UNIVERSITY OF CALIFORNIA

Los Angeles

# Robotic Spatial Autonomy: Multirobot Localization and Online SLAM

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical and Computer Engineering

by

Tsang-Kai Chang

2022

# ABSTRACT OF THE DISSERTATION

Robotic Spatial Autonomy: Multirobot Localization and Online SLAM

by

Tsang-Kai Chang
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2022
Professor Ankur M. Mehta, Chair

Spatial autonomy enables autonomous agents to interact with the environment intelligently and smoothly. This dissertation considers two particular scenarios to realize and improve the spatial autonomy for robots. In the first scenario, the cooperation among multiple robots to localize themselves is studied. I propose a new multirobot localization algorithm with observation and communication steps separated. This algorithm uses far less communication than other algorithms, which improves the efficiency and robustness. I furthermore develop a framework to optimize the observation and communication rates of the algorithm. I also study an online solution for simultaneous localization and mapping (SLAM). Current SLAM algorithms solve the trajectory and the map via an optimization problem, especially after loop closure. However, these algorithms are offline, whose computational cost increases significantly with time. Instead, we formulate the SLAM problem as an inference problem over a hidden Markov model, and this structure allows an online implementation. Therefore, the comparable accuracy can be achieved with far less computation. Both scenarios show more efficient spatial autonomy algorithms for robots when the information usage is carefully designed, and pave a way for robots with stronger autonomy.

The dissertation of Tsang-Kai Chang is approved.

Paulo Tabuada

Mayank R. Mehta

Jonathan Chau-Yan Kao

Ankur M. Mehta, Committee Chair

University of California, Los Angeles

2022

*To my parents*

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# VITA

2012 B.S. (Electrical Engineering), National Taiwan University.

2014 M.S. (Communication Engineering), National Taiwan University.

2020 Finalist of Qualcomm Innovation Fellowship

2016–present Ph.D. student, Electrical and Computer Engineering Department, UCLA.

# CHAPTER 1

# Introduction

One of the major robot potentials is to navigate and explore the world. The applications rely on this potential will also be ubiquitous: We are about to witness the road packed with autonomous driving vehicles. Robots are sent to deep oceans, volcanos, and even the outer space to explore the unknown. A swarm of drones will be able to completing tasks collaboratively [YBD18].

The essence of navigation and exploration is an accurate and efficient spatial autonomy capability. The spatial autonomy includes that the robot can build a map of the environment as well as locate itself in this map. The map and its own spatial state together enable the robot to interact with the environment. Since this problem is fundamental, this problem has been investigated as simultaneous localization and mapping, or SLAM, in robotics for long.

To fully characterize the spatial autonomy, I divide it into three aspects: sensing, algorithm, and representation. The first aspect regards the data collected from the sensors to realize spatial autonomy. With the advanced of optical and MEMS technology, there are now a variety of options of sensors for autonomous robots, including laser, LIDAR, IMUs, etc. The raw data are usually classified into proprioceptive and exteroceptive: the former one encodes the robot dynamic, while the later one relates the robot and the environment. The second aspect focuses on how algorithms can realize the spatial autonomy with the available information. In addition to simply realizing spatial autonomy, we would also expect these algorithms to be accurate, efficient, and robust in face of all the imperfections in the reality. The last aspect characterizes how the outputs of the spatial autonomy algorithms are stored.

This is particularly important from a systematic perspective. The representation should allow rapid update, since the robot is mobile and the information keeps accumulating. The representation should also be accessible, for various tasks like navigation, path planning, etc.

Animals serve a vivid example to understand spatial autonomy. For mammals particularly, including human beings, the visual system and the vestibular system provide exteroceptive and proprioceptive sensing data, respectively. These two types of sensing data together construct our spatial understanding of the environment. Also, neuroscientists have discovered that the position is encoded in place cells and grid cells, while these two types of neurons have different firing patterns [OD71, HFM05]. The mechanism (or the algorithm) that takes sensing data and produces all these cell firing patterns is not fully understood yet. While the brain is such an integrated and complicated computer, it is difficult to precisely assess the contribution of each factor. Researchers devise novel techniques and experiments to uncover the algorithm from the sensing to the representation. For example, researchers use virtual reality to determine the effect of two types of information [RKW13, CLK19].

The focus of this dissertation is on designing spatial autonomy algorithms. Even though we now have high-performance SLAM algorithms that are about to be ready to be applied in our daily lives, there are still fundamental challenges in some critical scenarios. I investigated the excessive information exchange in multirobot systems, and the offline nature of SLAM algorithms. In both scenarios, I am interested in improving the efficiency of information usage to realize spatial autonomy, in which we can either lower the communication cost or lower the information requirement.

## 1.1 Multirobot Localization Algorithm

The first scenario in this dissertation is the localization in multirobot systems, where multiple robots try to determine their spatial states collaboratively. Multirobot systems have better robustness and scalability compared to the single robot systems for various tasks, but acquiring their spatial states remains the premise of all high-level tasks. Multirobot systems introduce more opportunities for localization than a single robot. First, if the spatial state of other robots is known, observing other robots is similar to observe a landmark, which could improve the localization performance. Second, robots can share their information and spatial states via wireless communication, which might also improve the overall performance. However, designing a multirobot localization algorithm has to take the distributed nature into account to ensure its efficiency and resilience. The traditional multirobot localization algorithm, which is extended from a single-robot localization algorithm, requires extensive communication, which poses vulnerability even with a single communication failure.

Before delving into the algorithm design, I first study how informations are combined in a distributed system [FCM20]. While the localization is commonly treated as a filtering problem, combining spatial states is equivalent to fusing probability distributions. The fusion of probability distributions can be generally formulated as finding a barycenter of several distributions. With different problem formulation, the resulting fused distribution has different properties. In particular, we are interested in the Kullback-Leibler (KL) barycenter, which ensures that the fused distribution is conservative. A special case of KL barycenter is the covariance intersection when only consider Gaussian distributions are considered. We also extend the KL barycenter to von-Mises Fisher distributions, which are used for circular data. We then apply this fusion scheme together with the von-Mises Fisher filter to study the estimation and the fusion of non-Euclidean data.

Equipped with covariance intersection, I developed a multirobot localization algorithm that ensures the estimation consistency and also separates the communication and the observation [CCM20, CCM22]. This separation not only reduces the required communication, but also enhances the resilience of the localization algorithm in the face of communication failures and adversaries. We compare the algorithm with other multirobot localization algorithms, and evaluate them with the UTIAS dataset. The result shows the accuracy as well as the resilience of our algorithm.

Since our multirobot localization system uses covariance intersection, the communication update may introduce unnecessary uncertainty to keep the estimate conservative. Therefore, I further investigate the optimal scheduling of this algorithm [CM18b]. I take both the cost of operations as well as the contribution to the estimation into account, and propose a method to determine the optimal operation rate according to different goals.

## 1.2 Online SLAM

The other scenario I studied in this dissertation is the SLAM problem. I particularly consider a system with only a camera and a IMU sensor, which is often called a visual-inertial system. Common SLAM algorithms for visual-inertial systems usually consist of two parts: frontend reconstructs the local trajectory, while the backend part takes care of the global consistency, especially after loop closures are detected. There are various methods for the frontend, but the backend is mainly treated as a nonlinear least squares (NLS) optimization problem. In a technical term, the frontend will produce several constraints regarding the spatial states and the map, and the backend finds the best spatial states and map from a numerical optimization procedure.

The NLS formulation gives a straightforward understanding for SLAM backend, but there is an implementation concern. The NLS optimization problem is an offline problem, which means that the scale of the problem increases with the number of the data. For robotic systems, the data keep arriving, and the cost of the backend optimization problem keeps increasing and eventually becomes very large. Now, the interesting question is how can we design an online SLAM backend, which will be more efficient than the offline optimization counterpart. In other words, it is a question about how can we discard information while maintaining the estimation accuracy.

To fundamentally come up with a solution, I go back to the original problem formulation, which is the state transition model that describe the state dynamics and the observability.

Instead of regarding the SLAM problem as a NLS optimization problem, we can leverage the intrinsic hidden Markov model (HMM) structure of spatial dynamics and model the SLAM problem as an inference problem. The inference problem with hidden variables is typically solved by the EM algorithm, and thus we can design an online SLAM algorithm with the online EM algorithm for HMMs. Due to the dependency of the HMM, the block size of this algorithm should keep increasing. Therefore, we arrive at the block online EM (BOEM) SLAM algorithm. We verify the BOEM SLAM on the EuRoC datasets, and show the efficiency of the BOEM algorithm. To realize an end-to-end SLAM system building on the BOEM algorithm, as other state-of-the-art SLAM algorithms are already mature now, more work should be done.

# CHAPTER 2

# Barycenter of Distributions

In robotic systems, we often use probability distributions to represent informations. For example, we usually use Gaussian distributions, where the mean represents our knowledge while the covariance matrix represents the uncertainty. When we consider multiagent systems, we will encounter the problem of combining information from several agents. Conceptually, it is similar to find a representative "average" of all distributions, which is also known as the "barycenter".

To motivate the general barycenter formulation, we begin with a simple weighted average example. For an Euclidean space $\mathbb{R}^d$, the weighted average $x_c$ of points $x_1, \ldots, x_N \in \mathbb{R}^d$ with positive weights $w_1, \ldots, w_N$ summing to 1 is simply $x_c = \sum_{n=1}^{N} w_n x_n$. In fact, $x_c$ is the solution of the following optimization problem

$$x_c = \underset{x \in \mathbb{R}^d}{\arg\min} \sum_{n=1}^{N} w_n (x - x_n)^2. \tag{2.1}$$

Let $\mathbb{S}^{d-1}$ denote the $d$-dimensional unit hypersphere, i.e., $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d : x^\mathsf{T} x = 1\}$. Similarly, if we consider circular terms $\theta_1, \ldots, \theta_N \in \mathbb{S}^1$ with the circular distance between two terms as $d(\theta, \theta') = 1 - \cos(\theta - \theta')$, we can formulate an optimization problem to find the barycenter of these circular terms. In particular,

$$\theta_c = \underset{\theta \in \mathbb{S}^1}{\arg\min} \sum_{n=1}^{N} w_n (1 - \cos(\theta - \theta_n)). \tag{2.2}$$

We get

$$\exp(i\theta_c) = \sum_{n=1}^{N} w_n \exp(i\theta_n). \tag{2.3}$$

These examples show that finding the barycenter can be formulated as an optimization problem, as long as we determine a proper distance measure. By formulating an optimization problem, the barycenter problem becomes more tangible since we can now use optimization tools. In this chapter, we will extend the barycenter formulation to probability distributions, discuss the properties of the results, and derive the Kullback-Leibler barycenter for both Gaussian distribution and von Mises-Fisher (vMF) distributions.

