pose propagation equations:

$$\hat{\mathbf{p}}_{R_{k+1|k}} = \hat{\mathbf{p}}_{R_{k|k}} + \begin{bmatrix} v_{m_k}\delta t c(\hat{\phi}_{R_{k|k}}) \\ v_{m_k}\delta t s(\hat{\phi}_{R_{k|k}}) \end{bmatrix} \tag{A.1}$$

$$\hat{\phi}_{R_{k+1|k}} = \hat{\phi}_{R_{k|k}} + \omega_{m_k}\delta t \tag{A.2}$$

Similarly, the commonly used expressions for the Jacobian matrices $\mathbf{\Phi}_{R_k}$ and $\mathbf{G}_{R_k}$ can be derived from (2.6), (2.8) and (2.9). Specifically, by substituting the robot displacement ${}^{R_k}\hat{\mathbf{p}}_{R_{k+1}} = \begin{bmatrix} v_{m_k}\delta t & 0 \end{bmatrix}^T$ into (2.8), we have:

$$\mathbf{\Phi}_{R_k} = \begin{bmatrix} 1 & 0 & -v_{m_k}\delta t s(\hat{\phi}_{R_{k|k}}) \\ 0 & 1 & v_{m_k}\delta t c(\hat{\phi}_{R_{k|k}}) \\ 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

To derive the Jacobian matrix $\mathbf{G}^o_{R_k}$ with respect to the odometry vector $\mathbf{u}_k$, instead of ${}^{R_k}\hat{\mathbf{x}}_{R_{k+1}}$, we apply the chain rule of differentiation as follows:

$$\mathbf{G}^o_{R_k} = \frac{\partial(\mathbf{x}_{R_{k+1}})}{\partial({}^{R_k}\mathbf{x}_{R_{k+1}})}\Big|_{{}^{R_k}\hat{\mathbf{x}}_{R_{k+1}}} \times \frac{\partial({}^{R_k}\mathbf{x}_{R_{k+1}})}{\partial \mathbf{u}_k}\Big|_{\mathbf{u}_{m_k}} \tag{A.4}$$

The first term is the Jacobian with respect to the robot pose change (displacement and orientation change), evaluated at the estimate ${}^{R_k}\hat{\mathbf{x}}_{R_{k+1}}$, and is given in (2.9). The second term is the Jacobian of the robot pose change with respect to $\mathbf{u}_k$. Since ${}^{R_k}\mathbf{x}_{R_{k+1}} = \begin{bmatrix} v_k\delta t & 0 & \omega_k\delta t \end{bmatrix}^T$, this Jacobian is simply given by:

$$\frac{\partial({}^{R_k}\mathbf{x}_{R_{k+1}})}{\partial \mathbf{u}_k}\Big|_{\mathbf{u}_{m_k}} = \begin{bmatrix} \delta t & 0 \\ 0 & 0 \\ 0 & \delta t \end{bmatrix} \tag{A.5}$$

Therefore, substitution of (A.5) and (2.9) into (A.4) yields:

$$\mathbf{G}^o_{R_k} = \begin{bmatrix} \delta t c(\hat{\phi}_{R_{k|k}}) & 0 \\ \delta t s(\hat{\phi}_{R_{k|k}}) & 0 \\ 0 & \delta t \end{bmatrix} \tag{A.6}$$

We thus showed how the commonly used expressions for (2.2)-(2.4), as well as the state and noise Jacobians can be derived.

A.2 Proof of Lemma 2.4.1

The proof is based on mathematical induction, by verifying the structure of the kth order Lie derivatives. We define the Lie derivative of a C^∞ function h on an open subset $\mathcal{S} \subset \mathbb{R}^{dim(\mathbf{x})}$ along an analytic vector field $\mathbf{f}$ on $\mathcal{S}$, as:

$$L_{\mathbf{f}}h = (dh)\mathbf{f} \tag{A.7}$$

where dh is the gradient of h with respect to the state vector $\mathbf{x}$. We start by noting the following identities, which will be useful in the ensuing derivations.

$$\begin{aligned}\frac{d\rho}{d\mathbf{x}} &= \begin{bmatrix} -\frac{\delta x}{\rho} & -\frac{\delta y}{\rho} & 0 & \frac{\delta x}{\rho} & \frac{\delta y}{\rho} \end{bmatrix} \\ &= \begin{bmatrix} -c\theta & -s\theta & 0 & c\theta & s\theta \end{bmatrix} \qquad \text{(A.8)} \\ \frac{d\psi}{d\mathbf{x}} &= \frac{1}{\rho}\begin{bmatrix} \frac{\delta y}{\rho} & -\frac{\delta x}{\rho} & -\rho & -\frac{\delta y}{\rho} & \frac{\delta x}{\rho} \end{bmatrix} \\ &= \frac{1}{\rho}\begin{bmatrix} s\theta & -c\theta & -\rho & -s\theta & c\theta \end{bmatrix} \qquad \text{(A.9)}\end{aligned}$$

where $\delta x \triangleq x_L - x_R$, $\delta y \triangleq y_L - y_R$, and $\theta \triangleq \psi + \phi_R$.

We first prove that if h has the special structure shown in (2.15), then the zeroth- and first-order Lie derivatives are functions of ρ and ψ only.

By applying the chain rule of differentiation, the zeroth-order (i.e., $k = 0$) Lie derivative is computed as follows:

$$L^0 h \triangleq \frac{dh}{d\mathbf{x}} = \underbrace{\begin{bmatrix} \frac{\partial h}{\partial \rho} & \frac{\partial h}{\partial \psi} \end{bmatrix}}_{\mathbf{A}_0} \begin{bmatrix} \frac{d\rho}{d\mathbf{x}} \\ \frac{d\psi}{d\mathbf{x}} \end{bmatrix} \qquad \text{(A.10)}$$

It is important to note that since h is a function of ρ and ψ only, the terms $\frac{\partial h}{\partial \rho}$ and $\frac{\partial h}{\partial \psi}$ are also functions of ρ and ψ only. As a result, the matrix $\mathbf{A}_0$ is a function of ρ and ψ, whose exact structure depends on the particular measurement function h.

The first-order (i.e., $k = 1$) Lie derivatives are calculated according to the definition (A.7), and employing the results of (A.8) and (A.9), as:

$$\begin{aligned} L^1_{\mathbf{f_1}} h &= \begin{bmatrix} \frac{\partial h}{\partial \rho} & \frac{\partial h}{\partial \psi} \end{bmatrix} \begin{bmatrix} \frac{d\rho}{d\mathbf{x}} \\ \frac{d\psi}{d\mathbf{x}} \end{bmatrix} \mathbf{f}_1 \\ &= \mathbf{A}_0 \begin{bmatrix} -c\theta c\phi_R - s\theta s\phi_R \\ \frac{1}{\rho}(s\theta c\phi_R - c\theta s\phi_R) \end{bmatrix} = \mathbf{A}_0 \begin{bmatrix} -c\psi \\ \frac{s\psi}{\rho} \end{bmatrix} \qquad \text{(A.11)} \\ L^1_{\mathbf{f_2}} h &= \begin{bmatrix} \frac{\partial h}{\partial \rho} & \frac{\partial h}{\partial \psi} \end{bmatrix} \begin{bmatrix} \frac{d\rho}{d\mathbf{x}} \\ \frac{d\psi}{d\mathbf{x}} \end{bmatrix} \mathbf{f}_2 = \mathbf{A}_0 \begin{bmatrix} 0 \\ -1 \end{bmatrix} \qquad \text{(A.12)}\end{aligned}$$

We thus see that both the zeroth- and the first-order Lie derivatives are functions of ρ and ψ only. This is the base case for the proof by induction.

Now assume the k-th order Lie derivatives $L^k_{\mathbf{f}_i} h$, $i = 1, 2$, are functions of ρ and ψ only.[1]

[1] Extension of this analysis to the case of mixed k-th order Lie derivatives is straightforward, though more involved in terms of notation; thus, it is omitted to preserve presentation clarity.

Then their gradients can be computed by:

$$\frac{d(L_{\mathbf{f}_i}^k h)}{d\mathbf{x}} = \underbrace{\begin{bmatrix} \frac{\partial}{\partial \rho}(L_{\mathbf{f}_i}^k h) & \frac{\partial}{\partial \psi}(L_{\mathbf{f}_i}^k h) \end{bmatrix}}_{\mathbf{A}_{k_i}} \begin{bmatrix} \frac{d\rho}{d\mathbf{x}} \\ \frac{d\psi}{d\mathbf{x}} \end{bmatrix} \tag{A.13}$$

where $\mathbf{A}_{k_i}$ is a function of ρ and ψ only. Thus, the $(k+1)$-th order Lie derivatives are computed as follows:

$$\begin{aligned} L_{\mathbf{f}_1}^{k+1} h &= \begin{bmatrix} \frac{\partial}{\partial \rho}(L_{\mathbf{f}_1}^k h) & \frac{\partial}{\partial \psi}(L_{\mathbf{f}_1}^k h) \end{bmatrix} \begin{bmatrix} \frac{d\rho}{d\mathbf{x}} \\ \frac{d\psi}{d\mathbf{x}} \end{bmatrix} \mathbf{f}_1 \\ &= \mathbf{A}_{k_1} \begin{bmatrix} -c\theta c\phi_R - s\theta s\phi_R \\ \frac{1}{\rho}(s\theta c\phi_R - c\theta s\phi_R) \end{bmatrix} = \mathbf{A}_{k_1} \begin{bmatrix} -c\psi \\ \frac{s\psi}{\rho} \end{bmatrix} \end{aligned} \tag{A.14}$$

$$L_{\mathbf{f}_2}^{k+1} h = \begin{bmatrix} \frac{\partial}{\partial \rho}(L_{\mathbf{f}_2}^k h) & \frac{\partial}{\partial \psi}(L_{\mathbf{f}_2}^k h) \end{bmatrix} \begin{bmatrix} \frac{d\rho}{d\mathbf{x}} \\ \frac{d\psi}{d\mathbf{x}} \end{bmatrix} \mathbf{f}_2 = \mathbf{A}_{k_2} \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tag{A.15}$$

Clearly, the $(k+1)$-th order Lie derivatives are also functions of ρ and ψ only, and the proof by induction is complete.

A.3 Proof of Lemma 2.4.2

Employing the expressions for the Lie derivatives derived in Appendix A.2, we have:

$$
\begin{aligned}
d\mathcal{G} &= \operatorname{span}\left\{ \begin{array}{llll}
\frac{d(L^0 h_1)}{d\mathbf{x}}, & & \ldots, & \frac{d(L^0 h_n)}{d\mathbf{x}}, \\
\frac{d(L^1_{\mathbf{f}_1} h_1)}{d\mathbf{x}}, & \frac{d(L^1_{\mathbf{f}_2} h_1)}{d\mathbf{x}}, & \ldots, & \frac{d(L^1_{\mathbf{f}_1} h_n)}{d\mathbf{x}}, \frac{d(L^1_{\mathbf{f}_2} h_n)}{d\mathbf{x}}, \\
\ldots & & \ldots & \ldots \\
\frac{d(L^k_{\mathbf{f}_1} h_1)}{d\mathbf{x}}, & \frac{d(L^k_{\mathbf{f}_2} h_1)}{d\mathbf{x}}, & \ldots, & \frac{d(L^k_{\mathbf{f}_1} h_n)}{d\mathbf{x}}, \frac{d(L^k_{\mathbf{f}_2} h_n)}{d\mathbf{x}}
\end{array} \right\} \\
&= \operatorname{span}\left\{ \begin{array}{llll}
\mathbf{A}_0^1 \begin{bmatrix} \left(\frac{d\rho}{d\mathbf{x}}\right) \\ \left(\frac{d\psi}{d\mathbf{x}}\right) \end{bmatrix}, & & \ldots, & \mathbf{A}_0^n \begin{bmatrix} \left(\frac{d\rho}{d\mathbf{x}}\right) \\ \left(\frac{d\psi}{d\mathbf{x}}\right) \end{bmatrix}, \\
\mathbf{A}_{1_1}^1 \begin{bmatrix} \left(\frac{d\rho}{d\mathbf{x}}\right) \\ \left(\frac{d\psi}{d\mathbf{x}}\right) \end{bmatrix}, \mathbf{A}_{1_2}^1 \begin{bmatrix} \left(\frac{d\rho}{d\mathbf{x}}\right) \\ \left(\frac{d\psi}{d\mathbf{x}}\right) \end{bmatrix}, & & \ldots, & \mathbf{A}_{1_1}^n \begin{bmatrix} \left(\frac{d\rho}{d\mathbf{x}}\right) \\ \left(\frac{d\psi}{d\mathbf{x}}\right) \end{bmatrix}, \mathbf{A}_{1_2}^n \begin{bmatrix} \left(\frac{d\rho}{d\mathbf{x}}\right) \\ \left(\frac{d\psi}{d\mathbf{x}}\right) \end{bmatrix}, \\
\ldots & & \ldots & \ldots \\
\mathbf{A}_{k_1}^1 \begin{bmatrix} \left(\frac{d\rho}{d\mathbf{x}}\right) \\ \left(\frac{d\psi}{d\mathbf{x}}\right) \end{bmatrix}, \mathbf{A}_{k_2}^1 \begin{bmatrix} \left(\frac{d\rho}{d\mathbf{x}}\right) \\ \left(\frac{d\psi}{d\mathbf{x}}\right) \end{bmatrix}, & & \ldots, & \mathbf{A}_{k_1}^n \begin{bmatrix} \left(\frac{d\rho}{d\mathbf{x}}\right) \\ \left(\frac{d\psi}{d\mathbf{x}}\right) \end{bmatrix}, \mathbf{A}_{k_2}^n \begin{bmatrix} \left(\frac{d\rho}{d\mathbf{x}}\right) \\ \left(\frac{d\psi}{d\mathbf{x}}\right) \end{bmatrix}
\end{array} \right\} \\
&= \operatorname{span}\left\{ \underbrace{\begin{bmatrix} \mathbf{A}_0^1 \\ \vdots \\ \mathbf{A}_0^n \\ \mathbf{A}_{1_1}^1 \\ \mathbf{A}_{1_2}^1 \\ \vdots \\ \mathbf{A}_{1_1}^n \\ \mathbf{A}_{1_2}^n \\ \vdots \\ \mathbf{A}_{k_1}^1 \\ \mathbf{A}_{k_2}^1 \\ \vdots \\ \mathbf{A}_{k_1}^n \\ \mathbf{A}_{k_2}^n \end{bmatrix}}_{\mathbf{A}_K} \underbrace{\begin{bmatrix} \left(\frac{d\rho}{d\mathbf{x}}\right) \\ \left(\frac{d\psi}{d\mathbf{x}}\right) \end{bmatrix}}_{\mathbf{M}} \right\} =: \operatorname{span}\left(\mathbf{A}_K \mathbf{M}\right)
\end{aligned}
\tag{A.16}
$$

where the superscript i in $\mathbf{A}^i$ refers to the measurement function h_i (i.e., $i = 1, 2, ..., n$). We will now show that $\mathbf{A}_K$ *is of full column rank for any type of measurements (i.e., distance-and-bearing, distance-only, and bearing-only measurements).*

Distance-and-bearing measurements We first consider the distance-and-bearing measurements which are given by:

$$h_1(\mathbf{x}) = \rho = ||\mathbf{p}_L - \mathbf{p}_R|| \tag{A.17}$$

$$h_2(\mathbf{x}) = \psi = \text{atan2}(y_L - y_R, x_L - x_R) - \phi_R \tag{A.18}$$

In this case, we have

$$\mathbf{A}_0^1 = \begin{bmatrix} \frac{\partial h_1}{\partial \rho} & \frac{\partial h_1}{\partial \psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{A.19}$$

$$\mathbf{A}_0^2 = \begin{bmatrix} \frac{\partial h_2}{\partial \rho} & \frac{\partial h_2}{\partial \psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{A.20}$$

Hence, $\mathbf{A}_K = \begin{bmatrix} \mathbf{A}_0^1 \\ \mathbf{A}_0^2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{I}_2 \\ \vdots \end{bmatrix}$ is full-rank.