## 2.1 Kullback-Leibler Barycenter

The Kullback-Leibler (KL) divergence quantifies the difference between two distributions. For distributions $P$ and $Q$ of a continuous random variable, the KL divergence is defined as

$$D_{KL}(P||Q) = \int p(x) \log \frac{p(x)}{q(x)} \, dx, \tag{2.4}$$

where $p$ and $q$ denotes the probability densities of $P$ and $Q$. With a slight abuse of distance notation, the corresponding weighted average with KL divergence for probability density functions $p_1, \ldots, p_N$ and weights $w_1, \ldots, w_N$ can then be defined as

$$p_c = \arg\inf_{p \in \mathcal{P}} \sum_{n=1}^{N} w_n D_{KL}(p||p_n), \tag{2.5}$$

where $\mathcal{P}$ is a set of probability density functions.[1] We will call $p_c$ the *KL average* of $p_1, \ldots, p_N$ with weights $w_1, \ldots, w_N$.

When using probability densities to represent information, the KL average can be interpreted as a conservative fusion of the original distributions $p_1, \ldots, p_N$. In [BJA12], an axiomatic definition of conservative approximation is proposed, and the KL average satisfies this definition. From their analysis, the KL average does not double count common information, and does not set the distribution to 0 if one of the source distributions is non-zero.

[1] Due to the asymmetry of KL divergence, another KL barycenter can be defined by switching its two arguments, which is discussed in [GBC20]. Basically, the first KL average in (2.5) has the conservative properties, and it is also the most commonly adopted method. We therefore only focus on the first KL average, and just call it the KL barycenter.

In [BC14], the authors use the interpretation that the KL divergence in (2.4) quantifies the information gain achieved when moving from a prior density $q(x)$ to a posterior density $p(x)$. Therefore, the KL average is the one that minimizes the sum of the information gains from the initial densities. We will demonstrate the conservative nature in the following two particular distributions.

## 2.2 Covariance Intersection

The covariance intersection (CI) is actually the KL barycenter of Gaussian distributions.

**Theorem 1.** *Given $N$ multivariable Gaussian distributions $p_1, \dots, p_N$ with the mean $\mu_1, \dots, \mu_N$ and the covariance matrices $\Sigma_1, \dots, \Sigma_N$, together with positive weights $w_1, \dots, w_N$ summing to 1, the mean $\mu_c$ and covariance matrix $\Sigma_c$ of the KL barycenter distribution satisfy*

$$\Sigma_c^{-1} = \sum_{n=1}^{N} w_n \Sigma_n^{-1}, \tag{2.6}$$

$$\Sigma_c^{-1} \mu_c = \sum_{n=1}^{N} w_n \Sigma_n^{-1} \mu_n. \tag{2.7}$$

The proof can be found in [BC14].

In fact, CI is first developed in the data fusion context, and is later recognized as a KL barycenter. In the data fusion setting, we use Gaussian random vectors as estimates, and can then substantiate the previous discussion on conservativeness with concrete examples. A consistent estimate can be seen as a conservative estimate regarding the estimation uncertainty intuitively. In other words, a conservative estimate reports larger uncertainty than the estimate really provides, so as to avoid over-confidence data fusion. When the covariance matrix represents the uncertainty of the estimate, a consistent estimate can be considered as an estimate that has larger covariance matrix in the positive definite sense.

The aforementioned over-confidence problem and the double counting problem can be avoided if estimation consistency is maintained. Formally, a consistent estimate is meanpreserving and has no smaller covariance matrix in the positive definite sense, with the following definition:

**Definition 1** (Estimation consistency). *An estimate* $\hat{\mathbf{z}}$ *of a real vector* $z$ *is a Gaussian random vector with mean* $\mathsf{E}[\hat{\mathbf{z}}]$ *and covariance* $\Sigma_{\hat{\mathbf{z}}}$*. The estimation* $\hat{\mathbf{z}}'$ *of* $z$ *is called* consistent *of* $\hat{\mathbf{z}}$ *if* $\mathsf{E}[\hat{\mathbf{z}}'] = \mathsf{E}[\hat{\mathbf{z}}]$ *and* $\Sigma_{\hat{\mathbf{z}}'} \geq \Sigma_{\hat{\mathbf{z}}}$*.*

**Lemma 1** (Covariance intersection [JU97, CAM02, Sun04, RNA15]). *Given* $N$ *consistent estimates* $\hat{\mathbf{z}}_n$ *of* $\hat{\mathbf{z}}$ *with covariances* $\Sigma_n$ *for* $n = 1, \dots, N$*, the estimate* $\hat{\mathbf{z}}'$ *is also consistent of* $\hat{\mathbf{z}}$ *where* $\hat{\mathbf{z}}'$ *is obtained by CI.*

CI is able to combine several consistent estimates which might be correlated, and the result stays consistent.

## 2.3 KL Barycenter of von Mises-Fisher Distributions

In addition to Gaussian distribution, we extend the KL barycenter to non-Gaussian cases by considering von Mises-Fisher (vMF) distributions. VMF distributions reside on hyperspheres, and are often used to model the orientation in robotics and sensor networks. The barycenter of vMF distribution naturally arises in sensor fusion and data aggregation [SRA14]. The vMF distribution is also common in machine learning. In practice, it is common to normalize data to remove the length information leaving only the directional information. The vMF distribution is then often used to model the distribution of directional data. Several methods are then proposed to process directional data, including [BDG05, HBM17, DFC18, CYG20].

The vMF distribution for $x \in \mathbb{S}^{d-1}$ is given by

$$p_d(x; \mu, \kappa) = C_d(\kappa) \exp(\kappa \mu^\mathsf{T} x). \tag{2.8}$$

The vMF distribution depends on two parameters: the mean direction $\mu \in \mathbb{S}^{d-1}$ and the concentration parameter $\kappa > 0$. The normalization constant is given by

$$C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)}, \tag{2.9}$$

where $I_\alpha(\kappa)$ denotes the modified Bessel function of the first kind.

The KL barycenter of vMF distributions can be formulated as the solution of following problem:

$$p_c = \arg\inf_{p \in \mathcal{P}_{vMF}} \sum_{n=1}^{N} w_n D_{KL}(p||p_n), \tag{2.10}$$

where $\mathcal{P}_{vMF}$ is the set of all vMF distributions and $p_1, \ldots, p_N \in \mathcal{P}_{vMF}$.

**Theorem 2.** *Given $N$ vMF distributions $p_1, \ldots, p_N$ with the mean directions $\mu_1, \ldots, \mu_N$ and the concentration parameters $\kappa_1, \ldots, \kappa_N$, together with positive weights $w_1, \ldots, w_N$ summing to $1$, the mean direction $\mu_c$ and concentration parameter $\kappa_c$ of the KL barycenter distribution satisfy*

$$\kappa_c \mu_c = \sum_{n=1}^{N} w_n \kappa_n \mu_n. \tag{2.11}$$

The proof can be found in [FCM20]. In summary, by using Lagrange multipliers, the optimization problem (2.10) can be formulated in $\mathbb{R}^n$, which can then be solved analytically. The equation (2.11) shows the computation of the KL barycenter is actually very easy. Since $\mu_c$ is a unit vector, $\kappa_c \mu_c$ in (2.11) can be viewed as a vector in $\mathbb{R}^d$. Based on Theorem 2, the KL barycenter of vMF distributions can be obtained conveniently by averaging vectors in $\mathbb{R}^d$.

We visualize the KL barycenter of 3 vMF distribution with detail in Table 2.1. Following the result in Theorem 2, we obtain the KL barycenter. With the configuration in Table 2.1, the mean direction and the concentration parameter of the resulting distribution is $[0.139, 0.928, 0.345]^\intercal$ and 12.39, respectively. In Fig. 2.1, we can see that the distribution of the KL barycenter represents the "average" of the three distributions. Moreover, the

Table 2.1: Parameters of KL barycenter simulation

| Distribution | $\mu^{\mathsf{T}}$ | $\kappa$ | Weight $w_n$ |
|---|---|---|---|
| 1 | $[0, 0, 1]$ | 20 | 0.3 |
| 2 | $[1/\sqrt{3}, 1/\sqrt{3}, -1/\sqrt{3}]$ | 25 | 0.4 |
| 3 | $[-0.5, 1/\sqrt{2}, 0.5]$ | 27 | 0.3 |

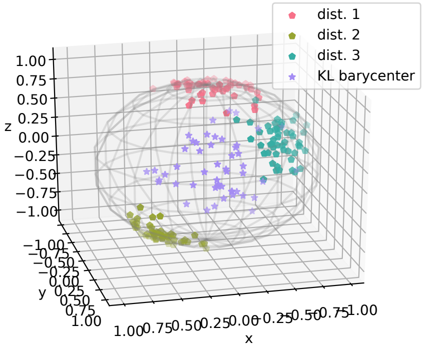


Figure 2.1: The KL barycenter of 3 distributions. Each vMF distribution is sample with 50 points. The KL barycenter, calculated by Theorem 1, visually resides at the average of the 3 distributions.

concentration parameter $\kappa$ of the KL barycenter is smaller than the 3 fused distributions, due to the conservative fusion nature of the KL barycenter discussed.

### 2.3.1 von Mises Filters with Overlapping Sensors

To demonstrate that the proposed KL barycenter produces conservative results, we consider a scenario with 2 estimators. These two estimators try to estimate a dynamic circular value $\boldsymbol{\theta}_t \in \mathbb{S}^1$ that evolves according to

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + u_t + \mathbf{w}_t, \tag{2.12}$$

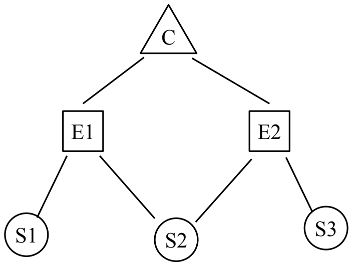


Figure 2.2: The topology of the fusion between 2 dependent von Mises filters. S1, S2, and S3 are the sensors that observe $\boldsymbol{\theta}_t$ by (2.13). E1 and E2 are the von Mises filters. The von Mises filters E1 and E2 operate the time and the observation updates for 20 times, then send its own estimate to the fusion center C for the final fused result. For the KL average fusion, the weights are fixed as $w_1 = 0.6$ and $w_2 = 0.4$.

where $u_t$ is the input and $w_t$ is the process noise modeled by $vM(0, \kappa_w)$. These two estimators do not directly observe $\theta_t$, but receive the observation data from distributed sensors, which observe $\theta_t$ by

$$\mathbf{o}_{k,t} = \boldsymbol{\theta}_t + \boldsymbol{\nu}_{k,t}, \tag{2.13}$$

where $o_{k,t}$ is the observation from sensor $k$. In (2.13), $\nu_{k,t}$ is the observation noise of sensor $k$, and is modeled by $vM(0, \kappa_{\nu,k})$. All process noises and observation noises are independent of the rest of the system.

We consider a system with 2 estimators and 3 sensors, with the system topology in Fig. 2.2. We choose a constant input $u_t = 0.7$ with $\kappa_w = 7$. For the sensor parameters, $\kappa_{\nu,1} = 3.3$, $\kappa_{\nu,2} = 4.4$, and $\kappa_{\nu,3} = 2.2$. The two estimators E1 and E2 use the von Mises filters to dynamically estimate $\theta_t$, but the estimates are dependent since both estimators use the observation from S2. These two estimators run 20 iterations of both the time update and the observation update, and then send their own estimate to the fusion center C.

We compare various fusion protocols at the fusion center C, including our KL average fusion protocol and the fusion equation assuming independence from [ARC09], in Fig. 2.3. As a benchmark, we also plot the optimal fusion where the estimator can directly obtain

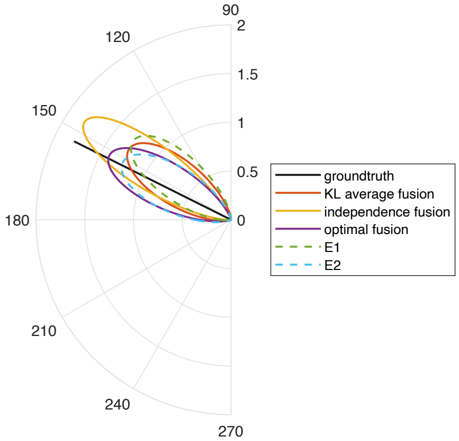


Figure 2.3: The density function of vMF distributions under different fusion methods. The independence fusion produces over-confident estimates, since its concentration parameter is larger than that of the optimal fusion. On the contrary, the KL barycenter gives reasonable estimation without even knowing the exact independent observation data, which shows its efficiency in distributed networks.

the raw observation data, and fuse them. However, the optimal fusion is not practical in distributed networks.

In Fig. 2.3, the concentration parameter of the independence fusion is larger than that of the optimal fusion. Therefore, the independence fusion is definitely over-confident, which effects the estimation reliability. We further execute the identical simulation for 200 trials, and summarize the statistics of the concentration parameters of these three fusion methods in Table 2.2. The concentration parameters from the independence fusion are significantly higher than those of the optimal fusion. On the contrary, the result from the derived KL average gives reasonable concentration parameter. In Table 2.2, the concentration parameters of the KL average are smaller than those of the optimal fusion, since the proposed KL average fusion is a conservative fusion scheme. Moreover, no independence is required in the derived KL average fusion, which shows its efficiency in distributed networks.

Table 2.2: The statistics of the concentration parameters $\kappa$ under different fusion methods over 200 trials

| Fusion method | Mean | Standard deviation |
|---|---|---|
| Optimal fusion | 13.467 | 1.210 |
| KL average fusion | 11.042 | 0.771 |
| Independence fusion [ARC09] | 21.825 | 1.491 |

# CHAPTER 3

# Multirobot Localization I: Algorithm and Resilience

Localization is one of the most fundamental elements for spatial autonomy. As multiple robots form a team to improve robustness and scalability, localization of a multirobot system is therefore a premise to the successful deployment of such system to achieve high-level goals. Compared to single robot localization, there are two additional sources that enable multiple robots to localize themselves cooperatively. First, a robot can observe other robots and the relative observation between them can enhance the overall localization performance. Second, robots can share their information with one another, which can also improve the overall localization performance. Among all approaches of cooperative localization, we focus on the EKF-based approaches primarily due to its computational efficiency.

While cooperative localization takes advantage of relative observation and inter-robot communication, designing a cooperative localization algorithm has its own difficulties. By considering localization as an estimation problem, maintaining estimation consistency remains challenging for multirobot systems. Intuitively, as a correlation implies the dependency between two estimates, if these two estimates are fused with underestimated correlation, the resulting estimate no longer accounts for the estimation uncertainty and the *over-confidence problem* occurs. An extreme example is to regard two repetitive datapoints as two independent information in data fusion, and therefore some researchers also refer this problem as the *double counting problem.*

In order to keep each inter-robot correlation updated, communication is often extensively performed, but excessive communication poses resilience concern. In the seminal cooperative localization algorithm [RB02], an all-to-all communication is required after every observation in order to maintain correlations equivalent to a centralized EKF. Tracking correlations in a distributed system not only requires extensive communications, but it also makes the system vulnerable to even a single communication failure. The following works attempt to decrease the amount of communication either by introducing additional server in the distributed system [KHG18], or by sacrificing the estimation consistency [LNY13, LSR18]. For either improvement, communication is regarded as a supplementary step in an observation update, whichtakes place right after a relative observation. However, such association strongly relies on the assumption that communication is available whenever needed. In short, in most cooperative localization algorithms, communication between robots is either excessive or is implicitly assumed to be always available and free from failure, which makes these algorithms less resilient.

To maintain estimation consistency and to ensure communication resilience, we separate the communication step from the observation step in the proposed algorithm by using the covariance intersection (CI) fusion technique [JU97, CAM02, Sun04, RNA15]. In this algorithm, the estimation consistency is directly assured by CI. Since the communication step is explicit in the proposed algorithm, communication is no longer a complementary part after observation updates but rather an independent source contributing further information. Therefore, the proposed algorithm does not need excessive communications, and communication unavailability will not affect our algorithm's observation update which enhances the resilience.

In addition to the proposal of the algorithm, we provide a performance analysis on the proposed algorithm. By interpreting the proposed algorithm as a distributed estimation algorithm, we investigate the covariance boundedness criterion to assure an upper bound on localization performance. To address the nature of multiagent system, the analysis takes the configuration of observation and communication into account with graph description.

## 3.1 Related Work

The concept of cooperative localization is first proposed in [KNH94], and the term "cooperative localization" is later coined in [RDM98]. The work [KH00] extends the techniques in [KNH94] to an experimental setting. The cooperative localization is also developed in a team of small robots [GK01] to globally localize the team. While these algorithms are able to use the relative observation and the inter-robot communication, they are limited to particular system settings. In the early stage of the cooperative localization development, these algorithms establish the basis for current cooperative localization algorithms that are more fundamental and general.

Depending on the underlying framework, we classify current cooperative localization algorithms into three categories: EKF-based approaches, particle filter-based approaches, and optimization-based approaches. In this section, we highlight the advantages of each algorithm as well their limitations, with specific focus on estimation consistency and communication resilience.

### 3.1.1 Extended Kalman Filter-based Approaches

EKF-based approaches are the mainstream for cooperative localization algorithms. The benchmark of the EKF-based approach is established in the seminal paper [RB02], together with its theoretical analyses in [MR06,HTM11]. This cooperative localization algorithm emphasizes the importance of correlations between inter-robot estimates, and is fundamentally free from the over-confidence problem. In particular, this algorithm is exactly a distributed implementation of the centralized Kalman filter, and the correlations between inter-robot estimates are well tracked and updated. However, the communication cost for this distributed implementation is very high. In particular, an all-to-all communication is needed after every observation. As a result, the algorithm performance is very susceptible to the communication failure. The estimation consistency no longer holds with a single communication failure, which impairs the system's overall resilience. To lower the communication cost in [RB02], a server in charge of calculating and broadcasting the estimation information is introduced in [KHG18]. Therefore, all-to-all communication is no longer necessary to recover exact inter-robot correlations. However, the introduction of the server makes the whole system less distributed. Not only is the entire system more vulnerable to the server's failure, but also an initial setup of the server is required.

Beyond the proposal of the algorithm itself, the theoretical analysis of this centralized-equivalent algorithm is reported in [MR06,HTM11]. In [MR06], with an implicit assumption that all-to-all communication is available and successful at all times, the observation configuration criterion of the bounded covariance is thoroughly investigated. We conduct a similar boundedness analysis for our algorithm, especially on the observation and the communication configurations without the assumption of perfect communication. In [HTM11], the linearization consistency issue of EKF-based localization algorithms is presented, with focus on the linearization points while calculating the Jacobians. In this chapter, we maintain the assumption that linearization error is small, and focus on the linear estimation in the performance analysis.

The over-confidence problem can be avoided by only fusing uncorrelated estimates in a multirobot system. This idea is substantiated by keeping a bank of filters in each robot [BWL09, ZR19]. In [BWL09], each underwater vehicle maintains a bank of EKFs and only uncorrelated estimates in the bank are fused subsequently. A similar localization method is proposed for mobile robots while simultaneously tracking targets in [ZR19]. The main disadvantage of this category is that the number of the filters in a single bank grows exponentially with the number of the robots in a system, which imposes significant storage cost. Another way to realize this idea, called the state exchange scheme, is proposed in [KCA06]. Specifically, there is no fusion but rather replacement within robot estimates to maintain the independence between robot estimates. Historical information is therefore discarded with the arrival of new information, which leads to extremely inefficient estimates.

Instead of retrieving exact inter-robot correlations, some approaches approximate the correlations and thus largely decrease communication cost. Covariance intersection is often applied in these approaches, since it can fuse several estimates without knowing the correlations and maintain estimation consistency at the same time. To the best of our knowledge, the first application of CI for cooperative localization is the example in [ARM01]. Our algorithm is similar to this one, but we generalize the algorithm and systematically study the boundedness criterion in this paper. CI is also applied differently in the cooperative localization in [CNG13]. In this algorithm, each robot only keeps its own state estimate, and the relative observation is fused by CI. As a consequence, the estimation is very conservative in this method. On the contrary, in our algorithm, robots keep an estimate of the entire system, and relative observations can directly update the estimate. Consequently, our algorithm is not overly conservative, as verified by the experiments in Sec. **??**. An extended work of [CNG13] is presented in [KAC19] by incorporating the covariance union. However, as an even more conservative fusion scheme than CI, this algorithm with covariance union is too conservative for any practical use.