Distance-only measurements We now consider the case of distance-only measurement which is given by:

$$h(\mathbf{x}) = \rho = ||\mathbf{p}_L - \mathbf{p}_R|| \tag{A.21}$$

In this case, we have

$$\mathbf{A}_0 = \begin{bmatrix} \frac{\partial h}{\partial \rho} & \frac{\partial h}{\partial \psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{A.22}$$

The first-order Lie derivatives are computed as:

$$L_{\mathbf{f}_1}^1 h = \mathbf{A}_0 \begin{bmatrix} -c\psi \\ \frac{s\psi}{\rho} \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} -c\psi \\ \frac{s\psi}{\rho} \end{bmatrix} = -c\psi \tag{A.23}$$

$$L_{\mathbf{f}_2}^1 h = \mathbf{A}_0 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \tag{A.24}$$

Then, we compute $\mathbf{A}_{1_1}$ and $\mathbf{A}_{1_2}$ as:

$$\mathbf{A}_{1_1} = \begin{bmatrix} \frac{\partial}{\partial \rho}(L^1_{\mathbf{f}_1} h) & \frac{\partial}{\partial \psi}(L^1_{\mathbf{f}_1} h) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \rho}(-c\psi) & \frac{\partial}{\partial \psi}(-c\psi) \end{bmatrix} = \begin{bmatrix} 0 & s\psi \end{bmatrix} \tag{A.25}$$

$$\mathbf{A}_{1_2} = \begin{bmatrix} \frac{\partial}{\partial \rho}(L^1_{\mathbf{f}_2} h) & \frac{\partial}{\partial \psi}(L^1_{\mathbf{f}_2} h) \end{bmatrix} = \mathbf{0} \tag{A.26}$$

Now we have $\mathbf{A}_K = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_{1_1} \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & s\psi \\ \vdots & \vdots \end{bmatrix}$ which, in general, is full-rank.

Bearing-only measurements We finally consider the bearing-only measurement which is given by:

$$h(\mathbf{x}) = \psi = \text{atan2}(y_L - y_R, x_L - x_R) - \phi_R \tag{A.27}$$

In this case, we have

$$\mathbf{A}_0 = \begin{bmatrix} \frac{\partial h}{\partial \rho} & \frac{\partial h}{\partial \psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{A.28}$$

And the first-order Lie derivatives are computed as:

$$L^1_{\mathbf{f}_1} h = \mathbf{A}_0 \begin{bmatrix} -c\psi \\ \frac{s\psi}{\rho} \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} -c\psi \\ \frac{s\psi}{\rho} \end{bmatrix} = \frac{s\psi}{\rho} \tag{A.29}$$

$$L^1_{\mathbf{f}_2} h = \mathbf{A}_0 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1 \tag{A.30}$$

Then, we compute $\mathbf{A}_{1_1}$ and $\mathbf{A}_{1_2}$ as:

$$\mathbf{A}_{1_1} = \begin{bmatrix} \frac{\partial}{\partial \rho}(L^1_{\mathbf{f}_1} h) & \frac{\partial}{\partial \psi}(L^1_{\mathbf{f}_1} h) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \rho}\left(\frac{s\psi}{\rho}\right) & \frac{\partial}{\partial \psi}\left(\frac{s\psi}{\rho}\right) \end{bmatrix} = \begin{bmatrix} -\frac{s\psi}{\rho^2} & \frac{c\psi}{\rho} \end{bmatrix} \tag{A.31}$$

$$\mathbf{A}_{1_2} = \begin{bmatrix} \frac{\partial}{\partial \rho}(L^1_{\mathbf{f}_2} h) & \frac{\partial}{\partial \psi}(L^1_{\mathbf{f}_2} h) \end{bmatrix} = \mathbf{0} \tag{A.32}$$

It is clear that $\mathbf{A}_K = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_{1_1} \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{s\psi}{\rho^2} & \frac{c\psi}{\rho} \\ \vdots & \vdots \end{bmatrix}$ is, in general, full-rank.

Thus far, we have shown that for all three cases $\mathbf{A}_K$ is full-rank. Note that the rank of the product of $\mathbf{A}_K$ and $\mathbf{M}$ is given by [see (2.28)]:

$$\text{rank}(\mathbf{A}_K \mathbf{M}) = \text{rank}(\mathbf{M}) - \dim(\text{null}(\mathbf{A}_K) \bigcap \text{rng}(\mathbf{M})) \tag{A.33}$$

Since we have shown above that in general matrix $\mathbf{A}_K$ is full column rank, i.e., $\mathrm{null}(\mathbf{A}_K) = \varnothing$, we have $\mathrm{rank}(\mathbf{A}_K\mathbf{M}) = \mathrm{rank}(\mathbf{M})$. Therefore, the row-span of $\mathbf{A}_K\mathbf{M}$ is identical to the row-span of $\mathbf{M}$, i.e., $\frac{d\rho}{d\mathbf{x}}$ and $\frac{d\psi}{d\mathbf{x}}$.

$$
\begin{aligned}
d\mathcal{G} &= \underset{\text{row}}{\text{span}} \begin{bmatrix} -c\theta & -s\theta & 0 & c\theta & s\theta \\ \frac{s\theta}{\rho} & -\frac{c\theta}{\rho} & -1 & -\frac{s\theta}{\rho} & \frac{c\theta}{\rho} \end{bmatrix} \\
&= \underset{\text{row}}{\text{span}} \left\{ \mathbf{J}\ \mathbf{Diag}\left(\frac{1}{\rho}, 1\right) \mathbf{C}^T(\psi) \begin{bmatrix} s\phi_R & -c\phi_R & -c\phi_R\delta x - s\phi_R\delta y & -s\phi_R & c\phi_R \\ c\phi_R & s\phi_R & s\phi_R\delta x - c\phi_R\delta y & -c\phi_R & -s\phi_R \end{bmatrix} \right\} \\
&= \underset{\text{row}}{\text{span}} \begin{bmatrix} s\phi_R & -c\phi_R & -c\phi_R\delta x - s\phi_R\delta y & -s\phi_R & c\phi_R \\ c\phi_R & s\phi_R & s\phi_R\delta x - c\phi_R\delta y & -c\phi_R & -s\phi_R \end{bmatrix}
\end{aligned} \tag{A.34}
$$

A.4 Proof of Lemma 2.5.3

Under the Gaussianity assumption, it is $p(\mathbf{x}_{R_k}|\mathbf{z}_{0:k}) = \mathcal{N}(\hat{\mathbf{x}}_{R_{k|k}}, \mathbf{P}_{RR_{k|k}})$, where $\mathbf{P}_{RR_{k|k}}$ is the covariance matrix corresponding to the robot pose, obtained by partitioning the state covariance matrix as $\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{RR_{k|k}} & \mathbf{P}_{RL_{k|k}} \\ \mathbf{P}^T_{RL_{k|k}} & \mathbf{P}_{LL_{k|k}} \end{bmatrix}$, and $p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k}) = \mathcal{N}(\hat{\mathbf{x}}_{k+1|k}, \mathbf{P}_{k+1|k})$.

The first term of the cost function (2.47) is computed as:

$$
\begin{aligned}
&\int \left\|\mathbf{x}_{R_k} - \mathbf{x}^\star_{R_{k|k}}\right\|^2 p(\mathbf{x}_{R_k}|\mathbf{z}_{0:k}) d\mathbf{x}_{R_k} \\
=&\int \left(\mathbf{x}^T_{R_k}\mathbf{x}_{R_k} - 2\mathbf{x}^T_{R_k}\mathbf{x}^\star_{R_{k|k}} + \mathbf{x}^{\star T}_{R_{k|k}}\mathbf{x}^\star_{R_{k|k}}\right) p(\mathbf{x}_{R_k}|\mathbf{z}_{0:k}) d\mathbf{x}_{R_k} \\
=&\ \mathbb{E}\left(\mathbf{x}^T_{R_k}\mathbf{x}_{R_k}\right) - 2\mathbb{E}\left(\mathbf{x}^T_{R_k}\right)\mathbf{x}^\star_{R_{k|k}} + \mathbf{x}^{\star T}_{R_{k|k}}\mathbf{x}^\star_{R_{k|k}} \\
=&\ \mathrm{tr}\left(\mathbf{P}_{RR_{k|k}} + \hat{\mathbf{x}}_{R_{k|k}}\hat{\mathbf{x}}^T_{R_{k|k}}\right) - 2\hat{\mathbf{x}}^T_{R_{k|k}}\mathbf{x}^\star_{R_{k|k}} + \mathbf{x}^{\star T}_{R_{k|k}}\mathbf{x}^\star_{R_{k|k}} \\
=&\ \mathrm{tr}\left(\mathbf{P}_{RR_{k|k}}\right) + \hat{\mathbf{x}}^T_{R_{k|k}}\hat{\mathbf{x}}_{R_{k|k}} - 2\hat{\mathbf{x}}^T_{R_{k|k}}\mathbf{x}^\star_{R_{k|k}} + \mathbf{x}^{\star T}_{R_{k|k}}\mathbf{x}^\star_{R_{k|k}} \\
=&\ \mathrm{tr}\left(\mathbf{P}_{RR_{k|k}}\right) + \left\|\hat{\mathbf{x}}_{R_{k|k}} - \mathbf{x}^\star_{R_{k|k}}\right\|^2
\end{aligned} \tag{A.35}
$$

where $\mathbb{E}(\cdot)$ denotes expectation and $\mathrm{tr}(\cdot)$ the matrix trace. Proceeding similarly, the second term of the cost function (2.47) can be derived as:

$$
\int \left\|\mathbf{x}_{k+1} - \mathbf{x}^\star_{k+1|k}\right\|^2 p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k}) d\mathbf{x}_{k+1} = \mathrm{tr}\left(\mathbf{P}_{k+1|k}\right) + \left\|\hat{\mathbf{x}}_{k+1|k} - \mathbf{x}^\star_{k+1|k}\right\|^2 \tag{A.36}
$$

Using (A.35) and (A.36), as well as the fact that the true $\mathbf{P}_{RR_{k|k}}$ and $\mathbf{P}_{k+1|k}$ are independent of

the linearization points, the following equivalence is immediate:

$$\begin{aligned}
&\min_{\mathbf{x}^\star_{R_{k|k}},\, \mathbf{x}^\star_{k+1|k}} \operatorname{tr}\left(\mathbf{P}_{RR_{k|k}}\right) + \operatorname{tr}\left(\mathbf{P}_{k+1|k}\right) + \left\|\hat{\mathbf{x}}_{R_{k|k}} - \mathbf{x}^\star_{R_{k|k}}\right\|^2 + \left\|\hat{\mathbf{x}}_{k+1|k} - \mathbf{x}^\star_{k+1|k}\right\|^2 \\
\Leftrightarrow \quad &\min_{\mathbf{x}^\star_{R_{k|k}},\, \mathbf{x}^\star_{k+1|k}} \left\|\hat{\mathbf{x}}_{R_{k|k}} - \mathbf{x}^\star_{R_{k|k}}\right\|^2 + \left\|\hat{\mathbf{x}}_{k+1|k} - \mathbf{x}^\star_{k+1|k}\right\|^2
\end{aligned}$$

We now derive the following identities for the observability constraint (2.48) [see (2.35) and (2.42)]:

$$\begin{aligned}
& \mathbf{H}_{k+1}\mathbf{\Phi}_k \cdots \mathbf{\Phi}_{k_o}\mathbf{N} = \mathbf{0} \\
\Leftrightarrow \quad & \mathbf{H}_{L_{k+1}} \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}\left(\mathbf{p}^\star_{L_{k+1|k}} - \mathbf{p}^\star_{R_{k_o|k_o}} - \sum_{j=k_o+1}^{k} \Delta\mathbf{p}^\star_{R_j}\right) & \mathbf{I}_2 \end{bmatrix} \mathbf{N} = \mathbf{0} \\
\Leftrightarrow \quad & \mathbf{p}^\star_{L_{k+1|k}} - \mathbf{p}^\star_{R_{k|k}} = \hat{\mathbf{p}}_{L_{k_o|k_o}} - \mathbf{p}^\star_{R_{k|k-1}} + \sum_{j=k_o}^{k-1} \Delta\mathbf{p}^\star_{R_j}
\end{aligned}$$

This completes the proof.

A.5 Proof of Lemma 2.5.4

The constraint equation (2.55) states that the rows of $\mathbf{H}_{k+1}$ lie in the left nullspace of the matrix $\mathbf{V}$. Therefore, if $\mathbf{L}$ is a matrix whose rows span this nullspace, $\mathbf{H}_{k+1}$ can be written as:

$$\mathbf{H}_{k+1} = \mathbf{A}\mathbf{L} \tag{A.37}$$

where $\mathbf{A}$ is an unknown matrix which we seek to compute. We note that there are several possible ways of computing an appropriate matrix $\mathbf{L}$, whose rows lie in the nullspace of $\mathbf{V}$. For instance, such a matrix is given, in closed form, by the expression:

$$\mathbf{L} = \begin{bmatrix} \mathbf{I}_{\dim(\mathbf{z})} & \mathbf{0} \end{bmatrix} \left(\mathbf{I}_{\dim(\mathbf{x})} - \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\right) =: \mathbf{\Gamma}\mathbf{\Pi} \tag{A.38}$$

It is not difficult to see that $\mathbf{\Pi} := \mathbf{I}_{\dim(\mathbf{x})} - \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$ is an orthogonal projection matrix (i.e., $\mathbf{\Pi}^2 = \mathbf{\Pi}$ and $\mathbf{\Pi}^T = \mathbf{\Pi}$) and hence has the eigenvalues of either 1 or 0, whose *reduced* SVD can be written as $\mathbf{\Pi} = \mathbf{Q}\mathbf{Q}^T$. Using this result, $\mathbf{L}^T$ immediately becomes:

$$\mathbf{L}^T = \mathbf{Q}\mathbf{Q}^T\mathbf{\Gamma}^T \tag{A.39}$$

By substituting (A.39) into the cost function (2.54), we have:

$$\min \ ||\mathbf{H}_o - \mathbf{H}_{k+1}||_F^2 = ||\mathbf{Q}\mathbf{Q}^T\mathbf{\Gamma}^T\mathbf{A}^T - \mathbf{H}_o^T||_F^2 = ||\mathbf{Q}^T\mathbf{\Gamma}^T\mathbf{A}^T - \mathbf{Q}^T{\mathbf{H}_o}^T||_F^2 \tag{A.40}$$

$$\Rightarrow \ \mathbf{A} = \mathbf{H}_o\mathbf{Q}\left(\mathbf{\Gamma}\mathbf{Q}\right)^{-1} \tag{A.41}$$

Therefore, substitution of the above equation in (A.37) yields:

$$\begin{aligned}\mathbf{H}_{k+1} &= \mathbf{H}_o\mathbf{Q}\left(\mathbf{\Gamma}\mathbf{Q}\right)^{-1}\mathbf{\Gamma}\mathbf{Q}\mathbf{Q}^T = \mathbf{H}_o\mathbf{Q}\mathbf{Q}^T = \mathbf{H}_o\mathbf{\Pi} \\ &= \mathbf{H}_o\left(\mathbf{I}_{\dim(\mathbf{x})} - \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\right)\end{aligned} \tag{A.42}$$

This completes the proof.

A.6 Observability analysis for robocentric mapping

The robocentric mapping filter [26, 27] consists of three steps: prediction, update and composition. In order to construct the system equations to perform observability analysis, we combine the composition and prediction steps into one model. The state vector is ${}^{R_k}\mathbf{x}_k \triangleq \begin{bmatrix}{}^{R_k}\mathbf{x}_{G_k}^T & {}^{R_k}\mathbf{p}_{L_k}^T\end{bmatrix}^T$, where ${}^{R_k}\mathbf{x}_{G_k}$ denotes the pose of the origin of the global frame with respect to the robot local frame at time-step k, and ${}^{R_k}\mathbf{p}_{L_k}$ is the landmark position with respect to the robot local frame at time-step k. The propagation equations are given by:

$${}^{R_{k+1}}\mathbf{p}_{G_{k+1}} = \mathbf{C}^T({}^{R_k}\phi_{R_{k+1}})({}^{R_k}\mathbf{p}_{G_k} - {}^{R_k}\mathbf{p}_{R_{k+1}}) \tag{A.43}$$

$${}^{R_{k+1}}\phi_{G_{k+1}} = {}^{R_k}\phi_{G_k} - {}^{R_k}\phi_{R_{k+1}} \tag{A.44}$$

$${}^{R_{k+1}}\mathbf{p}_{L_{k+1}} = \mathbf{C}^T({}^{R_k}\phi_{R_{k+1}})({}^{R_k}\mathbf{p}_{L_k} - {}^{R_k}\mathbf{p}_{R_{k+1}}) \tag{A.45}$$

where ${}^{R_k}\mathbf{x}_{R_{k+1}} = \begin{bmatrix}{}^{R_k}\mathbf{p}_{R_{k+1}}^T & {}^{R_k}\phi_{R_{k+1}}\end{bmatrix}^T$ is the robot pose change between time-steps k and $k+1$. This can be obtained from the odometry measurements $\{v_{m_k}, \omega_{m_k}\}$, which is corrupted by zero-mean white Gaussian noise $\mathbf{w}_k$, i.e.,

$${}^{R_k}\mathbf{x}_{R_{k+1}} = {}^{R_k}\hat{\mathbf{x}}_{R_{k+1|k}} - {}^{R_k}\tilde{\mathbf{x}}_{R_{k+1|k}} = \begin{bmatrix} v_{m_k}\delta t \\ 0 \\ \omega_{m_k}\delta t \end{bmatrix} - \begin{bmatrix} \delta t & 0 \\ 0 & 0 \\ 0 & \delta t \end{bmatrix}\mathbf{w}_k$$

By linearizing the above equations (A.43)-(A.45) at the current estimates, ${}^{R_k}\hat{\mathbf{x}}_{k|k}$ and ${}^{R_k}\hat{\mathbf{x}}_{R_{k+1|k}}$, we obtain the error-state propagation equations as follows:

$$ {}^{R_{k+1}}\tilde{\mathbf{x}}_{G_{k+1|k}} = \mathbf{\Phi}_{G_k} {}^{R_k}\tilde{\mathbf{x}}_{G_{k|k}} + \mathbf{J}_{1_k} {}^{R_k}\tilde{\mathbf{x}}_{R_{k+1|k}} \tag{A.46} $$

$$ {}^{R_{k+1}}\tilde{\mathbf{p}}_{L_{k+1|k}} = \mathbf{\Phi}_{L_k} {}^{R_k}\tilde{\mathbf{p}}_{L_{k|k}} + \mathbf{J}_{2_k} {}^{R_k}\tilde{\mathbf{x}}_{R_{k+1|k}} \tag{A.47} $$

where

$$ \mathbf{\Phi}_{G_k} = \begin{bmatrix} \mathbf{C}^T({}^{R_k}\hat{\phi}_{R_{k+1|k}}) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{A.48} $$

$$ \mathbf{J}_{1_k} = \begin{bmatrix} -\mathbf{C}^T({}^{R_k}\hat{\phi}_{R_{k+1|k}}) & \mathbf{J}^T\mathbf{C}^T({}^{R_k}\hat{\phi}_{R_{k+1|k}})({}^{R_k}\hat{\mathbf{p}}_{G_{k|k}} - {}^{R_k}\hat{\mathbf{p}}_{R_{k+1|k}}) \\ \mathbf{0}_{1\times 2} & -1 \end{bmatrix} \tag{A.49} $$

$$ \mathbf{\Phi}_{L_k} = \mathbf{C}^T({}^{R_k}\hat{\phi}_{R_{k+1|k}}) \tag{A.50} $$

$$ \mathbf{J}_{2_k} = \begin{bmatrix} -\mathbf{C}^T({}^{R_k}\hat{\phi}_{R_{k+1|k}}) & \mathbf{J}^T\mathbf{C}^T({}^{R_k}\hat{\phi}_{R_{k+1|k}})({}^{R_k}\hat{\mathbf{p}}_{L_{k|k}} - {}^{R_k}\hat{\mathbf{p}}_{R_{k+1|k}}) \end{bmatrix} \tag{A.51} $$

Stacking (A.46) and (A.47) into a matrix form, we have:

$$ {}^{R_{k+1}}\tilde{\mathbf{x}}_{k+1|k} = \mathbf{\Phi}_k {}^{R_k}\tilde{\mathbf{x}}_{k|k} + \mathbf{G}_k {}^{R_k}\tilde{\mathbf{x}}_{R_{k+1|k}} \tag{A.52} $$

where

$$ \mathbf{\Phi}_k = \begin{bmatrix} \mathbf{\Phi}_{G_k} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi}_{L_k} \end{bmatrix} \tag{A.53} $$

$$ \mathbf{G}_k = \begin{bmatrix} \mathbf{J}_{1_k} \\ \mathbf{J}_{2_k} \end{bmatrix} \tag{A.54} $$

Due to the robot-relative formulation, the relative-position measurement at time-step $k+1$ becomes linear, i.e.,

$$ \mathbf{z}_{k+1} = {}^{R_{k+1}}\mathbf{p}_{L_{k+1}} + \mathbf{v}_{k+1} \tag{A.55} $$

where $\mathbf{v}_{k+1}$ is zero-mean white Gaussian measurement noise. The measurement error equation is given by:

$$ \begin{aligned} \tilde{\mathbf{z}}_{k+1} &= \mathbf{H}_{k+1} {}^{R_{k+1}}\tilde{\mathbf{x}}_{k+1|k} + \mathbf{v}_{k+1} \\ &= \begin{bmatrix} \mathbf{H}_{G_{k+1}} & \mathbf{H}_{L_{k+1}} \end{bmatrix} \begin{bmatrix} {}^{R_{k+1}}\tilde{\mathbf{x}}_{G_{k+1|k}} \\ {}^{R_{k+1}}\tilde{\mathbf{p}}_{L_{k+1|k}} \end{bmatrix} + \mathbf{v}_{k+1} \\ &= \begin{bmatrix} \mathbf{0}_{2\times 3} & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} {}^{R_{k+1}}\tilde{\mathbf{x}}_{G_{k+1|k}} \\ {}^{R_{k+1}}\tilde{\mathbf{p}}_{L_{k+1|k}} \end{bmatrix} + \mathbf{v}_{k+1} \end{aligned} \tag{A.56} $$

Proceeding similarly as before, we employ the local observability matrix to investigate the observability properties of robocentric mapping. Substituting the matrices $\mathbf{\Phi}_k$ [see (A.53)] and $\mathbf{H}_{k+1}$ [see (A.56)] into (2.22), we obtain the local observability matrix of this particular system as follows:

$$\mathbf{M} = \begin{bmatrix} \mathbf{0}_{2\times 3} & \mathbf{I}_2 \\ \mathbf{0}_{2\times 3} & \mathbf{\Phi}_{L_{k_o}} \\ \vdots & \vdots \\ \mathbf{0}_{2\times 3} & \mathbf{\Phi}_{L_{k_o+m-1}} \cdots \mathbf{\Phi}_{L_{k_o}} \end{bmatrix} \tag{A.57}$$

$$= \begin{bmatrix} \mathbf{0}_{2\times 3} & \mathbf{I}_2 \\ \mathbf{0}_{2\times 3} & \mathbf{C}^T\big({}^{R_{k_o}}\hat{\phi}_{R_{k_o+1|k_o}}\big) \\ \vdots & \vdots \\ \mathbf{0}_{2\times 3} & \mathbf{C}^T\big({}^{R_{k_o+m-1}}\hat{\phi}_{R_{k_o+m|k_o+m-1}}\big)\cdots\mathbf{C}^T\big({}^{R_{k_o}}\hat{\phi}_{R_{k_o+1|k_o}}\big) \end{bmatrix} \tag{A.58}$$

It is not difficult to see that in general, $\mathrm{rank}(\mathbf{M}) = 2$, and hence the robocentric mapping is unobservable. Moreover, the right nullspace can be found as follows:

$$\mathcal{N}(\mathbf{M}) = \underset{\text{col.}}{\mathrm{span}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.59}$$

From the structure of this nullspace, the unobservable states are the ones associated to the origin of the global frame in the robot's local frame, while the position of the landmark in the robot's local frame (i.e., the robot-to-landmark relative position) is observable. This agrees with what we found for the ideal EKF, and thus the robocentric mapping filter employs a system model that has the correct observability properties.

Appendix B

Appendices for Chapter 3

B.1 Proof of Lemma 3.4.4

In the general case where $N > 2$ robots comprise the team, proceeding similarly to the analysis of the ideal EKF-CL, the observability matrix $\mathbf{M}$ can be obtained as follows:

$$\mathbf{M} = \underbrace{-\mathbf{Diag}\left((\nabla \mathbf{h}_{k_o}^{(12)})\mathbf{A}(\hat{\phi}_{1_{k_o|k_o-1}}),\ \cdots,\ (\nabla \mathbf{h}_{k_o+m}^{(NN-1)})\mathbf{A}(\hat{\phi}_{N_{k_o+m|k_o+m-1}})\right)}_{\mathbf{D}} \times \tag{B.1}$$

$$\underbrace{\begin{bmatrix}
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{21}(k_o,k_o) & -\mathbf{I}_2 & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{31}(k_o,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{N1}(k_o,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \cdots & -\mathbf{I}_2 & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & -1 \\
\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\\
-\mathbf{I}_2 & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{1N}(k_o,k_o) \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{2N}(k_o,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{N-1N}(k_o,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\\
\\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{21}(k_o+1,k_o) & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{22}(k_o+1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{31}(k_o+1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{33}(k_o+1,k_o) & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{N1}(k_o+1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{NN}(k_o+1,k_o) \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & -1 \\
\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\\
-\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{11}(k_o+1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{1N}(k_o+1,k_o) \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{22}(k_o+1,k_o) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{2N}(k_o+1,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{N-1N}(k_o+1,k_o) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{21}(k_o+m,k_o+m-1) & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{22}(k_o+m,k_o+m-1) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{31}(k_o+m,k_o+m-1) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{33}(k_o+m,k_o+m-1) & \cdots & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \cdots & \mathbf{0}_{1\times 2} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{N1}(k_o+m,k_o+m-1) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{NN}(k_o+m,k_o+m-1) \\
\mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & -1 \\
\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\\
-\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{11}(k_o+m,k_o+m-1) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{1N}(k_o+m,k_o+m-1) \\
\mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_2 & -\mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{22}(k_o+m,k_o+m-1) & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{2N}(k_o+m,k_o+m-1) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & \cdots & \mathbf{I}_2 & \mathbf{J}\boldsymbol{\Delta}\hat{\mathbf{p}}_{N-1N}(k_o+m,k_o+m-1) \\
\mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 0 & \cdots & \mathbf{0}_{1\times 2} & 1
\end{bmatrix}}_{\mathbf{U}}$$

Analogously to the proof of Lemma 3.4.2, we denote $\mathbf{U} \triangleq \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_{3N} \end{bmatrix}$, and observe that

$$\mathbf{u}_1 = -\sum_{i=2}^{N} \mathbf{u}_{3i-2}\,, \quad \mathbf{u}_2 = -\sum_{i=2}^{N} \mathbf{u}_{3i-1}$$

while

$$\sum_{i=1}^{N} \mathbf{u}_{3i} \neq \sum_{i=2}^{N} \alpha_{2i-1}\mathbf{u}_{3i-2} + \sum_{i=2}^{N} \alpha_{2i}\mathbf{u}_{3i-1}$$

where $\begin{bmatrix} \alpha_{2i-1} \\ \alpha_{2i} \end{bmatrix} \triangleq -\mathbf{J}\boldsymbol{\delta}\hat{\mathbf{p}}_{i1}(k_o, k_o)$, $\forall i = 2, \ldots, N$. This is due to that fact that $\mathbf{u}_{3i}$, $i = 1, \ldots, N$, become general vectors and hence are no longer linear combinations of any other columns. This is in contrast to the case of the ideal EKF-CL (see Lemma 3.4.2). As a result, one possible basis of the range of matrix $\mathbf{U}$ is its column vectors $\{\mathbf{u}_i\}_{i=3}^{3N}$, i.e., $\mathrm{rng}(\mathbf{U}) = \underset{\text{col.}}{\mathrm{span}} \begin{bmatrix} \mathbf{u}_3 & \cdots & \mathbf{u}_{3N} \end{bmatrix}$. Thus, $\mathrm{rank}(\mathbf{U}) = 3N - 2$. Analogously, we observe that in general $\mathbf{D}\mathbf{u}_i \neq \mathbf{0}$, for $i = 3, \ldots, 3N$. Moreover, note that any vector $\mathbf{x} \in \mathrm{rng}(\mathbf{U}) \setminus \mathbf{0}$ can be written as $\mathbf{x} = \sum_{i=1}^{3N-2} \beta_i \mathbf{u}_{i+2}$ for some $\beta_i \in \mathbb{R}$, where β_i's are not simultaneously equal to zero. Thus, we see that in general $\mathbf{D}\mathbf{x} = \sum_{i=1}^{3N-2} \beta_i \mathbf{D}\mathbf{u}_{i+2} \neq \mathbf{0}$, which implies that $\mathbf{x}$ does not belong to the null space, $\mathrm{null}(\mathbf{D})$, of $\mathbf{D}$. Therefore, $\dim(\mathrm{null}(\mathbf{D}) \bigcap \mathrm{rng}(\mathbf{U})) = 0$, and, finally, based on theorem (4.5.1) in [113] [also see (2.28)], $\mathrm{rank}(\mathbf{M}) = \mathrm{rank}(\mathbf{U}) - \dim(\mathrm{null}(\mathbf{D}) \bigcap \mathrm{rng}(\mathbf{U})) = \mathrm{rank}(\mathbf{U}) = 3N - 2$.

B.2 Proof of Lemma 3.5.2

Under the Gaussianity assumption, it is $p(\mathbf{x}_k|\mathbf{z}_{0:k}) = \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$, and $p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k}) = \mathcal{N}(\hat{\mathbf{x}}_{k+1|k}, \mathbf{P}_{k+1|k})$. The first term of the cost function (3.49) is computed as

$$\begin{aligned}
&\int \left\|\mathbf{x}_k - \mathbf{x}^\star_{k|k}\right\|^2 p(\mathbf{x}_k|\mathbf{z}_{0:k}) d\mathbf{x}_k \\
=&\int \left(\mathbf{x}_k^T\mathbf{x}_k - 2\mathbf{x}_k^T\mathbf{x}^\star_{k|k} + \mathbf{x}^{\star T}_{k|k}\mathbf{x}^\star_{k|k}\right) p(\mathbf{x}_k|\mathbf{z}_{0:k}) d\mathbf{x}_k \\
=&\ \mathbb{E}\left(\mathbf{x}_k^T\mathbf{x}_k|\mathbf{z}_{0:k}\right) - 2\mathbb{E}\left(\mathbf{x}_k^T|\mathbf{z}_{0:k}\right)\mathbf{x}^\star_{k|k} + \mathbf{x}^{\star T}_{k|k}\mathbf{x}^\star_{k|k} \\
=&\ \mathrm{tr}\left(\mathbf{P}_{k|k} + \hat{\mathbf{x}}_{k|k}\hat{\mathbf{x}}_{k|k}^T\right) - 2\hat{\mathbf{x}}^T_{k|k}\mathbf{x}^\star_{k|k} + \mathbf{x}^{\star T}_{k|k}\mathbf{x}^\star_{k|k} \\
=&\ \mathrm{tr}\left(\mathbf{P}_{k|k}\right) + \hat{\mathbf{x}}^T_{k|k}\hat{\mathbf{x}}_{k|k} - 2\hat{\mathbf{x}}^T_{k|k}\mathbf{x}^\star_{k|k} + \mathbf{x}^{\star T}_{k|k}\mathbf{x}^\star_{k|k} \\
=&\ \mathrm{tr}\left(\mathbf{P}_{k|k}\right) + \left\|\hat{\mathbf{x}}_{k|k} - \mathbf{x}^\star_{k|k}\right\|^2
\end{aligned} \tag{B.2}$$

Proceeding similarly, the second term of the cost function (3.49) can be derived as

$$\int \left\|\mathbf{x}_{k+1} - \mathbf{x}^\star_{k+1|k}\right\|^2 p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k}) d\mathbf{x}_{k+1} = \mathrm{tr}\left(\mathbf{P}_{k+1|k}\right) + \left\|\hat{\mathbf{x}}_{k+1|k} - \mathbf{x}^\star_{k+1|k}\right\|^2 \tag{B.3}$$