While algorithms based on CI ensure estimation consistency, some other algorithms with approximated correlations do not maintain such property. The split covariance intersection is applied in cooperative localization in [LNY13]. The main drawback of this approach is that the independent part and the dependent part can not be clearly split. Therefore, the fusion in relative observation is problematic, as mentioned in [CNG13]. In [LSR18], the exact covariance matrix is approximated by a block diagonal matrix, and the inter-robot correlations are thus suppressed. Since the estimation consistency is not maintained, the over confidence problem can occur when applying this algorithm. A cooperative localization algorithm that targets at the scenario with measurements at different time instances is proposed in [IGR12]. However, the fundamental Kalman filtering assumption of the noise independence has to be contradicted to avoid recursive updates among robots, which also raises the same concerns of estimation consistency.

### 3.1.2 Particle Filter-based Approaches

To alleviate the nonlinearity issues in multirobot localization, particle filters are often applied [FBK00, HMS03b, PBM12]. However, the correlations among robot estimation is not easy to handle in particle filter-based cooperative localization algorithms. One of the early attempts to applying particle filter in cooperative localization can be found in [FBK00]. However, the correlations between robots are ignored, and the result is overly confident. In [HMS03b], a dependency tree is introduced to alleviate the double counting problem between two robots, but it only avoids the most obvious cases and still cannot prevent the over-confidence problem from happening. In [PBM12], a particle clustering method is introduced to reduce the computational complexity of particle filter-based methods, but correlations between estimates are not explicitly addressed. In fact, the authors wrongly assume the independence between the estimates in different robots in reciprocal sampling.

In summary, particle filter-based multirobot cooperative localization algorithms cannot track the correlations between distributed estimates easily. Moreover, particle filters are generally more computationally expensive compared to Kalman filter-based approaches.

### 3.1.3 Optimization-based Approaches

Cooperative localization can also be solved with optimization-based methods, including maximum likelihood estimation [HMS03a] and maximum a posteriori estimation [NRM09]. Optimization-based approaches first formulate cooperative localization as a nonlinear least squares problem in a centralized fashion, and then is directly solved offline. To counter the centralized modeling and offline solving for localization, excessive communication is necessary between distributed robots. As a result, in both algorithms [HMS03a, NRM09], robots have to broadcast their information to the entire team regularly. In terms of the offline nature, the authors in [NRM09] partially tackle this problem by marginalization, but an all-to-all communication is expected in this marginalization step. For optimization-based approaches, the burden of communication makes them less popular compared to those aforementioned algorithms.

## 3.2 Multirobot Cooperative Localization Algorithm

We consider a multi-robot system in the 2D scenario with $N$ robots. At time $t$, the spatial state of robot $i$ is given by $q_{i,t} = [\theta_{i,t}, p_{i,t}^{\mathsf{T}}]$, which includes the orientation $\theta_{i,t}$ and the Cartesian position $p_{i,t} = [x_{i,t}, y_{i,t}]^{\mathsf{T}}$, for $i \in \{1, \ldots, N\}$. We assume that spatial states across all robots are in a common reference frame, which can be initialized by the cooperative localization setting [TZZ10, ZR13]. The robots can observe several distinguishable landmarks whose positions are given. While all landmarks serve as a reference for absolute spatial state, without loss of generality, we consider a single landmark in the environment, denoted by $\lambda$.

In the proposed cooperative localization algorithm, robot $i$ has to track its own spatial state and the positions of other robots. We can represent the state of the entire system estimated by robot $i$ as

$$s_{i,t} = \left[p_{1,t}^{\mathsf{T}}, \ldots, q_{i,t}^{\mathsf{T}}, \ldots, p_{N,t}^{\mathsf{T}}\right]^{\mathsf{T}}. \tag{3.1}$$

We consider that case where the orientations of other robots are not tracked by robot $i$. The state defined in (3.1) is similar to the one in the EKF SLAM [DNC01], where those $p_{i,t}$ in (3.1) are not stationary but dynamic. The proposed algorithm remains valid when the orientations of other robots are tracked with various sensing modalities. In fact, the proposed algorithm will be easier if all robots track the same state. We instead demonstrate the necessary steps when the robots track different states in the model (3.1).

Based on the Kalman filtering, robot $i$ keeps a Gaussian estimate of $s_{i,t}$, denoted by $\hat{\mathbf{s}}_{i,t}$, with mean $\bar{s}_{i,t}$ and covariance $\Sigma_{i,t}$. Depending on the type of arriving information, the proposed cooperative localization algorithm contains three updates:

- the *time propagation update* at the arrival of the proprioceptive information,

- the *observation update* at the arrival of the exteroceptive information, and
- the *communication update* at the inter-robot communication.

The proposed algorithm does not require communication after the inter-robot observation. Therefore, all these three sources of informations contribute independently and complementarily to achieve localization.

### 3.2.1 Time Propagation Update

The time propagation update is performed when robot $i$ has the proprioceptive information of the system, which consists of its own odometry input and those of other robots. Robot $i$ has the odometry input $u_{i,t}$, and estimates its next spatial state by a generic motion model:

$$\mathbf{q}_{i,t+1} = f(\mathbf{q}_{i,t}, u_{i,t} + \mathbf{w}_{i,t}), \tag{3.2}$$

where $\mathbf{w}_{i,t}$ is the input noise and it is modeled as a zero-mean Gaussian random vector with covariance matrix $Q_w$.

The odometry inputs for other robots, $u_{j,t}$, $j \neq i$, however, are not available for robot $i$. Without the exact value, we regard $u_{j,t}$ as a random variable, and the variability of that random variable is large enough to incorporate all possible values and to ignore the noise effect. The goal is not to guess the odometry input of other robots, but to maintain large estimation uncertainty that the estimate can be corrected during the observation or the communication updates. To be specific, we model the input $u_{j,t}$ as a Gaussian random vector with covariance matrix $Q_u$ large enough to maintain the estimation consistency. That is, for robot $i$,

$$\mathbf{p}_{j,t+1} = f_p(\mathbf{p}_{j,t}, u_{j,t}), \quad j \neq i. \tag{3.3}$$

As the input noise in each robot is independent, the time update for $\hat{\mathbf{s}}_{i,t}$ can be easily obtained, as in Algorithm 1.

**Algorithm 1** The time propagation update for robot $i$

```
**Input**: $\bar{s}_{i,t}$, $\Sigma_{i,t}$, $u_{i,t}$
**Output**: $\bar{s}_{i,t+1}$, $\Sigma_{i,t+1}$

$$\bar{s}_{i,t+1} = \begin{bmatrix} f_p(\bar{p}_{1,t}, \mathsf{E}[u_{1,t}]) \\ \vdots \\ f(\bar{q}_{i,t}, u_{i,t}) \\ \vdots \\ f_p(\bar{p}_{N,t}, \mathsf{E}[u_{N,t}]) \end{bmatrix}.$$
$F_i = \frac{\partial f(q,u)}{\partial q}(\bar{q}_{i,t}, u_{i,t})$.
$F_j = \frac{\partial f_p(p,u)}{\partial p}(\bar{p}_{j,t}, \mathsf{E}[u_{j,t}])$, for $j \neq i$.
$G_i = \frac{\partial f(q,u)}{\partial u}(\bar{q}_{i,t}, u_{i,t})$.
$G_j = \frac{\partial f_p(p,u)}{\partial u}(\bar{p}_{j,t}, \mathsf{E}[u_{j,t}])$, for $j \neq i$.
$Q = \mathsf{Diag}(G_1 Q_u G_1^\mathsf{T}, \ldots, G_i Q_w G_i^\mathsf{T}, \ldots, G_N Q_u G_N^\mathsf{T})$.
$\Sigma_{t+1} = \mathsf{Diag}(F_1, \ldots, F_N) \Sigma_t \, \mathsf{Diag}(F_1, \ldots, F_N)^\mathsf{T} + Q$.
```

### 3.2.2 Observation Update

When robot $i$ observes either the landmark or other robots, the observation update is performed with the exteroceptive information. Specifically, robot $i$ observes the landmark in the environment according to model

$$\mathbf{o}_{i\lambda,t} = h_{i\lambda}(\mathbf{q}_{i,t}) + \mathbf{v}_{i\lambda,t}, \tag{3.4}$$

where the $\mathbf{v}_{i\lambda,t}$ is the observation noise modeled as zero-mean Gaussian with covariance $R_{i\lambda,t}$. The relative observation model between two robots is similarly given as

$$\mathbf{o}_{ij,t} = h_{ij}(\mathbf{q}_{i,t}, \mathbf{p}_{j,t}) + \mathbf{v}_{ij,t}. \tag{3.5}$$

In reality, robot $i$ can observe more than one object at the same time, and the observation results may therefore be correlated. Thus, we define the set $O_{i,t}$ as the set of

**Algorithm 2** The observation update for robot $i$

```
**Input**: $\bar{s}_{i,t}$, $\Sigma_{i,t}$, $o_{i,t}$
**Output**: $\bar{s}_{i,t^+}$, $\Sigma_{i,t^+}$

    $H_i = \left[\frac{\partial h_{ij}(s)}{\partial s}(\bar{s}_{i,t})\right]_{j\in O_{i,t}}.$
    $\bar{s}_{i,t^+} = \bar{s}_{i,t} + \Sigma_{i,t}H_i^\mathsf{T}(H_i\Sigma_{i,t}H_i^\mathsf{T} + R_{i,t})^{-1}(o_{i,t} - H_i\bar{s}_{i,t}).$
    $\Sigma_{i,t^+}^{-1} = \Sigma_{i,t}^{-1} + H_i^\mathsf{T}R_{i,t}^{-1}H_i.$
```

objects that robot $i$ observes at time $t$, which may includes landmarks and robots. With $O_{i,t} = \{i_1, i_2, \ldots, i_{n_i}\}$, we stack all the measurements at time $t$ into the vector $o_{i,t}$,

$$o_{i,t} = \begin{bmatrix} o_{ii_1,t} \\ \vdots \\ o_{ii_{n_i},t} \end{bmatrix} = [o_{ij,t}]_{j\in O_{i,t}}, \tag{3.6}$$

together with the entire observation noise $\mathbf{v}_{i,t} = [\mathbf{v}_{ij,t}]_{j\in O_{i,t}}$. With the covariance of the noise $\mathbf{v}_{i,t}$ denoted by $R_{i,t}$, we have the EKF observation updates:

$$\bar{s}_{i,t^+} = \bar{s}_{i,t} + \Sigma_{i,t}H_i^\mathsf{T}(H_i\Sigma_{i,t}H_i^\mathsf{T} + R_{i,t})^{-1}(o_{i,t} - H_i\bar{s}_{i,t}), \tag{3.7}$$

and

$$\Sigma_{i,t^+}^{-1} = \Sigma_{i,t}^{-1} + H_i^\mathsf{T}R_{i,t}^{-1}H_i, \tag{3.8}$$

where the observation matrix is the stacked matrix given by

$$H_i = \left[\frac{\partial h_{ij}(s)}{\partial s}(\bar{s}_{i,t})\right]_{j\in O_{i,t}}. \tag{3.9}$$

### 3.2.3 Communication Update

When robot $j$ sends its estimation information, in particular $\bar{s}_{j,t}$ and $\Sigma_{j,t}$, to robot $i$, robot $i$ can use this information to update its own estimation. However, the correlation between $\hat{\mathbf{s}}_{i,t}$

and $\hat{\mathbf{s}}_{j,t}$ is hard to track in a distributed system. Without knowing the exact correlations, we use CI to fuse these estimates to maintain the estimation consistency.