Using (B.2) and (B.3), as well as the fact that the true $\mathbf{P}_{k|k}$ and $\mathbf{P}_{k+1|k}$ are independent of the linearization points, the following equivalence holds:

$$\begin{aligned}
&\min_{\mathbf{x}^\star_{k|k},\, \mathbf{x}^\star_{k+1|k}} \ \mathrm{tr}\left(\mathbf{P}_{k|k}\right) + \mathrm{tr}\left(\mathbf{P}_{k+1|k}\right) + \left\|\hat{\mathbf{x}}_{k|k} - \mathbf{x}^\star_{k|k}\right\|^2 + \left\|\hat{\mathbf{x}}_{k+1|k} - \mathbf{x}^\star_{k+1|k}\right\|^2 \\
\Leftrightarrow \min_{\mathbf{x}^\star_{k|k},\, \mathbf{x}^\star_{k+1|k}} &\ \left\|\hat{\mathbf{x}}_{k|k} - \mathbf{x}^\star_{k|k}\right\|^2 + \left\|\hat{\mathbf{x}}_{k+1|k} - \mathbf{x}^\star_{k+1|k}\right\|^2
\end{aligned} \tag{B.4}$$

We now derive the following identities for the observability constraint (3.50) [see (3.39) and (3.43)]:

$$\begin{aligned}
&\mathbf{H}_{k+1}\mathbf{\Phi}_k \cdots \mathbf{\Phi}_{k_o}\mathbf{N} = \mathbf{0} \\
\Leftrightarrow &\sum_{\tau=k_o}^{k} (\mathbf{p}^\star_{2_{\tau|\tau}} - \mathbf{p}^\star_{2_{\tau|\tau-1}}) - \sum_{\tau=k_o}^{k} (\mathbf{p}^\star_{1_{\tau|\tau}} - \mathbf{p}^\star_{1_{\tau|\tau-1}}) + \\
&\left(\mathbf{p}^\star_{2_{k_o|k_o-1}} - \mathbf{p}^\star_{1_{k_o|k_o-1}}\right) - \left(\hat{\mathbf{p}}_{2_{k_o|k_o-1}} - \hat{\mathbf{p}}_{1_{k_o|k_o-1}}\right) = \mathbf{0}
\end{aligned} \tag{B.5}$$

$$\begin{aligned}
\Leftrightarrow &\mathbf{p}^\star_{2_{k|k}} - \mathbf{p}^\star_{1_{k|k}} = \left(\mathbf{p}^\star_{2_{k|k-1}} - \mathbf{p}^\star_{1_{k|k-1}}\right) - \\
&\sum_{\tau=k_o}^{k-1} (\mathbf{p}^\star_{2_{\tau|\tau}} - \mathbf{p}^\star_{2_{\tau|\tau-1}}) + \sum_{\tau=k_o}^{k-1} (\mathbf{p}^\star_{1_{\tau|\tau}} - \mathbf{p}^\star_{1_{\tau|\tau-1}})
\end{aligned} \tag{B.6}$$

where we have used the fact that the linearization points, during propagation at time-step k_o are the propagated filter estimates, i.e., $\mathbf{p}^{\star}_{1_{k_o|k_o-1}} = \hat{\mathbf{p}}_{1_{k_o|k_o-1}}$ and $\mathbf{p}^{\star}_{2_{k_o|k_o-1}} = \hat{\mathbf{p}}_{2_{k_o|k_o-1}}$. This completes the proof.

B.3 Proof of Lemma 3.5.3

We first introduce the following definition similar to (3.34), which will be useful for the ensuring derivations:

$$\boldsymbol{\Delta}\mathbf{p}^{\star}_{ij}(k,\ell) \triangleq \mathbf{p}^{\star}_{i_{k|k-1}} - \mathbf{p}^{\star}_{j_{k_o|k_o-1}} - \sum_{\tau=k_o}^{\ell}\left(\mathbf{p}^{\star}_{j_{\tau|\tau}} - \mathbf{p}^{\star}_{j_{\tau|\tau-1}}\right) \tag{B.7}$$

Using this definition as well as (3.8), (3.11), and (3.43), we have the following identity [also see (3.39)]:

$$\begin{aligned}
&\mathbf{H}^{(ij)}_{k+1}\boldsymbol{\Phi}_k\cdots\boldsymbol{\Phi}_{k_o}\mathbf{N} = \mathbf{0}\\
\Leftrightarrow\ &\boldsymbol{\Delta}\mathbf{p}^{\star}_{ji}(k+1,k) - \boldsymbol{\Delta}\mathbf{p}^{\star}_{jj}(k+1,k) + \hat{\mathbf{p}}_{i_{k_o|k_o-1}} - \hat{\mathbf{p}}_{j_{k_o|k_o-1}} = \mathbf{0}\\
\Leftrightarrow\ &\sum_{\tau=k_o}^{k}(\mathbf{p}^{\star}_{j_{\tau|\tau}} - \mathbf{p}^{\star}_{j_{\tau|\tau-1}}) - \sum_{\tau=k_o}^{k}(\mathbf{p}^{\star}_{i_{\tau|\tau}} - \mathbf{p}^{\star}_{i_{\tau|\tau-1}}) + \left(\mathbf{p}^{\star}_{j_{k_o|k_o-1}} - \mathbf{p}^{\star}_{i_{k_o|k_o-1}}\right) - \left(\hat{\mathbf{p}}_{j_{k_o|k_o-1}} - \hat{\mathbf{p}}_{i_{k_o|k_o-1}}\right) = \mathbf{0}\\
\Leftrightarrow\ &\mathbf{p}^{\star}_{j_{k|k}} - \mathbf{p}^{\star}_{i_{k|k}} = \left(\mathbf{p}^{\star}_{j_{k|k-1}} - \mathbf{p}^{\star}_{i_{k|k-1}}\right) - \sum_{\tau=k_o}^{k-1}(\mathbf{p}^{\star}_{j_{\tau|\tau}} - \mathbf{p}^{\star}_{j_{\tau|\tau-1}}) + \sum_{\tau=k_o}^{k-1}(\mathbf{p}^{\star}_{i_{\tau|\tau}} - \mathbf{p}^{\star}_{i_{\tau|\tau-1}})\\
\Leftrightarrow\ &\sum_{\tau=k_o}^{k}(\mathbf{p}^{\star}_{i_{\tau|\tau}} - \mathbf{p}^{\star}_{i_{\tau|\tau-1}}) - \sum_{\tau=k_o}^{k}(\mathbf{p}^{\star}_{j_{\tau|\tau}} - \mathbf{p}^{\star}_{j_{\tau|\tau-1}}) + \left(\mathbf{p}^{\star}_{i_{k_o|k_o-1}} - \mathbf{p}^{\star}_{j_{k_o|k_o-1}}\right) - \left(\hat{\mathbf{p}}_{i_{k_o|k_o-1}} - \hat{\mathbf{p}}_{j_{k_o|k_o-1}}\right) = \mathbf{0}\\
\Leftrightarrow\ &\boldsymbol{\Delta}\mathbf{p}^{\star}_{ij}(k+1,k) - \boldsymbol{\Delta}\mathbf{p}^{\star}_{ii}(k+1,k) + \hat{\mathbf{p}}_{j_{k_o|k_o-1}} - \hat{\mathbf{p}}_{i_{k_o|k_o-1}} = \mathbf{0}\\
\Leftrightarrow\ &\mathbf{H}^{(ji)}_{k+1}\boldsymbol{\Phi}_k\cdots\boldsymbol{\Phi}_{k_o}\mathbf{N} = \mathbf{0}
\end{aligned} \tag{B.8}$$

This equivalence implies that the measurements $\mathbf{z}_{k+1}^{(ij)}$ and $\mathbf{z}_{k+1}^{(ji)}$ produce the same observability constraint. Moreover, from the above results, we have:

$$\begin{aligned}
&\mathbf{H}_{k+1}^{(ij)}\mathbf{\Phi}_k\cdots\mathbf{\Phi}_{k_o}\mathbf{N}=\mathbf{0}\\
\Leftrightarrow\ &\mathbf{p}_{j_{k|k}}^\star-\mathbf{p}_{i_{k|k}}^\star=\left(\mathbf{p}_{j_{k|k-1}}^\star-\mathbf{p}_{i_{k|k-1}}^\star\right)-\sum_{\tau=k_o}^{k-1}(\mathbf{p}_{j_{\tau|\tau}}^\star-\mathbf{p}_{j_{\tau|\tau-1}}^\star)+\sum_{\tau=k_o}^{k-1}(\mathbf{p}_{i_{\tau|\tau}}^\star-\mathbf{p}_{i_{\tau|\tau-1}}^\star)
\end{aligned}\tag{B.9}$$

$$\begin{aligned}
&\mathbf{H}_{k+1}^{(jm)}\mathbf{\Phi}_k\cdots\mathbf{\Phi}_{k_o}\mathbf{N}=\mathbf{0}\\
\Leftrightarrow\ &\mathbf{p}_{m_{k|k}}^\star-\mathbf{p}_{j_{k|k}}^\star=\left(\mathbf{p}_{m_{k|k-1}}^\star-\mathbf{p}_{j_{k|k-1}}^\star\right)-\sum_{\tau=k_o}^{k-1}(\mathbf{p}_{m_{\tau|\tau}}^\star-\mathbf{p}_{m_{\tau|\tau-1}}^\star)+\sum_{\tau=k_o}^{k-1}(\mathbf{p}_{j_{\tau|\tau}}^\star-\mathbf{p}_{j_{\tau|\tau-1}}^\star)
\end{aligned}\tag{B.10}$$

By subtracting (B.9) from (B.10), we have the following equivalence:

$$\begin{aligned}
&\left(\mathbf{p}_{m_{k|k}}^\star-\mathbf{p}_{j_{k|k}}^\star\right)-\left(\mathbf{p}_{j_{k|k}}^\star-\mathbf{p}_{i_{k|k}}^\star\right)=\\
&\mathbf{p}_{m_{k|k}}^\star-\mathbf{p}_{i_{k|k}}^\star=\left(\mathbf{p}_{m_{k|k-1}}^\star-\mathbf{p}_{i_{k|k-1}}^\star\right)-\sum_{\tau=k_o}^{k-1}(\mathbf{p}_{m_{\tau|\tau}}^\star-\mathbf{p}_{m_{\tau|\tau-1}}^\star)+\sum_{\tau=k_o}^{k-1}(\mathbf{p}_{i_{\tau|\tau}}^\star-\mathbf{p}_{i_{\tau|\tau-1}}^\star)\\
\Leftrightarrow\ &\mathbf{H}_{k+1}^{(im)}\mathbf{\Phi}_k\cdots\mathbf{\Phi}_{k_o}\mathbf{N}=\mathbf{0}
\end{aligned}\tag{B.11}$$

This result implies that for the three connected measurements, $\mathbf{z}_{k+1}^{(ij)}$, $\mathbf{z}_{k+1}^{(jm)}$ and $\mathbf{z}_{k+1}^{(im)}$, the observability constraint imposed by one measurement (e.g., $\mathbf{z}_{k+1}^{(im)}$) can be equivalently inferred from those imposed by the other two measurements (e.g., $\mathbf{z}_{k+1}^{(ij)}$ and $\mathbf{z}_{k+1}^{(jm)}$). Therefore, from the above results (B.8)-(B.11), given the connected RMG, the observability constraint imposed by any measurement $\mathbf{z}_{k+1}^{(ij)}$ can be equivalently inferred from those imposed by the measurements $\mathbf{z}_{k+1}^{(1i)}$ and $\mathbf{z}_{k+1}^{(1j)}$. We thus have [see (B.1)]:

$$\begin{aligned}
&\mathbf{H}_{k+1}\mathbf{\Phi}_k\cdots\mathbf{\Phi}_{k_o}\mathbf{N}=\mathbf{0}\\
\Leftrightarrow\ &\sum_{\tau=k_o}^{k}(\mathbf{p}_{i_{\tau|\tau}}^\star-\mathbf{p}_{i_{\tau|\tau-1}}^\star)-\sum_{\tau=k_o}^{k}(\mathbf{p}_{1_{\tau|\tau}}^\star-\mathbf{p}_{1_{\tau|\tau-1}}^\star)-\left(\mathbf{p}_{i_{k_o|k_o-1}}^\star-\mathbf{p}_{1_{k_o|k_o-1}}^\star\right)+\left(\hat{\mathbf{p}}_{i_{k_o|k_o-1}}-\hat{\mathbf{p}}_{1_{k_o|k_o-1}}\right)=\mathbf{0}\\
\Leftrightarrow\ &\mathbf{p}_{i_{k|k}}^\star-\mathbf{p}_{1_{k|k}}^\star=\left(\mathbf{p}_{i_{k|k-1}}^\star-\mathbf{p}_{1_{k|k-1}}^\star\right)-\sum_{\tau=k_o}^{k-1}(\mathbf{p}_{i_{\tau|\tau}}^\star-\mathbf{p}_{i_{\tau|\tau-1}}^\star)+\sum_{\tau=k_o}^{k-1}(\mathbf{p}_{1_{\tau|\tau}}^\star-\mathbf{p}_{1_{\tau|\tau-1}}^\star),\ \ \forall i=2,\ldots,N
\end{aligned}$$

Note that the equivalence of the cost function can be proved in the same way as in Lemma 3.5.2. This completes the proof.

Table B.1: Different state estimates used in computing the state-propagation Jacobian matrix of robot i ($i = 1, 2$) at time-step k for the four estimators (i.e., the standard EKF, the OC-EKFs)

Std-EKF/OC-EKF3	$\mathbf{\Phi}_{i_k} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{i_{k+1\|k}} - \hat{\mathbf{p}}_{i_{k\|k}}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -v_{m_{i,k}} s\hat{\phi}_{i_{k\|k}} \delta t \\ 0 & 1 & v_{m_{i,k}} c\hat{\phi}_{i_{k\|k}} \delta t \\ 0 & 0 & 1 \end{bmatrix}$
OC-EKF1	$\mathbf{\Phi}'_{i_k} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{i_{k+1\|k}} - \hat{\mathbf{p}}_{i_{k\|k-1}}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$
OC-EKF2	$\mathbf{\Phi}''_{i_k} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\left(\hat{\mathbf{p}}_{i_{k+1\|k}} - \hat{\mathbf{p}}_{i_{k\|k}} + (-1)^i \frac{\boldsymbol{\lambda}_k}{2}\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$

B.4 An example of OC-EKF CL

In the following, we provide a specific CL example to illustrate the implementation of the proposed OC-EKF estimators, in which a team of two robots using the unicycle motion model measure relative distance and bearing to each other. Note that the same models were used in our simulations (see Section 3.6).