The direct application of CI by (2.6) and (2.7) is problematic, because $\hat{\mathbf{s}}_{i,t}$ and $\hat{\mathbf{s}}_{j,t}$ do not estimate the same state. In particular, the orientation estimate of $\theta_{i,t}$ is in $\hat{\mathbf{s}}_{i,t}$ but not in $\hat{\mathbf{s}}_{j,t}$. In order to ensure that $\hat{\mathbf{s}}_{i,t}$ and $\hat{\mathbf{s}}_{j,t}$ represent the same state, we first have to remove the estimate of $\theta_j$ from $\hat{\mathbf{s}}_{j,t}$, and then add the dummy estimate of $\theta_i$. We denote the resulting estimate as $\hat{\mathbf{s}}^j_{i,t}$ and then the CI can be applicable at robot $i$.

To remove the estimate of $\theta_j$ from $\hat{\mathbf{s}}_{j,t}$, we use a $2N \times (2N+1)$ matrix defined by

$$[T_{j^-}]_{m,n} = \begin{cases} 1 & \text{if } m = n, n \leq 2(j-1) \\ 1 & \text{or } m = n-1, n \geq 2j \\ 0 & \text{otherwise} \end{cases} .$$

Therefore, $T_{j^-}\hat{\mathbf{s}}_{j,t}$ will be the estimate of $[p_{1,t}^\mathsf{T}, \ldots, p_{N,t}^\mathsf{T}]$ with mean $T_{j^-}\bar{s}_{j,t}$ and covariance matrix $T_{j^-}\Sigma_{j,t}T_{j^-}^\mathsf{T}$. Equivalently, the same estimate admits an information form with the information mean $(T_{j^-}\Sigma_{j,t}T_{j^-}^\mathsf{T})^{-1}T_{j^-}\bar{s}_{j,t}$ and the information matrix $(T_{j^-}\Sigma_{j,t}T_{j^-}^\mathsf{T})^{-1}$.

Next, we insert $\theta_i$ to the estimate $T_{j^-}\hat{\mathbf{s}}_{j,t}$ in the information form to obtain $\hat{\mathbf{s}}^j_{i,t}$. Since there is no information of $\theta_i$ from robot $j$, this step just ensures that the corresponding terms in the vector are matched, and the variance of $\theta_i$ in $\hat{\mathbf{s}}^j_{i,t}$ will be infinite. We use a $(2N+1) \times 2N$ matrix,

$$[T_{i^+}]_{m,n} = \begin{cases} 1 & \text{if } m = n, n \leq 2(i-1) \\ 1 & \text{if } m = n+1, n \geq 2i \\ 0 & \text{otherwise} \end{cases} ,$$

to append $\theta_i$. Thus, the information mean of $\hat{\mathbf{s}}^j_{i,t}$ will be $T_{i^+}(T_{j^-}\Sigma_{j,t}T_{j^-}^\mathsf{T})^{-1}T_{j^-}\bar{s}_{j,t}$, and the corresponding information matrix will be $T_{i^+}(T_{j^-}\Sigma_{j,t}T_{j^-}^\mathsf{T})^{-1}T_{i^+}^\mathsf{T}$. By this construction, the exact mean of $\theta_{i,t}$ in the estimate $\hat{\mathbf{s}}^j_{i,t}$ is not important, since the corresponding variance is infinite, and will not affect the result of CI.

**Algorithm 3** The communication update for robot $i$

```
**Input**: $\bar{s}^j_t$, $\Sigma_{j,t}$, $j \in C^*_{i,t}$
**Output**: $\bar{s}^i_{t^+}$, $\Sigma_{i,t^+}$

To construct the information form:
    $\bar{e}^j_{i,t} = T_{i^+}(T_{j^-}\Sigma_{j,t}T^{\mathsf{T}}_{j^-})^{-1}T_{j^-}\bar{s}_{j,t}$, $j \in C_{i,t}$
    $I^j_{i,t} = T_{i^+}(T_{j^-}\Sigma_{j,t}T^{\mathsf{T}}_{j^-})^{-1}T^{\mathsf{T}}_{i^+}$, $j \in C_{i,t}$

To fuse incoming estimates by CI:
    $\bar{s}_{i,t^+} = \Sigma_{i,t^+}\left(\sum_{j\in C_{i,t}} c_j\bar{e}^j_{i,t} + c_i\Sigma^{-1}_{i,t}\bar{s}_{i,t}\right)$.
    $\Sigma^{-1}_{i,t^+} = \sum_{j\in C_{i,t}} c_j I^j_{i,t} + c_i\Sigma^{-1}_{i,t}$.
```

We define the set $C_{i,t}$ to contain all robots whose information is received at robot $i$ at time $t$, and $C^*_{i,t} = C_{i,t} \cup \{i\}$. Together with the convex coefficient $\{c_j, j \in C^*_{i,t}\}$, we have the communication update described in Algorithm 3.

## 3.3 Boundedness Analysis of the Position Estimation Covariance

For the localization algorithm, the boundedness of the covariance matrix ensures that the estimation uncertainty is limited, which is essential for the success of the high-level tasks. Whether the estimation covariance matrix of each robot is bounded depends on the communication and the observation configurations of the entire multirobot system. To concretely study the covariance boundedness, we consider a system with unicycle motion model and the bearing-and-range measurements to demonstrate the analysis. We then derive the covariance upper bound of the estimation covariance, and apply the result from the distributed estimation algorithm to obtain the boundedness criterion.

To proceed the analysis, we furthermore impose two assumptions on the system:

1. Each robot has its orientation estimate, and the upper bound of the orientation estimate variance $\sigma_\theta^2$ is small and given.

2. The observation and communication configurations are invariant over time, including the CI coefficients.

As introduced in [MR06], the first assumption decouples the position estimation from the orientation estimation, which is the main source of the linearization inconsistency problem [JU01, BNG06, HD07]. As the EKF heavily relies on the linearization approximation, the requirement of small orientation error also ensures the applicability of ongoing analysis. The second assumption is imposed to assure that the entire system configuration is stationary. As a result, the boundedness analysis of the cooperative localization algorithm can be achieved by that of the distributed estimation algorithm [CM18a].

With the assumption that the orientation estimate is provided, all robots now estimate the same state, or the positions of all robots, denoted by $\xi_t = [p_{1,t}^\intercal, \ldots, p_{N,t}^\intercal]^\intercal$. The estimate of $\xi_t$ at robot $i$ is $\hat{\boldsymbol{\xi}}_{i,t}$, with mean $\bar{\xi}_{i,t}$ and covariance $\Phi_{i,t}$. While all the robots are estimating the same state, the communication step just degenerates to the vanilla CI step.

### 3.3.1 System Model

Given the velocity input $u_{i,t}$, the unicycle model describes the state propagation as

$$\mathbf{p}_{i,t+1} = \begin{bmatrix} \mathbf{x}_{i,t} + (u_{i,t} + \mathbf{w}_{i,t})\Delta t \cos\theta_{i,t} \\ \mathbf{y}_{i,t} + (u_{i,t} + \mathbf{w}_{i,t})\Delta t \sin\theta_{i,t} \end{bmatrix}, \tag{3.10}$$

where $\mathbf{w}_{i,t}$ denotes the input noise and $\Delta t$ is the time interval between two consecutive update points.

In terms of the observation model, we first set up a generic relative observation model, whose observability can be explicitly characterized. We then use the relative observation model as an intermediate step to analyze the bearing-and-range measurements. When robot $i$

observes object $j$, which can be either another robot or a landmark, the relative measurement $\mathbf{o}_{ij,t}$ is given by

$$\mathbf{o}_{ij,t} = C^{\mathsf{T}}(\theta_{i,t})(\mathbf{p}_{j,t} - \mathbf{p}_{i,t}) + \mathbf{v}_{ij,t}, \tag{3.11}$$

where $C(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ is the rotation matrix. The observation noise $\mathbf{v}_{ij,t}$ is a zero-mean Gaussian random vector with covariance $R_{v,ij}$. If robot $i$ observes object $j$ by the bearing measurement $\boldsymbol{\phi}_{ij}$ and the range measurement $\mathbf{r}_{ij}$, we characterize this measurement as

$$\begin{aligned} \mathbf{o}'_{ij,t} &= \begin{bmatrix} \boldsymbol{\phi}_{ij,t} \\ \mathbf{r}_{ij,t} \end{bmatrix} + \mathbf{v}'_{ij,t} \\ &= \begin{bmatrix} \tan^{-1}\left(\frac{\mathbf{y}_{j,t}-\mathbf{y}_{i,t}}{\mathbf{x}_{j,t}-\mathbf{x}_{i,t}}\right) - \theta_{i,t} \\ \sqrt{(\mathbf{x}_{j,t} - \mathbf{x}_{i,t})^2 + (\mathbf{y}_{j,t} - \mathbf{y}_{i,t})^2} \end{bmatrix} + \mathbf{v}'_{ij,t}. \end{aligned} \tag{3.12}$$

With the bearing measurement $\phi_{ij,t}$ and the range measurement $r_{ij,t}$, the relative measurement can be obtained by

$$o_{ij,t} = r_{ij,t} \begin{bmatrix} \cos(\phi_{ij,t}) \\ \sin(\phi_{ij,t}) \end{bmatrix}, \tag{3.13}$$

together with the noise by linearizing (3.13)

$$\mathbf{v}_{ij,t} = \begin{bmatrix} -r_{ij,t}\sin(\phi_{ij,t}) & \cos(\phi_{ij,t}) \\ r_{ij,t}\cos(\phi_{ij,t}) & \sin(\phi_{ij,t}) \end{bmatrix} \mathbf{v}'_{ij,t}. \tag{3.14}$$

### 3.3.2 Covariance Boundedness Analysis

We omit the technical derivation of covariance upper-bound $\Psi_{i,t}$ of $\Phi_{i,t}$, which can be found in [CCM22]. All we need now is that $\Psi_{i,t} \geq \Phi_{i,t}$, and the all coefficients of the updates of $\Psi_{i,t}$ are constant. We now apply the result of the distributed Kalman filter with CI in [CM18a] to analyze the covariance boundedness of $\Psi_{i,t}$. To explicitly characterize the relations among all robots, we use graphs to describe the observation and the communication configurations in the multirobot system. We define the observation graph and the communication graph separately to distinguish the observation and the communications relations.

We define the observation graph of robot $i$ as a graph $\mathcal{G}_{O_i} = \{\Omega^*, E_{O_i}\}$. The nodes of the graph $\Omega^* = \{1, \ldots, n, \lambda\}$, which includes all the robots as well as the landmark. The pair $(i, j) \in E_{O_i}$ if $j \in O_i$. In other words, the links in the observation graph $\mathcal{G}_{O_i}$ stand for the observation from robot $i$ to entity $j$. We also define the communication graph as a graph $\mathcal{G}_C = \{\Omega^*, E_C\}$, where $(i, j) \in E_C$ if $j \in C_i$. We then can use the following notation to collect all the robots that contribute the information to robot $i$ by the communication links.

**Definition 2** (Super Neighborhood [CM18a])**.** *For $j \neq i$, $j \in S_i$ if there exists a path in $\mathcal{G}_C$ from $j$ to $i$.*

We define that $S_i^* = S_i \cup \{i\}$.

**Proposition 1** (The Boundedness Criterion)**.** *If the graph $\mathcal{G}_i = (\Omega^*, \cup_{j \in S_i^*} E_{O_j})$ is weakly connected, then $\Phi_{i,t}$ is bounded.*

Proposition 1 states that as long as all the information collected by robot $i$ covers the entire robot team, the information is sufficient enough to localize the entire robot team, which leads to bounded $\Phi_{i,t}$ as well. Proposition 1 also signifies that the information can come either from observation or from communication, and both sources contribute to the localization performance.

## 3.4 Simulations and Experiments

In this section, we present the performance and the resilience of our algorithm as compared to other four state-of-the-art multirobot cooperative localization methods. Based on the state tracked in a single robot and the underlying method, for simplicity, we rename all 5 algorithms as

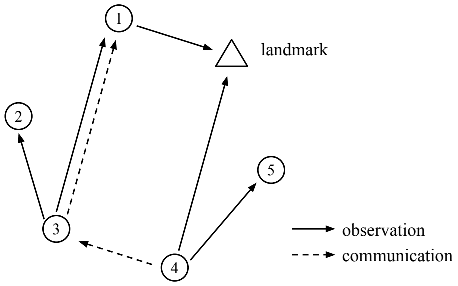


Figure 3.1: The system topology with $N = 5$ robots in the simulation. The communication graph is specified for the GS algorithm. For the LS algorithms, a fully connected communication graph is inherently required.

- the local-state centralized equivalent (LS-Cen) [RB02],
- the local-state covariance intersection (LS-CI) [CNG13],
- the local-state split covariance intersection (LS-SCI) [LNY13],
- the local-state block diagonal approximation (LS-BDA) [LSR18],
- our global-state covariance intersection (GS-CI).

As the LS-Cen algorithm uses the entire available information without any approximation, the result of LS-Cen can be regarded as the optimal performance. We first simulate all methods with generated data, which not only shows that our algorithm requires far sparser communication topology to achieve comparable performance of other methods, but also visualizes the boundedness analysis. Next, we analyze all methods in a common multirobot dataset, and show that our algorithm is more resilient during unfavorable and adverse communication loss than other algorithms.

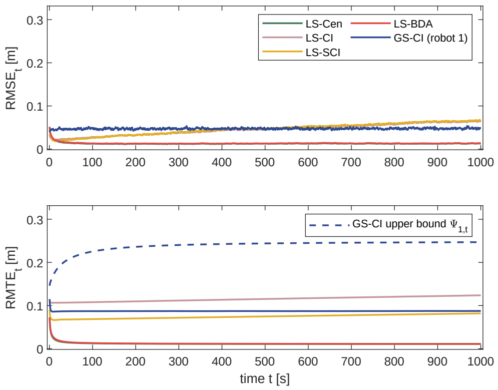


Figure 3.2: The cooperative localization performance with generated data. As for the communication graph, local-state (LS) algorithms assume all-to-all and perfect communication, and the global-state (GS) algorithm follows the graph in Fig. 3.1. The RMSE plot of LS-Cen and that of LS-BDA are overlapped. For the proposed GS-CI, robot 1 has bounded covariance matrix, as suggested by Proposition 1.

### 3.4.1 Simulation

To begin with, we investigate the performances of all 5 algorithms with generated data.[1] In this simulation, we consider that the orientation estimate is given for $N = 5$ robots. For each robot, the velocity input $u_{i,t}$ is taken uniformly between $[-0.09, 0.09]$ m/s, in which the velocity input variance in GS-CI can then be calculated. Fig. 3.1 specifies the observation graph for the multirobot system. In terms of the communication graph, for local state (LS) algorithms, a fully connected communication graph is inherently required and therefore communication after each relative observation step is assumed to be perfect. For the global state (GS) algorithm, the communication is constrained as in Fig. 3.1, which is far sparser than those communication graphs for LS algorithms.

To quantify the estimation performance against the ground truth, we define the root-

[1] The code of this subsection is available at https://github.com/tsangkai/multirobot_localization.

Table 3.1: Time-averaged RMSE of UTIAS datasets [m]

| sub-dataset | LS-Cen | LS-CI | LS-SCI | LS-BDA | GS-CI |
|---|---|---|---|---|---|
| 1 | 1.28 | 1.67 | 1.12 | 1.31 | 1.42 |
| 2 | 0.74 | 1.41 | 1.75 | 0.80 | 0.79 |
| 3 | 0.23 | 0.96 | 1.23 | 0.26 | 0.29 |
| 4 | 0.21 | 1.21 | 1.49 | 0.23 | 0.28 |
| 5 | 1.72 | 5.45 | 5.20 | 1.79 | 2.17 |
| 6 | 0.79 | 2.08 | 2.07 | 0.82 | 0.85 |
| 7 | 0.59 | 1.49 | 1.73 | 0.86 | 0.82 |
| 8 | 0.71 | 0.96 | 2.00 | 0.84 | 0.80 |
| 9 | 0.26 | 0.28 | 0.65 | 0.27 | 0.31 |

mean-squared-error (RMSE) of the entire $N$ robots as

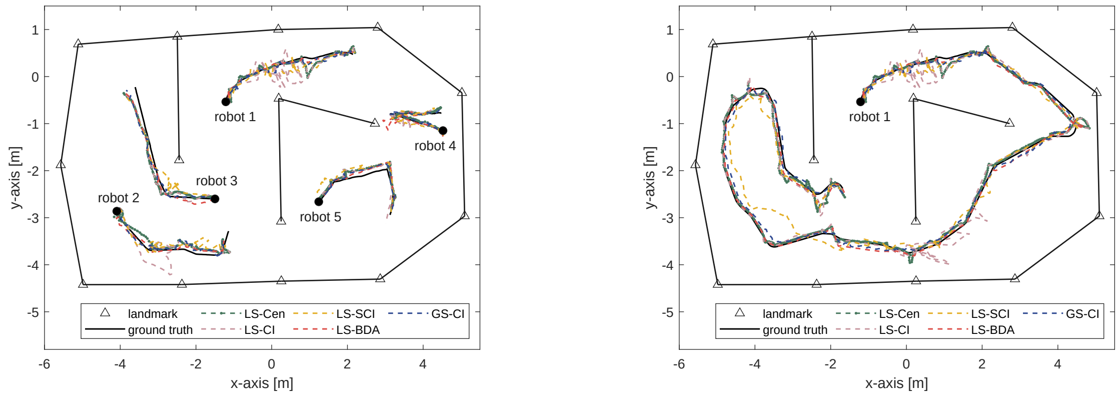
$$RMSE_t = \sqrt{\frac{\sum_{i=1}^{N} \left\| [\bar{p}_{i,t}]_i - p_{i,t} \right\|^2}{N}},$$

where $[\bar{p}_{i,t}]_j$ is the estimate of $p_{i,t}$ by robot $j$. We also consider the root-mean-trace-error (RMTE) to capture the uncertainty evaluated in the algorithm, defined as:

$$RMTE_t = \sqrt{\frac{\sum_{i=1}^{N} \mathsf{tr}([\Phi_{i,t}]_i)}{N}},$$

where $[\Phi_{i,t}]_j$ denotes the sub-covariance matrix of robot $j$ that relates to the position estimate of robot $i$ at time $t$. We plot the result in Fig. 3.2. In particular, for the GS-CI, we plot both the RMSE and the RMTE of robot 1 to discuss the boundedness analysis in Sec. 3.3.2.

Based on the RMSE in Fig. 3.2, the LS-BDA and the proposed GS-CI show desirable results since their RMSEs remain relatively constant. However, the LS-BDA does not guarantee the estimation consistency, and achieve this performance with the fully-connected communication graph. Other CI-based methods, including LS-CI and LS-SCI, have increasing localization error over time, due to the overly conservative estimation.



(a) All 5 robots for 20 sec

(b) Robot 1 for 150 sec

Figure 3.3: The trajectories of robots with different localization algorithms in sub-dataset 9 starting 1500 sec. The communication is assumed to be available whenever needed. The proposed GS-CI is comparable to the LS-Cen, whose result is regarded as the best achievable performance.

Even though the proposed GS-CI shows desirable result, the required communication graph specified in Fig. 3.1 is far sparser in the GS-CI than those of the LS algorithms. Especially, as the graph $\mathcal{G}_1$ is weakly connected, Proposition 1 assures the boundedness of $\Phi_{1,t}$. In fact, besides the observation of the landmark, the rest of the information of robot 1 comes from the single communication from robot 3. This simulation thus shows how the observation and the communication are treated as complementary information sources in the proposed algorithm. In addition to the sparseness of the communication graph, the proposed GS-CI has the estimates of the entire robot team by design, which facilitates the cooperative planning within the multirobot system.