Suppose that at the first time-step, $k = 0$, the robot poses are initialized by $\hat{\mathbf{x}}_{i_{0|0}}$ and $\mathbf{P}_{i_{0|0}}$, for $i = 1, 2$. Following the standard practice, we employ the approximation that the velocity and heading are constant during each propagation interval and thus obtain ${}^k\hat{\mathbf{x}}_{i_{k+1}} = [v_{m_{i,k}} \;\; 0 \;\; \omega_{m_{i,k}}]^T \delta t$, where $\mathbf{u}_{m_{i,k}} = [v_{m_{i,k}} \;\; \omega_{m_{i,k}}]^T$ are the linear and rotational velocity measurements, for $i = 1, 2$, respectively, and δt is the sampling period. Substitution into (3.2)-(3.3) yields the following common equations for robot pose propagation:

$$\hat{\mathbf{p}}_{i_{k+1|k}} = \hat{\mathbf{p}}_{i_{k|k}} + \begin{bmatrix} v_{m_{i,k}} \, c\hat{\phi}_{i_{k|k}} \\ v_{m_{i,k}} \, s\hat{\phi}_{i_{k|k}} \end{bmatrix} \delta t \tag{B.12}$$

$$\hat{\phi}_{i_{k+1|k}} = \hat{\phi}_{i_{k|k}} + \omega_{m_{i,k}} \delta t \tag{B.13}$$

Once the propagated states are computed, we now calculate the state-propagation Jacobian matrix in order to propagate the covariance. It is important to note that this calculation depends on the particular filter used, and is one main difference between the four filters (i.e., the standard EKF, the OC-EKF1, the OC-EKF2, and the OC-EKF3) under consideration in this work. Table B.1 summarizes how the state-propagation Jacobian matrix is computed for each estimator. Specifically, in contrast to the standard EKF, the OC-EKF1 requires additional storage

of the last propagated state estimate $\hat{\mathbf{x}}_{k|k-1}$, and the OC-EKF2 requires an additional variable $\boldsymbol{\lambda}_k$, containing a running sum of all previous state corrections [see (3.53)], while the OC-EKF3 computes the state-propagation Jacobian in the same way as for the standard EKF. The noise Jacobian $\mathbf{G}_{i_k}$ with respect to the odometry vector $\mathbf{u}_{i_k}$ for both robots is computed according to (3.7) for all four estimators (by noting that ${}^k\mathbf{x}_{i_{k+1}} = [v_{i_k} \;\; 0 \;\; \omega_{i_k}]^T \delta t$). The measurement equations and the corresponding Jacobians for the distance and bearing measurement model are given by [see (3.11)-(3.13)]:

$$\mathbf{z}_{k+1}^{(ij)} = \begin{bmatrix} \sqrt{(x_{j_{k+1}} - x_{i_{k+1}})^2 + (y_{j_{k+1}} - y_{i_{k+1}})^2} \\ \text{atan2}\left((y_{j_{k+1}} - y_{i_{k+1}}), (x_{j_{k+1}} - x_{i_{k+1}})\right) - \phi_{i_{k+1}} \end{bmatrix} + \mathbf{v}_{k+1}^{(ij)} \tag{B.14}$$

$$\begin{aligned}
\mathbf{H}_{k+1}^{(ij)} &= \begin{bmatrix} \mathbf{H}_{1_{k+1}}^{(ij)} & \mathbf{H}_{2_{k+1}}^{(ij)} \end{bmatrix} \\
&= -(\nabla \mathbf{h}_{k+1}^{(ij)}) \mathbf{A}(\hat{\phi}_{i_{k+1|k}}) \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{j_{k+1|k}} - \hat{\mathbf{p}}_{i_{k+1|k}}) & -\mathbf{I}_2 & 0 \\ \mathbf{0}_{1\times 2} & 1 & 0 & -1 \end{bmatrix} \\
&= -\begin{bmatrix} \frac{\left(\hat{\mathbf{p}}_{j_{k+1|k}}^T - \hat{\mathbf{p}}_{i_{k+1|k}}^T\right)\mathbf{C}(\hat{\phi}_{i_{k+1|k}})}{||\hat{\mathbf{p}}_{j_{k+1|k}} - \hat{\mathbf{p}}_{i_{k+1|k}}||} & 0 \\ \frac{\left(\hat{\mathbf{p}}_{j_{k+1|k}}^T - \hat{\mathbf{p}}_{i_{k+1|k}}^T\right)\mathbf{C}(\hat{\phi}_{i_{k+1|k}})\mathbf{J}^T}{||\hat{\mathbf{p}}_{j_{k+1|k}} - \hat{\mathbf{p}}_{i_{k+1|k}}||^2} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{C}^T(\hat{\phi}_{i_{k+1|k}}) & 0 \\ 0 & 1 \end{bmatrix} \times \\
&\qquad \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{j_{k+1|k}} - \hat{\mathbf{p}}_{i_{k+1|k}}) & -\mathbf{I}_2 & 0 \\ \mathbf{0}_{1\times 2} & 1 & 0 & -1 \end{bmatrix} \\
&= \begin{bmatrix} -\frac{\hat{\mathbf{p}}_{j_{k+1|k}}^T - \hat{\mathbf{p}}_{i_{k+1|k}}^T}{||\hat{\mathbf{p}}_{j_{k+1|k}} - \hat{\mathbf{p}}_{i_{k+1|k}}||} & 0 & \frac{\hat{\mathbf{p}}_{j_{k+1|k}}^T - \hat{\mathbf{p}}_{i_{k+1|k}}^T}{||\hat{\mathbf{p}}_{j_{k+1|k}} - \hat{\mathbf{p}}_{i_{k+1|k}}||} & 0 \\ \frac{\left(\hat{\mathbf{p}}_{j_{k+1|k}}^T - \hat{\mathbf{p}}_{i_{k+1|k}}^T\right)\mathbf{J}}{||\hat{\mathbf{p}}_{j_{k+1|k}} - \hat{\mathbf{p}}_{i_{k+1|k}}||^2} & -1 & -\frac{\left(\hat{\mathbf{p}}_{j_{k+1|k}}^T - \hat{\mathbf{p}}_{i_{k+1|k}}^T\right)\mathbf{J}}{||\hat{\mathbf{p}}_{j_{k+1|k}} - \hat{\mathbf{p}}_{i_{k+1|k}}||^2} & 0 \end{bmatrix}
\end{aligned} \tag{B.15}$$

which hold for $i, j = 1, 2$ and $i \neq j$, and are identical for the standard EKF and the OC-EKF1 and the OC-EKF2. In contrast, the OC-EKF3 obtains the corresponding measurement Jacobian by projecting (B.15) onto the observable subspace [see (3.68)]. Given these expressions, we proceed to use the standard EKF propagation and update equations.

Appendix C

Appendices for Chapter 4

C.1 Proof of Lemma 4.4.1

We start with the linearization error in the linear regression:

$$\mathbf{e}(\mathbf{x}) = \mathbf{y} - (\mathbf{A}\mathbf{x} + \mathbf{b}) = \mathbf{y} - \mathbf{A}_1\mathbf{x}_1 - \mathbf{A}_2\mathbf{x}_2 - \mathbf{b} \tag{C.1}$$

Substituting (C.1) in the expression of the expected value of the squared linearization error (4.1), the cost function we seek to minimize becomes:

$$\begin{aligned} &\int_{-\infty}^{+\infty} ||\mathbf{y} - \mathbf{A}_1\mathbf{x}_1 - \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}||^2 p(\mathbf{x}) d\mathbf{x} \;\; = \\ &\iint_{-\infty}^{+\infty} ||\mathbf{y} - \mathbf{A}_1\mathbf{x}_1 - \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}||^2 p(\mathbf{x}_2|\mathbf{x}_1) p(\mathbf{x}_1) d\mathbf{x}_1 d\mathbf{x}_2 \end{aligned} \tag{C.2}$$

where we have employed the notation $||\boldsymbol{\alpha}||^2 \triangleq \boldsymbol{\alpha}^T\boldsymbol{\alpha}$, and the property $p(\mathbf{x}) = p(\mathbf{x}_1, \mathbf{x}_2) = p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_1)$. Now using the standard sample-based approximation, $p(\mathbf{x}_1) \simeq \sum_{i=0}^{r} w_i \delta(\mathbf{x}_1 - \mathcal{X}_{1_i})$, where the samples are selected to match the mean and covariance of $p(\mathbf{x}_1)$ [see (4.2)

and (4.3)], we rewrite the cost function (C.2) as follows:

$$\begin{aligned} c :=& \sum_{i=0}^{r} w_i \int_{-\infty}^{+\infty} ||\mathcal{Y}_i - \mathbf{A}_1 \mathcal{X}_{1_i} - \mathbf{A}_2 \mathbf{x}_2 - \mathbf{b}||^2 p(\mathbf{x}_2 | \mathbf{x}_1 = \mathcal{X}_{1_i}) d\mathbf{x}_2 \\ =& \sum_{i=0}^{r} w_i \mathbb{E}\left(||\mathcal{Y}_i - \mathbf{A}_1 \mathcal{X}_{1_i} - \mathbf{A}_2 \mathbf{x}_2 - \mathbf{b}||^2 \right) \\ =& \sum_{i=0}^{r} w_i \Big[\left(\mathcal{Y}_i - \mathbf{A}_1 \mathcal{X}_{1_i} - \mathbf{b}\right)^T \left(\mathcal{Y}_i - \mathbf{A}_1 \mathcal{X}_{1_i} - \mathbf{b}\right) - \\ & 2 \left(\mathcal{Y}_i - \mathbf{A}_1 \mathcal{X}_{1_i} - \mathbf{b}\right)^T \mathbf{A}_2 \mathbb{E}(\mathbf{x}_2) \Big] + \mathrm{tr}\left[\mathbf{A}_2 \mathbb{E}(\mathbf{x}_2 \mathbf{x}_2^T) \mathbf{A}_2^T \right] \end{aligned} \tag{C.3}$$

where have used the following identity:

$$\mathbb{E}(\mathbf{x}_2^T \mathbf{A}_2^T \mathbf{A}_2 \mathbf{x}_2) = \mathrm{tr}\left[\mathbf{A}_2 \mathbb{E}(\mathbf{x}_2 \mathbf{x}_2^T) \mathbf{A}_2^T \right] \tag{C.4}$$

Note that the expectation operator $\mathbb{E}(\cdot)$ is with respect to the pdf $p(\mathbf{x}_2 | \mathbf{x}_1 = \mathcal{X}_{1_i})$. For the Gaussian case, this pdf can be expressed analytically as follows:

$$p(\mathbf{x}_2 | \mathbf{x}_1 = \mathcal{X}_{1_i}) = \mathcal{N}\left(\hat{\mathcal{X}}_{2_i}, \bar{\mathbf{P}}_{\hat{\mathbf{x}}_2 \hat{\mathbf{x}}_2} \right) = \tag{C.5}$$

$$\mathcal{N}\left(\underbrace{\hat{\mathbf{x}}_2 + \mathbf{P}_{\mathbf{x}_2 \mathbf{x}_1} \mathbf{P}_{\mathbf{x}_1 \mathbf{x}_1}^{-1} \left(\mathcal{X}_{1_i} - \hat{\mathbf{x}}_1\right)}_{\hat{\mathcal{X}}_{2_i}}, \underbrace{\mathbf{P}_{\mathbf{x}_2 \mathbf{x}_2} - \mathbf{P}_{\mathbf{x}_2 \mathbf{x}_1} \mathbf{P}_{\mathbf{x}_1 \mathbf{x}_1}^{-1} \mathbf{P}_{\mathbf{x}_1 \mathbf{x}_2}}_{\bar{\mathbf{P}}_{\hat{\mathbf{x}}_2 \hat{\mathbf{x}}_2}} \right)$$

Based on (C.5), we have:

$$\mathbb{E}\left(\mathbf{x}_2 \mathbf{x}_2^T \right) = \mathbb{E}\left(\mathbf{x}_2 \right) \mathbb{E}\left(\mathbf{x}_2 \right)^T + \bar{\mathbf{P}}_{\hat{\mathbf{x}}_2 \hat{\mathbf{x}}_2} = \hat{\mathcal{X}}_{2_i} \hat{\mathcal{X}}_{2_i}^T + \bar{\mathbf{P}}_{\hat{\mathbf{x}}_2 \hat{\mathbf{x}}_2} \tag{C.6}$$

Substituting $\mathbb{E}(\mathbf{x}_2) \triangleq \hat{\mathcal{X}}_{2_i}$ from (C.5) and $\mathbb{E}(\mathbf{x}_2 \mathbf{x}_2^T)$ from (C.6) in (C.3) and performing algebraic manipulations, we obtain:

$$c = \sum_{i=0}^{r} w_i ||\mathcal{Y}_i - \mathbf{A}_1 \mathcal{X}_{1_i} - \mathbf{A}_2 \hat{\mathcal{X}}_{2_i} - \mathbf{b}||^2 + \mathrm{tr}\left(\mathbf{A}_2 \bar{\mathbf{P}}_{\hat{\mathbf{x}}_2 \hat{\mathbf{x}}_2} \mathbf{A}_2^T \right) \tag{C.7}$$

Our goal is to minimize the cost function in (C.7) with respect to $\mathbf{b}$ and $\mathbf{A} \triangleq \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{bmatrix}$. To do so, we first compute the optimal solution for $\mathbf{b}$, by setting the derivative of (C.7) with

respect to $\mathbf{b}$ to zero. This yields:

$$\begin{aligned}\frac{\partial c}{\partial \mathbf{b}} &= -2\sum_{i=0}^{r} w_i \left(\mathcal{Y}_i - \mathbf{A}_1\mathcal{X}_{1_i} - \mathbf{A}_2\hat{\mathcal{X}}_{2_i} - \mathbf{b}\right) = \mathbf{0} \;\Rightarrow \\ \mathbf{b} &= \sum_{i=0}^{r} w_i\mathcal{Y}_i - \mathbf{A}_1\sum_{i=0}^{r} w_i\mathcal{X}_{1_i} - \mathbf{A}_2\sum_{i=0}^{r} w_i\hat{\mathcal{X}}_{2_i} \\ &= \bar{\mathbf{y}} - \mathbf{A}_1\hat{\mathbf{x}}_1 - \mathbf{A}_2\hat{\mathbf{x}}_2 \end{aligned} \tag{C.8}$$

where for the last step, we have used (4.6), the equality $\hat{\mathbf{x}}_1 = \sum_{i=0}^{r} w_i\mathcal{X}_{1_i}$, and the identity $\hat{\mathbf{x}}_2 = \sum_{i=0}^{r} w_i\hat{\mathcal{X}}_{2_i}$, which stems from (C.5).

Substituting (C.8) in (C.7), we have:

$$c' = \sum_{i=0}^{r} w_i||\tilde{\mathcal{Y}}_i - \mathbf{A}_1\tilde{\mathcal{X}}_{1_i} - \mathbf{A}_2\tilde{\mathcal{X}}_{2_i}||^2 + \mathrm{tr}\left(\mathbf{A}_2\bar{\mathbf{P}}_{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_2}\mathbf{A}_2^T\right) \tag{C.9}$$

where

$$\tilde{\mathcal{X}}_{1_i} \triangleq \mathcal{X}_{1_i} - \hat{\mathbf{x}}_1 \tag{C.10}$$

$$\tilde{\mathcal{Y}}_i \triangleq \mathcal{Y}_i - \bar{\mathbf{y}} \tag{C.11}$$

$$\tilde{\mathcal{X}}_{2_i} \triangleq \hat{\mathcal{X}}_{2_i} - \hat{\mathbf{x}}_2 = \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\tilde{\mathcal{X}}_{1_i} \tag{C.12}$$

Note that (C.5) was used in (C.12). Taking derivatives of the cost function in (C.9) with respect to $\mathbf{A}_1$ and $\mathbf{A}_2$, and setting them equal to zero, we obtain:

$$\frac{\partial c'}{\partial \mathbf{A}_1} = -2\sum_{i=0}^{r} w_i \left(\tilde{\mathcal{Y}}_i - \mathbf{A}_1\tilde{\mathcal{X}}_{1_i} - \mathbf{A}_2\tilde{\mathcal{X}}_{2_i}\right)\tilde{\mathcal{X}}_{1_i}^T = \mathbf{0} \tag{C.13}$$

$$\frac{\partial c'}{\partial \mathbf{A}_2} = -2\sum_{i=0}^{r} w_i \left(\tilde{\mathcal{Y}}_i - \mathbf{A}_1\tilde{\mathcal{X}}_{1_i} - \mathbf{A}_2\tilde{\mathcal{X}}_{2_i}\right)\tilde{\mathcal{X}}_{2_i}^T + 2\mathbf{A}_2\bar{\mathbf{P}}_{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_2} = \mathbf{0} \tag{C.14}$$

At this point, we use the fact that, due to the selection of the sigma points, we have:

$$\sum_{i=0}^{r} w_i \tilde{\mathcal{Y}}_i \tilde{\mathcal{X}}_{1_i}^T = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1} \tag{C.15}$$

$$\sum_{i=0}^{r} w_i \tilde{\mathcal{X}}_{1_i} \tilde{\mathcal{X}}_{1_i}^T = \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1} \tag{C.16}$$

$$\sum_{i=0}^{r} w_i \tilde{\mathcal{X}}_{2_i} \tilde{\mathcal{X}}_{1_i}^T = \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1} \sum_{i=0}^{r} w_i \tilde{\mathcal{X}}_{1_i} \tilde{\mathcal{X}}_{1_i}^T = \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1} \tag{C.17}$$

$$\begin{aligned} \sum_{i=0}^{r} w_i \tilde{\mathcal{Y}}_i \tilde{\mathcal{X}}_{2_i}^T &= \sum_{i=0}^{r} w_i \tilde{\mathcal{Y}}_i \tilde{\mathcal{X}}_{1_i}^T \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} \\ &= \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} \end{aligned} \tag{C.18}$$

$$\begin{aligned} \sum_{i=0}^{r} w_i \tilde{\mathcal{X}}_{2_i} \tilde{\mathcal{X}}_{2_i}^T &= \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1} \left(\sum_{i=0}^{r} w_i \tilde{\mathcal{X}}_{1_i} \tilde{\mathcal{X}}_{1_i}^T \right) \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} \\ &= \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} \end{aligned} \tag{C.19}$$

where (C.12) was used for deriving these relations. Substituting the above results in (C.13) and (C.14) yields:

$$\bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1} - \mathbf{A}_1 \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1} - \mathbf{A}_2 \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1} = \mathbf{0} \tag{C.20}$$

$$\begin{aligned} \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} - \mathbf{A}_1 \mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} - \\ \mathbf{A}_2 \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} - \mathbf{A}_2 \bar{\mathbf{P}}_{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_2} = \mathbf{0} \end{aligned} \tag{C.21}$$

It is easy to verify (e.g., by substitution) that the solution to the above system of equations (C.20)-(C.21) is $\mathbf{A}_1 = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}$ and $\mathbf{A}_2 = \mathbf{0}$. This completes the proof.