### 3.4.2 Communication Resilience Experiment on the UTIAS Dataset

To demonstrate the resilience to communication failures of our algorithm, we use the University of Toronto Institute for Aerospace Studies (UTIAS) Multi-Robot Cooperative Lo-

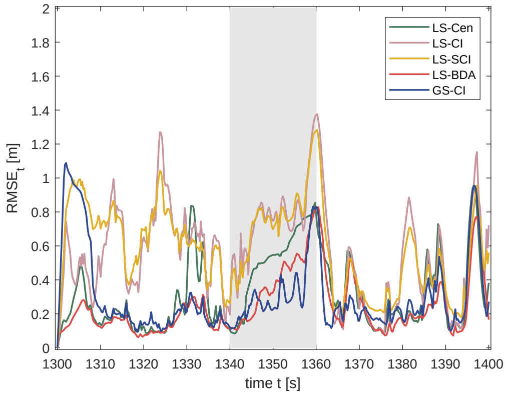


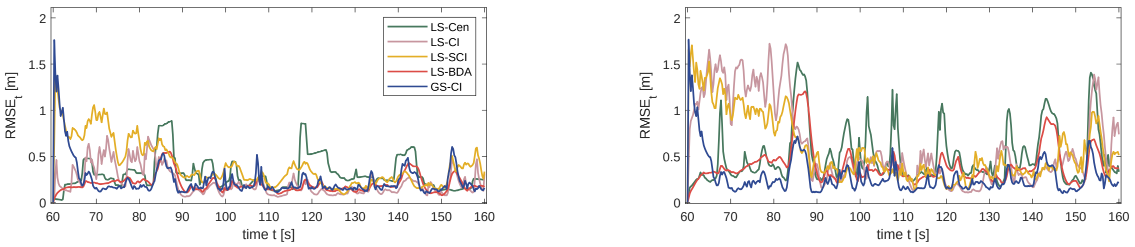
Figure 3.4: The RMSE with blocked communication from 1340 to 1360 sec of sub-dataset 9. The communication remains available besides the window between 1340 to 1360 sec. The proposed GS-CI shows resilience during this 20 sec time window by separating the communication update and the observation update.

calization and Mapping dataset [LHB11]. This dataset is a cohesive collection of odometry and observation data from $N = 5$ robots, together with accurate ground truth data of robot and landmark positions. This dataset is also widely used across several works as a common benchmark dataset.

We first test those 5 algorithms on the entire 9 sub-datasets with all communication available on the first 500 sec.[2] Each algorithm estimates both the orientation and the position, and we mainly consider the position estimation here. We record the time-averaged $RMSE_t$ in Table 3.1 for all 9 sub-datasets. As expected, the LS-Cen algorithm has the lowest localization error in the entire 9 sub-datasets. Overall, the proposed GS-CI has comparable localization performance compared to the LS-Cen, which is consistent with the previous simulation.

Among all 9 sub-datasets in the UTIAS dataset, sub-dataset 9 is the only one that contains barriers, thus creating a more challenging scenario with its occasional occlusions in

[2] The code of this subsection is available at https://git.uclalemur.com/kjchen/tro2020/tree/master/v3.



(a) $\rho = 0.1$

(b) $\rho = 0.9$

Figure 3.5: The RMSE of a 100 sec snapshot in sub-dataset 9 with two different communication link failure probabilities $\rho$. As there are more communication failures, the estimation error is larger with $\rho = 0.9$ that that with $\rho = 0.1$ for all algorithms. However, algorithms are affected differently. For instance, between 140 and 150 sec, the proposed GS-CI does not have a significant increase in the estimation error, and thus shows its resilience.

observations. We therefore select sub-dataset 9 to demonstrate the communication resilience in the following. To visualize this sub-dataset as well as the localization algorithms, we plot the estimated trajectories of all 5 robots in Fig. 3.3. Both figures shows that the proposed GS-CI is comparable to the LS-Cen, whose result is regarded as the best achievable performance in the ideal scenario.

To investigate the communication resilience of each algorithm, we consider the scenario where the communication is blocked from an adverse source, and study the localization performance dynamics during this period. While different time windows show similar trends, we plot the time window between 1300 and 1400 sec of sub-dataset 9 in Fig. 3.4 as an example. During the entire 100 sec time window, the communication is entirely blocked from 1340 to 1360 sec, while the communication remains available for the rest of the time. In this 20 sec window, which is marked as shaded area in Fig. 3.4, the estimation errors of all cooperative localization algorithms increase, but the proposed GS-CI has the lowest slope. In other words, by separating the communication update and the observation update, our algorithm is less susceptible from the communication unavailability but continues integrating informa-

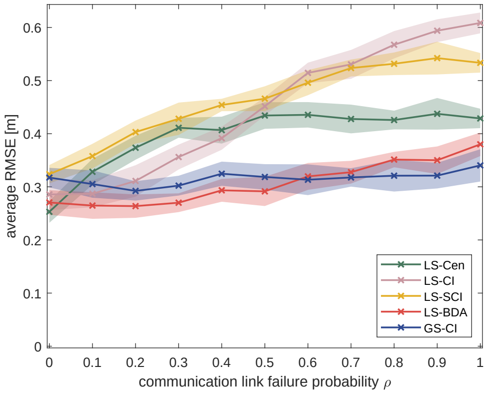


Figure 3.6: The time-averaged RMSE with varying communication link failure probabilities $\rho$ of the first 500 sec of sub-dataset 9. We analyze every 50 sec and plot the 3 standard deviation error bar for all 10 windows. The proposed GS-CI is only slightly affected by the increase of $\rho$, and it shows resilience across different $\rho$ values, especially in unfavorable communication conditions.

tion from the observation updates. For LS algorithms, since communication is essential to complete the some observation updates, the localization performances are largely impaired in this 20 sec window.

We further consider the effect of the communication link failure probability on those cooperative localization algorithms. In particular, we consider the scenario in which all the communication links between two robots exist, but suffer from failures with a constant probability $\rho$. For instance, the number of communications after the relative observation of LS-BDA is 2. Therefore, with probability $(1-\rho)^2$, the relative observation update of LS-BDA can be completed successfully without communication failure.

To emphasize the effects on estimation dynamics, we plot the 100 sec snapshots with $\rho = 0.1$ and $\rho = 0.9$ in Fig. 3.5. The former case with $\rho = 0.1$ is close to the ideal case where all the communication is assumed perfect, while the later case with $\rho = 0.9$ is similar to the 20 sec window with blocked communication in Fig. 3.4. By comparing the two snapshots, the effect of the communication link failure probability $\rho$ on the cooperative localization algorithms becomes noticeable. For instance, between 140 and 150 sec, all the estimation errors increase with $\rho = 0.9$ due to communication failures, but the resilience of each algorithm differs. Among all LS algorithms, the LS-BDA shows its estimation accuracy when $\rho = 0.1$. However, while the LS-BDA has comparable performance to our GS-CI with $\rho = 0.1$, it has overall worse localization performance with $\rho = 0.9$. Such comparison substantiates the resilience of our GS-CI under the communication failure.

To characterize the resilience performance under various scenarios, we plot the time-averaged RMSE against the communication link failure probability $\rho$ on the first 500 sec of sub-dataset 9 in Fig. 3.6. In general, the increase of the communication link failure probability $\rho$ has negative impact on all algorithms, as the information coming from the communication becomes less available. However, as the communication failure probability $\rho$ increases, the LS-Cen and the LS-BDA algorithms suffer from higher localization error, even though they show superb localization performance in the ideal cases. On the contrary, the proposed GS-CI maintains a relatively flat curve as the communication failure probability $\rho$ increases. As the proposed GS-CI is only slightly affected by the increase of $\rho$, it shows resilience across different $\rho$ values, especially in unfavorable communication conditions.

# CHAPTER 4

# Multirobot Localization II: Optimal Scheduling

For the algorithm developed in the previously chapter, we use CI to enable a fully distributed multirobot localization algorithm. We now further investigate the effect of the conservative nature of CI on the localization performance. In more detailed, CI in communication update introduces extra uncertainty to ensure the estimation consistency. Therefore, the communication update brings in information, as well as extra uncertainty. To use this algorithm, the critical question is when to take the communication update? And what is the effect on the overall performance? We formulate this problem as an optimal scheduling problem of the operations in the localization algorithm, which trades off between the operation cost and the resulting performance. We particular consider the observation update and the communication update, since both operations can contribute to the estimation accuracy.

We layout the optimal scheduling method in this chapter, which consists of 3 steps: First, we derive the covariance upper bound for the algorithm, which is identical to Sec. 3.3.2. The coefficients of the upper bound is independent of the state, and we therefore only discuss the upper bound. Second, we find the limiting continuous-time approximation of the discrete-time covariance update. The analysis of discrete-time modeling requires the existence of recurrent points, which restricts the flexibility of the method. The corresponding covariance update the follows a continuous-time Riccati recursion, and the steady-state localization uncertainty can be represented by the solution of the corresponding continuous-time algebraic Riccati equation (CARE). Finally, depending on different goals, we formulate and numerically solve two optimization problems: the cost minimization problem and the trace minimization problem.

In addition to the optimal scheduling method, we furthermore analyze the effect of communication update on the estimation performance. We explicitly focus on the CI parameter $\alpha$ and the communication rate $f_c$. We use Frechet derivative to characterize the effects of these two parameters on the steady-state covariance. The analysis analytically shows the information that the communication update brings as well as the uncertainty it induces, which completely characterizes the communication update.

## 4.1 Setup

The system setup is identical to the one in previous chapter, and we only briefly state it here for self-consistency, and emphasize the difference. We consider a 2D multirobot system where robots are indexed by $\{1, \dots, N\}$, and several landmarks whose locations are known by the robots in advance. The position of robot $i$ at time $t$ is regarded as the state, denoted as $p_{i,t} = [x_{i,t}, y_{i,t}]^\mathsf{T}$ The state of the whole system is denoted by $\xi_t = [p_{1,t}^\mathsf{T}, \dots, p_{N,t}^\mathsf{T}]^\mathsf{T}$.

We denote the time difference between two consecutive time update as $T_p$ and the CI coefficient as $\alpha$. We also impose a few assumptions for derivation simplicity:

1. Robot $i$ only receive information from robot $k$, who has no information source.
2. The observation rate is larger than the communication rate.

We now can summarize the covariance update of $\Sigma_{i,t}$ as well as the covariance upper bound $\Pi_{i,t}$ in Algorithm 4.

To sum up, the covariance update equations for exact covariance $\Sigma_{i,t}$ are state dependent. We instead find the update equations of $\Pi_{i,t}$ with constant coefficient. With the same initial covariance, or $\Sigma_{i,0} = \Pi_{i,0}$, we have $\Sigma_{i,t} \leq \Pi_{i,t}$ for all $t$. Therefore, we interpret the constant update equations as the upper bound of the exact covariance, and optimize over the upper bound instead.