C.2 Proof of Lemma 4.6.1

Using (4.67), we write the equality constraint on $\mathbf{A}_1$ (4.64) as $\mathbf{A}_1 \mathbf{U} = \mathbf{0}$. This equation states that the rows of $\mathbf{A}_1$ lie in the left nullspace of the 5×3 matrix $\mathbf{U}$. Therefore, if $\mathbf{L}$ is a 2×5 matrix whose rows span this nullspace, we can write $\mathbf{A}_1$ as:

$$\mathbf{A}_1 = \mathbf{B}\mathbf{L} \tag{C.22}$$

where $\mathbf{B}$ is an $m \times 2$ unknown matrix that we seek to compute. We note that there are several possible ways of computing an appropriate matrix $\mathbf{L}$, whose rows lie in the nullspace of $\mathbf{U}$. For

instance, such a matrix is given, in closed form, by the expression (4.66). Substituting (C.22) in the original problem formulation [see (4.4) and (4.63)], we obtain:

$$\min_{\mathbf{B},\mathbf{b}} \quad \sum_{i=0}^{10} w_i \left[\mathcal{Z}_i - (\mathbf{B}\mathcal{X}_i' + \mathbf{b})\right]^T \left[\mathcal{Z}_i - (\mathbf{B}\mathcal{X}_i' + \mathbf{b})\right] \tag{C.23}$$

where we have defined $\mathcal{X}_i' \triangleq \mathbf{L}\mathcal{X}_i$, $i = 0, \ldots, 10$. This becomes an unconstrained minimization problem with respect to the design variables $\mathbf{B}$ and $\mathbf{b}$, and has exactly the same structure as that in (4.4). Thus, by analogy, the optimal solution of $\mathbf{B}$ is computed by [see (4.5)]:

$$\mathbf{B} = \bar{\mathbf{P}}_{\mathbf{z}\boldsymbol{\ell}} \mathbf{P}_{\boldsymbol{\ell}\boldsymbol{\ell}}^{-1} \tag{C.24}$$

where

$$\bar{\mathbf{P}}_{\mathbf{z}\boldsymbol{\ell}} = \sum_{i=0}^{10} w_i (\mathcal{Z}_i - \bar{\mathbf{z}})(\mathbf{L}\mathcal{X}_i - \mathbf{L}\bar{\mathbf{x}})^T = \bar{\mathbf{P}}_{\mathbf{zx}} \mathbf{L}^T$$

$$\mathbf{P}_{\boldsymbol{\ell}\boldsymbol{\ell}} = \sum_{i=0}^{10} w_i (\mathbf{L}\mathcal{X}_i - \mathbf{L}\bar{\mathbf{x}})(\mathbf{L}\mathcal{X}_i - \mathbf{L}\bar{\mathbf{x}})^T = \mathbf{L}\bar{\mathbf{P}}_{\mathbf{xx}} \mathbf{L}^T$$

By combining these two identities with those of (C.24) and (C.22), we obtain the optimal solution of $\mathbf{A}_1$ [see (4.65)].

Appendix D

Appendices for Chapter 6

D.1 Proof of Lemma 6.2.1

We prove this result by mathematical induction. Specifically, we start by the base case of $k = 0$, in which since no marginalization is involved and the observability matrix is simply the first measurement Jacobian, i.e., $\mathbf{M} = \mathbf{H}_0$ [see (6.10)], the information matrix can be directly written in the desired form:

$$\mathbf{A} = \mathbf{A}_0 = \mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{H}_0 =: \mathbf{M}^T \mathbf{\Sigma} \mathbf{M} \tag{D.1}$$

We now consider the case of $k = 1$, in which the state vector is $\mathbf{x}_{0:1} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix}$ and hence the full information matrix is given by [see (6.18)]:

$$\begin{aligned} \mathbf{A} &= \mathcal{F}_0^T \mathbf{Q}_0^{-1} \mathcal{F}_0 + \mathcal{H}_0^T \mathbf{R}_0^{-1} \mathcal{H}_0 + \mathcal{H}_1^T \mathbf{R}_1^{-1} \mathcal{H}_1 \\ &= \begin{bmatrix} \mathbf{\Phi}_0^T \mathbf{Q}_0^{-1} \mathbf{\Phi}_0 + \mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{H}_0 & -\mathbf{\Phi}_0^T \mathbf{Q}_0^{-1} \\ -\mathbf{Q}_0^{-1} \mathbf{\Phi}_0 & \mathbf{Q}_0^{-1} + \mathbf{H}_1^T \mathbf{R}_1^{-1} \mathbf{H}_1 \end{bmatrix} \end{aligned} \tag{D.2}$$

In order to obtain the information matrix of $\mathbf{x}_0$, we marginalize out $\mathbf{x}_1$ by employing the Schur

complement:

$$\begin{aligned}\mathbf{A}_0 &= \boldsymbol{\Phi}_0^T\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0 + \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0 - \left(\boldsymbol{\Phi}_0^T\mathbf{Q}_0^{-1}\right)\underbrace{\left(\mathbf{Q}_0^{-1}+\mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1\right)^{-1}}_{\boldsymbol{\Xi}_1}\left(\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0\right)\\ &= \boldsymbol{\Phi}_0^T\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0 + \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0 - \left(\boldsymbol{\Phi}_0^T\mathbf{Q}_0^{-1}\right)\left[\mathbf{Q}_0 - \mathbf{Q}_0\mathbf{H}_1^T(\mathbf{H}_1\mathbf{Q}_0\mathbf{H}_1^T+\mathbf{R}_1)^{-1}\mathbf{H}_1\mathbf{Q}_0\right]\left(\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0\right)\\ &= \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0 + \left(\mathbf{H}_1\boldsymbol{\Phi}_0\right)^T\left(\mathbf{H}_1\mathbf{Q}_0\mathbf{H}_1^T+\mathbf{R}_1\right)^{-1}\left(\mathbf{H}_1\boldsymbol{\Phi}_0\right)\\ &= \begin{bmatrix}\mathbf{H}_0\\ \mathbf{H}_1\boldsymbol{\Phi}_0\end{bmatrix}^T\begin{bmatrix}\mathbf{R}_0^{-1} & \mathbf{0}\\ \mathbf{0} & (\mathbf{H}_1\mathbf{Q}_0\mathbf{H}_1^T+\mathbf{R}_1)^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{H}_0\\ \mathbf{H}_1\boldsymbol{\Phi}_0\end{bmatrix} =: \mathbf{M}^T\boldsymbol{\Sigma}\mathbf{M}\end{aligned} \tag{D.3}$$

where we have used the Woodbury matrix identity [49] for computing $\boldsymbol{\Xi}_1$. It becomes clear that in this case the information matrix of $\mathbf{x}_0$ is factorized into the desired form.

To better understand the structure, we consider one more base case of $k = 2$, where the full information matrix of the entire state $\mathbf{x}_{0:2}$ is given by [see (6.18)]:

$$\begin{aligned}\mathbf{A} &= \sum_{\kappa=0}^{1}\mathcal{F}_\kappa^T\mathbf{Q}_\kappa^{-1}\mathcal{F}_\kappa + \sum_{\kappa=0}^{2}\mathcal{H}_\kappa^T\mathbf{R}_\kappa^{-1}\mathcal{H}_\kappa\\ &= \begin{bmatrix}\boldsymbol{\Phi}_0^T\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0 + \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0 & -\boldsymbol{\Phi}_0^T\mathbf{Q}_0^{-1} & \mathbf{0}\\ -\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0 & \mathbf{Q}_0^{-1}+\boldsymbol{\Phi}_1^T\mathbf{Q}_1^{-1}\boldsymbol{\Phi}_1+\mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1 & -\boldsymbol{\Phi}_1^T\mathbf{Q}_1^{-1}\\ \mathbf{0} & -\mathbf{Q}_1^{-1}\boldsymbol{\Phi}_1 & \mathbf{Q}_1^{-1}+\mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{H}_2\end{bmatrix}\end{aligned} \tag{D.4}$$

Similarly, by marginalizing out $\mathbf{x}_1$ and $\mathbf{x}_2$ using the Schur complement, we obtain the information matrix of the initial state $\mathbf{x}_0$:

$$\begin{aligned}\mathbf{A}_0 &= \boldsymbol{\Phi}_0^T\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0 + \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0\\ &\quad - \begin{bmatrix}\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0\\ \mathbf{0}\end{bmatrix}^T\underbrace{\begin{bmatrix}\mathbf{Q}_0^{-1}+\boldsymbol{\Phi}_1^T\mathbf{Q}_1^{-1}\boldsymbol{\Phi}_1+\mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1 & -\boldsymbol{\Phi}_1^T\mathbf{Q}_1^{-1}\\ -\mathbf{Q}_1^{-1}\boldsymbol{\Phi}_1 & \mathbf{Q}_1^{-1}+\mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{H}_2\end{bmatrix}^{-1}}_{\boldsymbol{\Xi}_2}\begin{bmatrix}\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0\\ \mathbf{0}\end{bmatrix}\end{aligned} \tag{D.5}$$

It is clear that due to the structure of the above equation, we only need to compute the top-leftmost submatrix $\boldsymbol{\Xi}_2(1,1)$, obtained by partitioning $\boldsymbol{\Xi}_2 = \begin{bmatrix}\boldsymbol{\Xi}_2(1,1) & \boldsymbol{\Xi}_2(1,2)\\ \boldsymbol{\Xi}_2(2,1) & \boldsymbol{\Xi}_2(2,2)\end{bmatrix}$, corresponding to the nonzero block of $\begin{bmatrix}\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0\\ \mathbf{0}\end{bmatrix}$. Hence, using the block matrix inversion lemma [49],

we have:

$$\begin{aligned}
\boldsymbol{\Xi}_2(1,1) &= \left[\mathbf{Q}_0^{-1} + \boldsymbol{\Phi}_1^T\mathbf{Q}_1^{-1}\boldsymbol{\Phi}_1 + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1 - \boldsymbol{\Phi}_1^T\mathbf{Q}_1^{-1}\left(\mathbf{Q}_1^{-1} + \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{H}_2\right)^{-1}\mathbf{Q}_1^{-1}\boldsymbol{\Phi}_1^T\right]^{-1} \\
&= \left[\mathbf{Q}_0^{-1} + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1 + \boldsymbol{\Phi}_1^T\mathbf{H}_2^T(\mathbf{H}_2\mathbf{Q}_2\mathbf{H}_2^T + \mathbf{R}_2)^{-1}\mathbf{H}_2\boldsymbol{\Phi}_1\right]^{-1} \\
&= \left\{\mathbf{Q}_0^{-1} + \begin{bmatrix}\mathbf{H}_1 \\ \mathbf{H}_2\boldsymbol{\Phi}_1\end{bmatrix}^T \begin{bmatrix}\mathbf{R}_1^{-1} & \mathbf{0} \\ \mathbf{0} & (\mathbf{H}_2\mathbf{Q}_2\mathbf{H}_2^T + \mathbf{R}_2)^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{H}_1 \\ \mathbf{H}_2\boldsymbol{\Phi}_1\end{bmatrix}\right\}^{-1} \\
&=: \left[\mathbf{Q}_0^{-1} + \mathbf{H}_{1:2}^T\mathbf{R}_{1:2}^{-1}\mathbf{H}_{1:2}\right]^{-1} \\
&= \mathbf{Q}_0 - \mathbf{Q}_0\mathbf{H}_{1:2}^T(\mathbf{H}_{1:2}\mathbf{Q}_0\mathbf{H}_{1:2}^T + \mathbf{R}_{1:2})^{-1}\mathbf{H}_{1:2}\mathbf{Q}_0
\end{aligned} \tag{D.6}$$

where we have defined $\mathbf{H}_{1:2} \triangleq \begin{bmatrix}\mathbf{H}_1 \\ \mathbf{H}_2\boldsymbol{\Phi}_1\end{bmatrix}$ and $\mathbf{R}_{1:2} \triangleq \text{Diag}\left(\mathbf{R}_1, \mathbf{H}_2\mathbf{Q}_2\mathbf{H}_2^T + \mathbf{R}_2\right)$. It is important to notice that, from the first equality of the above equations, $\boldsymbol{\Xi}_2^{-1}(1,1)$ is the Schur complement of $\boldsymbol{\Xi}_2^{-1}$ with respect to $\mathbf{x}_1$ (i.e., marginalizing out $\mathbf{x}_2$ from $\mathbf{x}_{1:2}$). In the second equality, we have also employed the Woodbury matrix identity. Substitution of (D.6) in (D.5) yields (by noting again that it is not necessary to compute the other submatrices of $\boldsymbol{\Xi}_2$):

$$\begin{aligned}
\mathbf{A}_0 &= \boldsymbol{\Phi}_0^T\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0 + \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0 - \boldsymbol{\Phi}_0^T\mathbf{Q}_0^{-1}\left[\mathbf{Q}_0 - \mathbf{Q}_0\mathbf{H}_{1:2}^T(\mathbf{H}_{1:2}\mathbf{Q}_0\mathbf{H}_{1:2}^T + \mathbf{R}_{1:2})^{-1}\mathbf{H}_{1:2}\mathbf{Q}_0\right]\mathbf{Q}_0^{-1}\boldsymbol{\Phi}_0 \\
&= \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0 + \boldsymbol{\Phi}_0^T\mathbf{H}_{1:2}^T(\mathbf{H}_{1:2}\mathbf{Q}_0\mathbf{H}_{1:2}^T + \mathbf{R}_{1:2})^{-1}\mathbf{H}_{1:2}\boldsymbol{\Phi}_0 \\
&= \begin{bmatrix}\mathbf{H}_0 \\ \mathbf{H}_{1:2}\boldsymbol{\Phi}_0\end{bmatrix}^T \begin{bmatrix}\mathbf{R}_0^{-1} & \mathbf{0} \\ \mathbf{0} & (\mathbf{H}_{1:2}\mathbf{Q}_0\mathbf{H}_{1:2}^T + \mathbf{R}_{1:2})^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{H}_0 \\ \mathbf{H}_{1:2}\boldsymbol{\Phi}_0\end{bmatrix} =: \mathbf{M}^T\boldsymbol{\Sigma}\mathbf{M}
\end{aligned} \tag{D.7}$$

It is clear that in this case we can also factorize the information matrix of $\mathbf{x}_0$ into the desired form.

We now consider the general case of $k = \kappa$. Suppose in the case of $k = \kappa - 1$, the information matrix of the initial state $\mathbf{x}_0$ can be factorized into the desired form and in particular $\boldsymbol{\Xi}_{\kappa-1}(1,1)$ whose inversion is the Schur complement of $\boldsymbol{\Xi}_{\kappa-1}^{-1}$ with respect to $\mathbf{x}_1$ by marginalizing out $\mathbf{x}_{2:\kappa-1}$ from $\mathbf{x}_{1:\kappa-1}$, assumes the following form [see (D.6)]:

$$\begin{aligned}
\boldsymbol{\Xi}_{\kappa-1}(1,1) &= \left(\mathbf{Q}_0^{-1} + \mathbf{H}_{1:\kappa-1}^T\mathbf{R}_{1:\kappa-1}^{-1}\mathbf{H}_{1:\kappa-1}\right)^{-1} \\
&= \mathbf{Q}_0 - \mathbf{Q}_0\mathbf{H}_{1:\kappa-1}^T\left(\mathbf{H}_{1:\kappa-1}\mathbf{Q}_0\mathbf{H}_{1:\kappa-1}^T + \mathbf{R}_{1:\kappa-1}\right)^{-1}\mathbf{R}_{1:\kappa-1}\mathbf{Q}_0
\end{aligned} \tag{D.8}$$

where $\mathbf{H}_{1:\kappa-1}$ and $\mathbf{R}_{1:\kappa-1}$ are defined similarly as in (D.6):

$$\mathbf{H}_{1:\kappa-1} \triangleq \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_{\kappa-1}\mathbf{\Phi}_{\kappa-2}\cdots\mathbf{\Phi}_1 \end{bmatrix} \tag{D.9}$$

$$\mathbf{R}_{1:\kappa-1} \triangleq \mathrm{Diag}\left(\mathbf{R}_1, \cdots, \mathbf{H}_{\kappa-1}\mathbf{Q}_{\kappa-1}\mathbf{H}_{\kappa-1}^T + \mathbf{R}_{\kappa-1}\right) \tag{D.10}$$

Note that by considering the full state $\mathbf{x}_{2:\kappa}$ (instead of $\mathbf{x}_{1:\kappa-1}$) and marginalizing $\mathbf{x}_{3:\kappa}$ from the full state, we will have a similar matrix as (D.8) whose time index is shifted by one and which is useful for the ensuing derivations.

$$\begin{aligned} \mathbf{\Xi}'_\kappa(1,1) &= \left(\mathbf{Q}_1^{-1} + \mathbf{H}_{2:\kappa}^T\mathbf{R}_{2:\kappa}^{-1}\mathbf{H}_{2:\kappa}\right)^{-1} \\ &= \mathbf{Q}_1 - \mathbf{Q}_1\mathbf{H}_{2:\kappa}^T\left(\mathbf{H}_{2:\kappa}\mathbf{Q}_1\mathbf{H}_{2:\kappa}^T + \mathbf{R}_{2:\kappa}\right)^{-1}\mathbf{R}_{2:\kappa}\mathbf{Q}_1 \end{aligned} \tag{D.11}$$

Due to the spare banded structure of the full information matrix (6.18), marginalization of $\mathbf{x}_{1:\kappa}$ from the full state $\mathbf{x}_{0:\kappa}$ using the Schur complement yields:

$$\mathbf{A}_0 = \mathbf{\Phi}_0^T\mathbf{Q}_0^{-1}\mathbf{\Phi}_0 + \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0 - \begin{bmatrix} \mathbf{Q}_0^{-1}\mathbf{\Phi}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}^T \mathbf{\Xi}_\kappa \begin{bmatrix} \mathbf{Q}_0^{-1}\mathbf{\Phi}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \tag{D.12}$$

As evident, to compute $\mathbf{A}_0$, we only need to calculate the top leftmost submatrix, $\mathbf{\Xi}_\kappa(1,1)$, corresponding to $\mathbf{Q}_0^{-1}\mathbf{\Phi}_0$. Note that in analogy to (D.8), $\mathbf{\Xi}_\kappa^{-1}(1,1)$ is the Schur complement of $\mathbf{\Xi}_\kappa^{-1}$ with respect to $\mathbf{x}_1$ by marginalizing out $\mathbf{x}_{2:\kappa}$ from $\mathbf{x}_{1:\kappa}$. Using (6.18) and (D.11), we compute $\mathbf{\Xi}_\kappa(1,1)$ as:

$$\begin{aligned} \mathbf{\Xi}_\kappa(1,1) &= \left[\mathbf{Q}_0^{-1} + \mathbf{\Phi}_1^T\mathbf{Q}_1^{-1}\mathbf{\Phi}_1 + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1 - \mathbf{\Phi}_1^T\mathbf{Q}_1^{-1}\mathbf{\Xi}'_\kappa(1,1)\mathbf{Q}_1^{-1}\mathbf{\Phi}_1\right]^{-1} \\ &= \left[\mathbf{Q}_0^{-1} + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1 + \mathbf{\Phi}_1^T\mathbf{H}_{2:\kappa}^T\left(\mathbf{H}_{2:\kappa}\mathbf{Q}_1\mathbf{H}_{2:\kappa}^T + \mathbf{R}_{2:\kappa}\right)^{-1}\mathbf{H}_{2:\kappa}\mathbf{\Phi}_1\right]^{-1} \\ &= \left\{\mathbf{Q}_0^{-1} + \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_{2:\kappa}\mathbf{\Phi}_1 \end{bmatrix}^T \begin{bmatrix} \mathbf{R}_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{2:\kappa}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_{2:\kappa}\mathbf{\Phi}_1 \end{bmatrix}\right\}^{-1} \\ &=: \left(\mathbf{Q}_0^{-1} + \mathbf{H}_{1:\kappa}\mathbf{R}_{1:\kappa}^{-1}\mathbf{H}_{1:\kappa}\right)^{-1} \\ &= \mathbf{Q}_0 - \mathbf{Q}_0\mathbf{H}_{1:\kappa}^T\left(\mathbf{H}_{1:\kappa}\mathbf{Q}_0\mathbf{H}_{1:\kappa}^T + \mathbf{R}_{1:\kappa}\right)^{-1}\mathbf{R}_{1:\kappa}\mathbf{Q}_0 \end{aligned} \tag{D.13}$$

Substituting (D.13) in (D.12), we obtain the the information matrix of $\mathbf{x}_0$ in the desired form:

$$\begin{aligned}
\mathbf{A}_0 &= \boldsymbol{\Phi}_0^T \mathbf{Q}_0^{-1} \boldsymbol{\Phi}_0 + \mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{H}_0 \\
&\quad - \boldsymbol{\Phi}_0^T \mathbf{Q}_0^{-1} \left[\mathbf{Q}_0 - \mathbf{Q}_0 \mathbf{H}_{1:\kappa}^T \left(\mathbf{H}_{1:\kappa} \mathbf{Q}_0 \mathbf{H}_{1:\kappa}^T + \mathbf{R}_{1:\kappa} \right)^{-1} \mathbf{R}_{1:\kappa} \mathbf{Q}_0 \right] \mathbf{Q}_0^{-1} \boldsymbol{\Phi}_0 \\
&= \mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{H}_0 + \boldsymbol{\Phi}_0^T \mathbf{H}_{1:\kappa}^T \left(\mathbf{H}_{1:\kappa} \mathbf{Q}_0 \mathbf{H}_{1:\kappa}^T + \mathbf{R}_{1:\kappa} \right)^{-1} \mathbf{H}_{1:\kappa} \boldsymbol{\Phi}_0 \\
&= \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_{1:\kappa} \boldsymbol{\Phi}_0 \end{bmatrix}^T \begin{bmatrix} \mathbf{R}_0^{-1} & \mathbf{0} \\ \mathbf{0} & \left(\mathbf{H}_{1:\kappa} \mathbf{Q}_0 \mathbf{H}_{1:\kappa}^T + \mathbf{R}_{1:\kappa} \right)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_{1:\kappa} \boldsymbol{\Phi}_0 \end{bmatrix} \\
&=: \mathbf{M}^T \boldsymbol{\Sigma} \mathbf{M}
\end{aligned} \tag{D.14}$$

This completes the proof.

Appendix E

Appendices for Chapter 7

E.1 Proof of Lemma 7.4.1

We first note that the following finite dimensional Mountain Pass Theorem (MPT) will be useful for the ensuing analysis.

Theorem E.1.1. [Theorem 5.2, [81]] *Suppose that a continuous function $f \in \mathcal{C}^1(\mathbb{R}^N;\mathbb{R})$ is coercive and possesses two distinct strict relative minima $\mathbf{x}_1$ and $\mathbf{x}_2$.*[1] *Then f possesses a third critical point $\mathbf{x}_3$, which is distinct from $\mathbf{x}_1$ and $\mathbf{x}_2$, and characterized by:*

$$f(\mathbf{x}_3) = \inf_{\Sigma\in\Gamma} \max_{\mathbf{x}\in\Sigma} f(\mathbf{x}) \tag{E.1}$$

where $\Gamma = \{\Sigma \subset \mathbb{R}^N; \Sigma$ is compact and connected, and $\mathbf{x}_1, \mathbf{x}_2 \in \Sigma\}$. Moreover, $\mathbf{x}_3$ is not a relative minimizer; that is, in every neighborhood of $\mathbf{x}_3$, there exists a point $\mathbf{x}$ such that $f(\mathbf{x}) < f(\mathbf{x}_3)$.[2]

To preserve the clarity of presentation, without loss of generality, we hereafter translate the global frame of reference to the sensor's local frame (i.e., $\mathbf{p}_S = \mathbf{0}$), and also drop the time indices as well as the subscript " T " denoting target. Thus, the minimization of (7.20) can be

[1] *A function $f : \mathbb{R}^N \to \mathbb{R}$ is coercive iff it is bounded from below and is proper in the sense that $f(\mathbf{x}) \to \infty$ for $||\mathbf{x}|| \to \infty$.*

[2] *Γ is the class of paths (or curves) connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ (see Theorem 1.1, Ch. II, [151]). Note that, based on the proof of this theorem (see [81]), such a critical point $\mathbf{x}_3$* (E.1) *also exists for $f \in \mathcal{C}^0(\mathbb{R}^N;\mathbb{R})$, though, in this case, $\mathbf{x}_3$ is not necessary to be a non-minimum point. It is also important to notice that, an isolated inf-max critical point (i.e., mountain pass point), $\mathbf{x}_3$, is necessarily a saddle point (see Ch. 12, [81]).*

written as:

$$\min_{\mathbf{p}} c(\mathbf{p}) = \frac{1}{2}(\mathbf{p}-\hat{\mathbf{p}})^T \mathbf{P}_{\mathbf{pp}}^{-1}(\mathbf{p}-\hat{\mathbf{p}}) + \frac{1}{2\sigma_\rho^2}\left(z - \sqrt{x^2+y^2}\right)^2$$
$$\triangleq c_1(\mathbf{p}) + c_2(\mathbf{p}) \tag{E.2}$$

Recall that $\mathbf{p} = \begin{bmatrix} x & y \end{bmatrix}^T$ denotes the target position. In (E.2), given a function value $c(x,y) \triangleq \alpha$, $c_1(x,y) \triangleq \beta$ is an ellipse centered at $\hat{\mathbf{p}} = \begin{bmatrix} \hat{x} & \hat{y} \end{bmatrix}^T$, while depending on the available measurement value, $c_2(x,y) = \alpha - \beta \triangleq \gamma$ represents one or two circles centered at the origin, i.e., $x^2+y^2 = (z \pm \sigma_\rho\sqrt{2\gamma})^2$. For simplicity of analysis, in the following, we assume only one circle is associated with $c_2(x,y)$, while the analysis readily holds for two circles. Notice that the cost function $c(\mathbf{p})$ (E.2) is coercive but *not* $\mathcal{C}^1$ since it is not differentiable at the origin. Hence Theorem E.1.1 is not applicable directly.

Nevertheless, in what follows, we will first show that the inf-max point cannot be the origin. Based on that, we will use the MPT to show that there are at most 3 local minima, since $c(\mathbf{p})$ is coercive *and* $\mathcal{C}^1$ in $\mathbb{R}^2\backslash\{\mathbf{0}\}$. Subsequently, we will prove that there are at most 2 local minima.

E.1.1 Proof that there are at most 3 local minima

Remember that there are up to 4 distinct critical (stationary) points in $\mathbb{R}^2\backslash\{\mathbf{0}\}$ (see Section 7.4.2). Suppose that they are all strict local minima, denoted by $\mathbf{p}_i$, $i = 1,\ldots,4$. By the MPT (Theorem E.1.1) in $\mathbb{R}^2\backslash\{\mathbf{0}\}$, we have an inf-max point $\mathbf{p}$ for any pair of $\mathbf{p}_i$. If this $\mathbf{p}$ is different from any $\mathbf{p}_i$, we will have (at least) 5 critical points, which contradicts the fact that there are at most 4 critical points in $\mathbb{R}^2\backslash\{\mathbf{0}\}$. Therefore, any pair of $\mathbf{p}_i$ must share a common inf-max point at the origin which is a non-differential critical point. However, we now show by contradiction that this is not the case.

Specifically, suppose that the origin, $\mathbf{p} = \mathbf{0}$, is the desired common inf-max point. From (E.1), we know that $c(\mathbf{p}_i) < c(\mathbf{0}) \triangleq \alpha$, $i = 1,\ldots,4$. By continuity of $c(\cdot)$, we define the following level set:

$$\mathcal{S} = c^{-1}(\alpha) = \{\mathbf{p} \in \mathbb{R}^2 : c(\mathbf{p}) = c(\mathbf{0})\}$$

We now show some important properties of this level set $\mathcal{S}$ that will be useful for our proof. First of all, from the implicit function theorem (see Lemma 4 of Ch. 2 in [114]), $\mathcal{S}\backslash(0,0)$ is a smooth 1-dimensional manifold. A smooth connected 1-dimensional manifold is diffeomorphic

either to a circle or to some interval of real numbers (see Appendix of [114]).[3] Thus, from differential topology [114], a 1-dimensional manifold and thus $\mathcal{S}\backslash(0,0)$ is a *union of disjoint smooth curves (lines and circles)*. This also implies that $\mathcal{S}\backslash(0,0)$ has no isolated points.

Secondly, $\mathcal{S}\backslash(0,0)$ has *no closed* curves. To see this, by contradiction, suppose that one curve component of $\mathcal{S}\backslash(0,0)$ is closed, denoted by $\mathcal{S}_1$. By continuity of $c(\cdot)$ and compactness of $\mathcal{S}_1$, and based on the Weierstrass theorem [138], there will exist (at least) one local minimum inside $\mathcal{S}_1$, which has to be one of the points $\mathbf{p}_i$ since no other critical point exists. Note that such a local minimum cannot be on the boundary of $\mathcal{S}_1$ since $c(\mathbf{p}_i) < \alpha$. Without loss of generality, we assume $\mathbf{p}_1$ is the only local minimum inside $\mathcal{S}_1$. Note that there cannot be more than one local minima inside $\mathcal{S}_1$ (see the next paragraph below). Based on continuity of $c(\cdot)$ and compactness of $\mathcal{S}_1$, we can find a sufficiently small $\epsilon > 0$ so that $\mathcal{S}_1$ is contained in one closed component of $c^{-1}(\alpha+\epsilon)$, which is disjoint from $\mathcal{S}\backslash\mathcal{S}_1$ as they have different function values ($\alpha+\epsilon$ and α, respectively). Clearly, in this case, the inf-max value between $\mathbf{p}_1$ and $\mathbf{p}_i$ ($i \neq 1$) will be larger than α and is attained at a point other than $(0,0)$, which gives the contradiction. Therefore, all curves of $\mathcal{S}\backslash(0,0)$ are open-ended. Moreover, by continuity, $c(\cdot)$ attains α at any limit point (i.e., the open-end point) of the curves of $\mathcal{S}\backslash(0,0)$. However, $(0,0)$ is the only point where $c(\cdot)$ attains α except the curves of $\mathcal{S}\backslash(0,0)$, and thus the *only* common limit of all open-ended curves of $\mathcal{S}\backslash(0,0)$.

Lastly, let us consider the properties of the interior of $\mathcal{S}$. We first define the following set $\mathbf{\Theta}$ whose boundary is $\mathcal{S}$, i.e., $\mathbf{\Theta} = \{\mathbf{p} \in \mathbb{R}^2 : c(\mathbf{p}) < \alpha\} = \cup_j \mathbf{\Theta}_j$, where $\mathbf{\Theta}_j$ is j-th component of $\mathbf{\Theta}$. It is clear that $\mathbf{p}_i$ ($i = 1, \ldots, 4$) is contained in $\mathbf{\Theta}_j$ for some j, since $c(\mathbf{p}_i) < \alpha$. Note that, $\mathbf{\Theta}_j$ cannot contain (more than) two $\mathbf{p}_i$, since, otherwise, the inf-max point of these two $\mathbf{p}_i$ will be different from $(0,0)$ and will have a function value less than α. On the other hand, if $\mathbf{\Theta}_j$ does not contain any $\mathbf{p}_i$, similarly, based on the Weierstrass theorem [138], there will exist a new local minimum in the corresponding closure, $\bar{\mathbf{\Theta}}_j = \mathbf{\Theta}_j \cup \partial\mathbf{\Theta}_j$. This contradicts the fact that we have only 4 critical points different from $(0,0)$. Therefore, there are exactly 4 connected components $\mathbf{\Theta}_i$, each of which contains one $\mathbf{p}_i$. Furthermore, by continuity of $c(\cdot)$, the i-th boundary, $\partial\mathbf{\Theta}_i \subset \mathcal{S}$, contains at least one curve of the connected components of $\mathcal{S}\backslash(0,0)$. Since $(0,0)$ is a limit point of all curves, $(0,0) \in \partial\mathbf{\Theta}_i$ for every i, and thus,

$$\partial\mathbf{\Theta}_i = \cup_{j=1}^{n_i} \mathcal{S}_j \cup \{(0,0)\}$$

[3] A map $f : x \to y$ is called a *diffeomorphism* if f carries x homeomorphically onto y and if both f and f^{-1} are smooth.

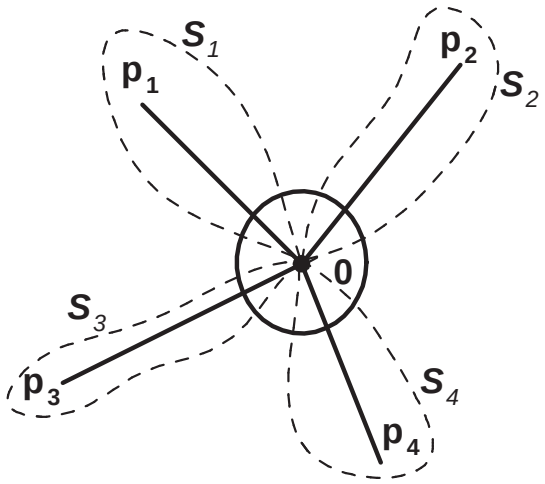


Figure E.1: Illustration of the topological configuration of four local minima, $\mathbf{p}_i$ ($i = 1, \ldots, 4$), and the level set $\mathcal{S}$. In this plot, the dashed curves represent the level set $\mathcal{S}$, and the solid lines are the paths which connects the local minima $\mathbf{p}_i$ and $\mathbf{p}_j$ and attains its maximum at the origin. Note that each of the paths is contained exactly in one interior of the connected components of $\mathcal{S}$. It is clear that the circle $c_2(\mathbf{p})$ intersects $\mathcal{S}$ at 8 different points.

where $\mathcal{S}_j$ is one curve of $\mathcal{S}\backslash(0,0)$ and n_i is the number of $\mathcal{S}_j$ contained in $\partial\mathbf{\Theta}_i$. Note that any two $\partial\mathbf{\Theta}_i$ do not share a common curve, since if they do, by definition, the shared curve will consist of (infinitely many) critical points. As a result, there are at least 4 distinct open curves in $\mathcal{S}\backslash(0,0)$ having $(0,0)$ as a common limit point (see Fig. E.1).

As seen from Fig. E.1, for a given measurement z, the circle $c_2(\mathbf{p})$ intersects the level set $\mathcal{S}$ at (at least) 8 intersection points denoted by $\mathbf{q}_i$, $i = 1, \ldots, 8$. We can show that all these 8 points belong to the same ellipse $c_1(\mathbf{p})$, by noting that

$$\begin{aligned} c(\mathbf{q}_i) =& c_1(\mathbf{q}_i) + c_2(\mathbf{q}_i) \\ \Rightarrow\ c_1(\mathbf{q}_i) =& c(\mathbf{q}_i) - c_2(\mathbf{q}_i) \triangleq \alpha - \gamma \triangleq \beta \end{aligned} \tag{E.3}$$

This indicates that the 8 intersection points, $\mathbf{q}_i$, belong to the same ellipse $c_1(\mathbf{p}) = \beta$. This results in 8 intersection points between the ellipse c_1 and the circle c_2. However, it is known from geometry that there are at most 4 intersection points between any ellipse and circle. Therefore, we conclude that $(0,0)$ cannot be an inf-max point.

By the MPT, there must exist an inf-max point between any two local minima among the 4 stationary points. Therefore, there are at most 3 local minima (by sharing one stationary point

as the common inf-max point).

E.1.2 Proof that there are at most 2 local minima

We now prove that there are at most 2 local minima by showing that the assumption of 3 local minima (i.e., $\mathbf{p}_i$, $i = 1, 2, 3$) leads to a contradiction. Denote the inf-max point, $\mathbf{m}_i$, corresponding to the two local minima, $\mathbf{p}_i$ and $\mathbf{p}_j$ ($i, j = 1, 2, 3$ and $i \neq j$),

$$c(\mathbf{m}_i) = \inf_{\Sigma_{ij} \in \Gamma} \max_{\mathbf{p} \in \Sigma_{ij}} c(\mathbf{p})$$

Recall that we have at most 5 critical points in total (i.e., 1 non-differentiable point at the origin and 4 stationary points). So, it is clear that there are only four possible cases that we need to examine in terms of $\mathbf{m}_i$ ($i = 1, 2, 3$):

- Case I: $\mathbf{m}_1 = \mathbf{m}_2 \neq \mathbf{m}_3 = \mathbf{0}$
- Case II: $\mathbf{m}_1 \neq \mathbf{m}_2 = \mathbf{m}_3 = \mathbf{0}$
- Case III: $\mathbf{m}_1 = \mathbf{m}_2 = \mathbf{m}_3 = \mathbf{0}$
- Case IV: $\mathbf{m}_1 = \mathbf{m}_2 = \mathbf{m}_3 \neq \mathbf{0}$

Our goal is to prove that all these four cases are impossible to occur and thus there are at most 2 local minima. In what follows, we first show that the first three cases with a zero inf-max point (i.e., Cases I, II and III) cannot occur, and then disprove Case IV.

We start by considering a special case where the prior estimate coincides with the sensor position (i.e., $\hat{\mathbf{p}} = \mathbf{0}$). By denoting $\mathbf{P}_{\mathbf{pp}}^{-\mathbf{1}} = \begin{bmatrix} s_1 & s_3 \\ s_3 & s_2 \end{bmatrix}$, we expand the cost function (E.2) in a

neighborhood of the origin as follows:

$$
\begin{aligned}
c(x,y) &= \frac{1}{2}\begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} s_1 & s_3 \\ s_3 & s_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \frac{1}{2\sigma_\rho^2}\left(z - \sqrt{x^2+y^2}\right)^2 \\
&= \left(\frac{s_1\sigma_\rho^2+1}{2\sigma_\rho^2}\right)x^2 + \left(\frac{s_2\sigma_\rho^2+1}{2\sigma_\rho^2}\right)y^2 + s_3xy - \frac{z}{\sigma_\rho^2}\sqrt{x^2+y^2} + \frac{z^2}{2\sigma_\rho^2} \\
&\triangleq Ax^2 + By^2 + Cxy - E\sqrt{x^2+y^2} + D \qquad \text{(E.4)} \\
&= \sqrt{x^2+y^2}\left(\frac{Ax^2+By^2+Cxy}{\sqrt{x^2+y^2}} - E\right) + D \\
&\leq \sqrt{x^2+y^2}\,(A|x| + B|y| + |C||x| - E) + D \qquad \text{(E.5)}
\end{aligned}
$$

where $E \triangleq \frac{z}{\sigma_\rho^2} > 0$, due to the positive distance measurement z. In the above expressions, we have employed the inequalities $|x| \leq \sqrt{x^2+y^2}$ and $|y| \leq \sqrt{x^2+y^2}$. Clearly, there exists a neighborhood of $(0,0)$ such that $A|x|+B|y|+|C||x|-E < 0$ and hence $c(x,y) < D = c(0,0)$, if $(x,y) \neq (0,0)$. By definition, $(0,0)$ becomes a local maximum and thus cannot be an inf-max point.[4] This is the contradiction, and therefore Cases I, II, and III cannot happen when $\hat{\mathbf{p}} = \mathbf{0}$.

Now consider the general case where $\hat{\mathbf{p}} \neq \mathbf{0}$. First, in Case I ($\mathbf{m}_1 = \mathbf{m}_2 \neq \mathbf{m}_3 = \mathbf{0}$), as in the previous proof of at most 3 local minima (see Section E.1.1), we can show that there are at least 4 intersection points between the circle $c_2(\mathbf{p})$ for a given z and the level set $\mathcal{S}$ (see Fig. E.2). If the circle c_2 collapses to a single point (i.e., with zero radius), clearly there is only 1 intersection point between the circle c_2 and the level set $\mathcal{S}$, and hence 1 intersection point between the circle c_2 and the ellipse c_1.[5] Importantly, by continuity of c_2 and compactness of $\mathcal{S}$, if perturbing c_2 by an arbitrarily small number in the neighborhood of origin, there are always at least 4 intersection points between the circle c_2 and the level set $\mathcal{S}$, and thus at least 4 intersection points between the circle c_2 and the ellipse c_1 [see (E.3)]. This perturbation results in the dynamics of the number of intersection points between the circle and the ellipse, changing from 1 to 4. However, this is not the case, since we know from geometry that by continuity of the circle and ellipse, if applying a small perturbation on the circle, 1 intersection point between the circle and ellipse can only dynamically change 0, 1 or 2 (instead of 4) intersection points.

[4] Note that if $z = 0$, then (E.4) becomes quadratic and has a unique global minimum at the origin (by noting that $A > 0$), which clearly contradicts the assumption of three local minima.

[5] Although depending on the measurement z, another circle possibly exists and thus may result in more intersection points between the circle c_2 and the ellipse c_1, we here consider the dynamics of the intersection point in a *neighborhood* of the origin.

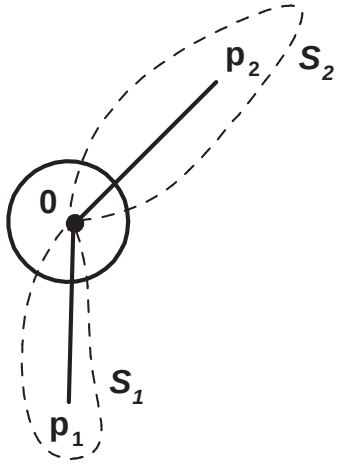


Figure E.2: Illustration of the circle $c_2(\mathbf{p}) = \alpha$ intersecting the level set $\mathcal{S}$ as well as the ellipse c_1 by a small perturbation on c_2. In this plot, the dashed curves represent the level set, the solid lines are the path which connects the local minima $\mathbf{p}_1$ and $\mathbf{p}_2$ and attains its maximum at the origin, and the solid circle represents c_2 after perturbation.

Moreover, as compared to Case I, in Case II there will be at least 6 (instead of 4) intersection points between the circle $c_2(\mathbf{p})$ for a given z and the level set $\mathcal{S}$, and thus we can show in a similar way that this case is also impossible.

In Case III ($\mathbf{m}_1 = \mathbf{m}_2 = \mathbf{m}_3 = \mathbf{0}$), interestingly, no matter whether $\hat{\mathbf{p}} = \mathbf{0}$ or $\hat{\mathbf{p}} \neq \mathbf{0}$, proceeding similarly as in Cases I and II and as in Section E.1.1, we can derive the contradiction that there will be 6 intersection points between the circle c_2 and the level set $\mathcal{S}$ (and thus the ellipse c_1), and hence show that this case is also impossible to occur.

At this point, we have ruled out Cases I-III. We will now disprove Case IV. Specifically, to simplify notations, we denote the common inf-max point by $\mathbf{m} \triangleq \mathbf{m}_1 = \mathbf{m}_2 = \mathbf{m}_3$. Then, we first consider a special case where the prior estimate coincides with the inf-max point, i.e., $\hat{\mathbf{p}} = \mathbf{m}$. In this case, we know that $\nabla c(\mathbf{p}) \mid_{\mathbf{p}=\mathbf{m}} = \mathbf{0}$, since $\mathbf{p} = \mathbf{m}$ is a critical (inf-max) point. Therefore, we have:

$$\begin{aligned} \mathbf{0} = \nabla c(\mathbf{p}) \mid_{\mathbf{p}=\mathbf{m}} &= \nabla c_1(\mathbf{p}) \mid_{\mathbf{p}=\mathbf{m}} + \nabla c_2(\mathbf{p}) \mid_{\mathbf{p}=\mathbf{m}} \\ &= \frac{1}{\sigma_\rho^2}(z - ||\mathbf{m}||)\frac{\mathbf{m}}{||\mathbf{m}||} \;\; \Rightarrow \;\; z = ||\mathbf{m}|| \;\; \Rightarrow \;\; c(\mathbf{m}) = 0 \end{aligned}$$

which clearly shows that $\mathbf{p} = \mathbf{m}$ is a global minimum (by noting the quadratic and nonnegative cost function (E.2)). This contradicts the assumption that $\mathbf{p} = \mathbf{m}$ is an inf-max point.

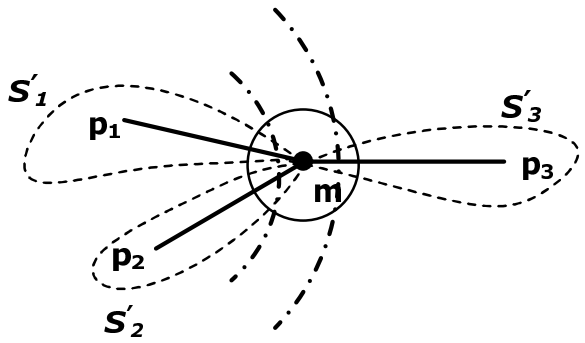


Figure E.3: Illustration of the circle $c_2(\mathbf{p}) = \alpha'$ intersecting the level set $\mathcal{S}'$ by a small perturbation on c_2. In this plot, the dashed curves represent the level set, the solid lines are the path which connects the local minima $\mathbf{p}_i$ and $\mathbf{p}_j$ and attains its maximum at $\mathbf{p} = \mathbf{m}$, the dash-dotted arcs (of the circles) represent c_2 before and after perturbation, and the solid circle denotes the neighborhood of $\mathbf{p} = \mathbf{m}$.

We finally consider the general scenario of Case IV where $\hat{\mathbf{p}} \neq \mathbf{m}$. We define the following level set:

$$\mathcal{S}' = \{\mathbf{p} \in \mathbb{R}^2 : c(\mathbf{p}) = c(\mathbf{m}) \triangleq \alpha'\}$$

Similar to Case III and Section E.1.1 (where instead the level set $\mathcal{S}$ is considered), the corresponding interior, $\mathbf{\Theta'} = \{\mathbf{p} \in \mathbb{R}^2 : c(\mathbf{p}) < \alpha'\}$, has three curve components, each of which contains exactly one $\mathbf{p}_i$ (see Fig. E.3). Consider the scenario where $\mathbf{p} = \mathbf{m}$ is an intersection point between the circle and the ellipse (by noting $c(\mathbf{m}) = c_1(\mathbf{m}) + c_2(\mathbf{m})$). In a neighborhood of $\mathbf{p} = \mathbf{m}$, if perturbing the circle $c_2(\mathbf{p}) = c_2(\mathbf{m}) \triangleq \gamma'$ by an arbitrarily small (positive or negative) value, the circle will either shrink or expand. Since there are 6 branches belonging to the level set $\mathcal{S}'$, by continuity and compactness of c_2 and $\mathcal{S}'$, in the *neighborhood* of $\mathbf{p} = \mathbf{m}$, there will exist at least 3 intersection points between the circle and the level set, and thus at least 3 intersection points between the circle and the ellipse [see (E.3)], either when the circle shrinks or expands. Thus, there exists an (arbitrarily) small perturbation on c_2 so that the number of intersection points between the circle and the ellipse, in the neighborhood of $\mathbf{p} = \mathbf{m}$, dynamically changes from 1 to (more than) 3. However, we know from geometry that this is not the case since 1 intersection point between a circle and an ellipse, by perturbation, can change to at most 2 in its *neighborhood*.

Thus far, we have proven that all four cases are impossible to occur if there are 3 local minima. As a result, there are at most 2 local minima. This completes the proof.