**Algorithm 4** The Exact and the Upper-Bound Covariance Updates of the Multirobot Cooperative Localization Algorithm

**Motion Propagation**

$$\Sigma_{i,t+1} = \Sigma_{i,t} + T_p^2 Q_u,$$

$$\Pi_{i,t+1} = \Pi_{i,t+1} + T_p^2 Q.$$

**Observation**

$$\Sigma_{i,t^+}^{-1} = \Sigma_{i,t^-}^{-1} + H_i^{\mathsf{T}} C(\theta_{i,t}) R_{o_{ij}}^{-1} C^{\mathsf{T}}(\theta_{i,t}) H_i, \tag{4.1}$$

$$\Pi_{i,t^+}^{-1} = \Pi_{i,t^-}^{-1} + H_i^{\mathsf{T}} R_i^{-1} H_i. \tag{4.2}$$

**Communication**

$$\Sigma_{i,t^+}^{-1} = \alpha \Sigma_{i,t^-}^{-1} + (1-\alpha)\Sigma_k^{-1},$$

$$\Pi_{i,t^+}^{-1} = \alpha \Pi_{i,t^-}^{-1} + (1-\alpha) H_k^{\mathsf{T}} R_k^{-1} H_k.$$

## 4.2 Continuous-Time Riccati Recursion with Operation Rates

To obtain the limiting continuous-time Riccati recursion, we have to construct the discrete-time modeling concerning the three types of updates. With operations being taken at different rates, we first define a time interval of the discrete time model $T$ equal to the period of communication update $T_c$. This implies that there are $T_c/T_p$ motion propagation updates, $T_c/T_o$ observation updates, and 1 communication update in each interval $T$. Even though the numbers of each action are fixed, the covariance update is still undetermined since the specific order of those actions is unspecified. We instead approximate the covariance update by reordering the operations: first observations, then communication, and then motion propagation.

In each time interval, the covariance update after observation and communication updates can be expressed as

$$\begin{aligned}
\Pi_{t^*} &= \left[\alpha\Pi_t^{-1} + \alpha\frac{T_c}{T_o}H_i^{\mathsf{T}}R_i^{-1}H_i + (1-\alpha)H_k^{\mathsf{T}}R_k^{-1}H_k\right]^{-1} \\
&= \left[\alpha\Pi_t^{-1} + \alpha\check{H}^{\mathsf{T}}\check{R}^{-1}\check{H}\right]^{-1} \\
&= \frac{1}{\alpha}\Pi_t - \frac{1}{\alpha}\Pi_t\check{H}^{\mathsf{T}}\left[\frac{1}{\alpha}\check{R} + \frac{1}{\alpha}\check{H}\Pi_t\check{H}^{\mathsf{T}}\right]^{-1}\check{H}\frac{1}{\alpha}\Pi_t \\
&= \frac{1}{\alpha}\Pi_t - \frac{1}{\sqrt{\alpha}}\Pi_t\check{H}^{\mathsf{T}}\left(\check{R} + \check{H}\Pi_t\check{H}^{\mathsf{T}}\right)^{-1}\check{H}\frac{1}{\sqrt{\alpha}}\Pi_t, \qquad (4.3)
\end{aligned}$$

where

$$\check{H} = \begin{bmatrix} H_i \\ H_k \end{bmatrix}, \quad \check{R} = \begin{bmatrix} \frac{T_o}{T_c}R_i & \\ & \frac{\alpha}{1-\alpha}R_k \end{bmatrix}.$$

The exact motion propagation update in a single interval is given by

$$\Pi_{i,t+1} = \Pi_{i,t^*} + \frac{T_c}{T_p}\left(\frac{T_p}{T_c}T\right)^2 Q = \Pi_{i,t^*} + \frac{T_p}{T_c}T^2 Q. \qquad (4.4)$$

Combining (4.3) and (4.4), the overall covariance update can be obtained as

$$\Pi_{t+1} = \check{F}\Pi_t\check{F}^{\mathsf{T}} + \frac{T_p}{T_c}T^2Q - \check{F}\Pi_t\check{H}^{\mathsf{T}}\left(\check{R} + \check{H}\Pi_t\check{H}^{\mathsf{T}}\right)^{-1}\check{H}\Pi_t\check{F}^{\mathsf{T}}, \qquad (4.5)$$

with $\check{F} = \frac{1}{\sqrt{\alpha}}I$. This overall covariance update is exactly a discrete-time Riccati recursion.

From the discrete-time Riccati recursion (4.5), the corresponding limiting continuous-time Riccati recursion can be obtained with the procedure in [SMG88]. To begin with, we let $\check{F}_c = \frac{1}{T}(\check{F} - I)$, and rewrite the update equation as

$$\frac{\Pi_{t+1} - \Pi_t}{T} = \check{F}_c\Pi_t + \Pi_t\check{F}_c^{\mathsf{T}} + \frac{T_p}{T_c}TQ - \Pi_t\check{H}^{\mathsf{T}}\left(T\check{R} + T\check{H}\Pi_t\check{H}^{\mathsf{T}}\right)^{-1}\check{H}\Pi_t + \frac{o(T)}{T}.$$

By taking $T \to 0$ while keeping $\check{F}_c$, $T\check{R}$, and $TQ$ constant, we obtain the limiting continuous-time Riccati recusion

$$\dot{\Pi}(t) = \check{F}_c\Pi(t) + \Pi(t)\check{F}_c^{\mathsf{T}} - \Pi(t)\check{H}^{\mathsf{T}}\check{R}_c^{-1}\check{H}\Pi(t) + Q_c. \qquad (4.6)$$

Since $T\check{R}$ and $TQ$ are constant, we can determine $\check{R}_c$ and $Q_c$ with their initial values. That is,

$$\check{R}_c = T_c\check{R} = \frac{1}{f_c}\check{R} = \begin{bmatrix} \frac{1}{f_o}R_i & \\ & \frac{1}{f_c}\frac{\alpha}{1-\alpha}R_k \end{bmatrix},$$

and

$$Q_c = \frac{T_p}{T_c} T_c Q = \frac{1}{f_p} Q.$$

The order in the original discrete-time recursion (4.3) does matter in the limiting continuous-time approximation. For example, if the order of communication and observation operations are interchanged, the resulting continuous-time recursion differs. While this effect is negligible in the following optimization, detailed analysis should be further explored.

## 4.3 Properties of the CARE

### 4.3.1 Convergence of the CARE

After establishing the relationship between the original discrete-time and the limiting continuous-time processes, we further require that the limiting continuous-time process inherits the convergence property from the original discrete-time counterpart. The convergence of estimation covariance in discrete-time modeling, which mainly depends on the observation and communication topologies, is discussed in the previous chapter, and we only consider the convergent case in this paper to emphasize the scheduling aspect.

**Theorem 3.** *If $(\check{F}, Q^{\frac{1}{2}})$ is stabilizable and $(\check{F}, H)$ is detectable in the discrete-time model, then both the discrete-time Riccati recursion (4.5) and the associated continuous-time Riccati recursion (4.6) converge to the solutions of the corresponding Riccati equations with positive semidefinite initial condition, respectively.*

Even though the conditions in Theorem 3 is only sufficient, they are very close to necessary condition with some tradeoff between initial condition and the requirement on controllability and observability. The details can be found in [KSH00].

With Theorem 3, once we assure the convergence in the discrete-time scenario, the convergence of the continuous-time recursion (4.6) is guaranteed, and the recursion converges to the solution $\Pi$ of the corresponding CARE

$$\check{F}_c\Pi + \Pi\check{F}_c^{\mathsf{T}} - \Pi\check{H}^{\mathsf{T}}\check{R}_c^{-1}\check{H}\Pi + Q_c = 0. \tag{4.7}$$

The solution $\Pi$ can then be interpreted as the estimation uncertainty of the particular configuration of the operable parameters, $f_p$, $f_o$, $f_c$ and CI parameter $\alpha$.

### 4.3.2 Effects of Parameters on $\Pi$

Even though (4.7) can be solved numerically without difficulty, the analytical effects of those parameters on $\Pi$ are still of great engineering concern. As all operations are associated with certain costs, one of the practical concern involves how should one allocate the operation resource to decrease the uncertainty $\Pi$ in the most efficient way. It can be shown that $\Pi$ decreases when increasing $f_p$ or $f_o$ based on the result in [FI01]. However, the effects of $f_c$ and $\alpha$ on $\Pi$ is less obvious. Conceptually, we want to know the implicit derivative of $\Pi$ with respect to $f_c$ and $\alpha$ in (4.7), which can be explicated by the Fréchet derivative in functional analysis [Zei95].

To begin with, the recursion (4.6) is rewritten as a function of operation parameters

$$G(\Pi, f_c, \alpha) = 2f_c\left(\frac{1}{\sqrt{\alpha}} - 1\right)\Pi - \Pi\check{H}^{\mathsf{T}}\check{R}_c^{-1}(f_c, \alpha)\check{H}\Pi + Q_c, \tag{4.8}$$

and the CARE (4.7) with solution $(\Pi_0, f_{c,0}, \alpha_0)$ is exactly

$$G(\Pi_0, f_{c,0}, \alpha_0) = 0. \tag{4.9}$$

#### 4.3.2.1 The communication rate

By applying the implicit function theorem of the Fréchet derivative on (4.9),

$$G_{f_c}\Delta f_c + G_\Pi \Pi_{f_c}\Delta f_c = 0.$$

With some manipulation, the Fréchet derivative of $\Pi_{f_c}\Delta f_c$ at $(\Pi_0, f_{c,0}, \alpha_0)$ is given by

$$\Pi_{f_c}\Delta f_c = -G_\Pi^{-1}G_{f_c}\Delta f_c. \tag{4.10}$$

For the first operator, $G_{f_c}\Delta f_c$ is the Fréchet derivative at $(\Pi_0, f_{c,0}, \alpha_0)$, given by

$$G_{f_c}\Delta f_c = 2\left(\frac{1}{\sqrt{\alpha_0}} - 1\right)\Pi_0 \Delta f_c + G_{\check{R}_c}\check{R}_{c,f_c}\Delta f_c,$$

where

$$G_{\check{R}_c}\Delta\check{R}_c = \Pi_0 \check{H}^{\mathsf{T}}\check{R}_{c,0}^{-1}\Delta\check{R}_c\check{R}_{c,0}^{-1}\check{H}\Pi_0,$$

with $\check{R}_{c,0} = \check{R}_c(f_{c,0}, \alpha_0)$ and

$$\check{R}_{c,f_c}\Delta f_c = \begin{bmatrix} 0 & \\ & -\frac{1}{f_{c,0}^2}\frac{\alpha_0}{1-\alpha_0}R_k \end{bmatrix}\Delta f_c.$$

In the following, the operator $G_\Pi^{-1}\Delta\Pi$ is the inverse of the operator

$$G_\Pi \Delta\Pi = F_{\mathrm{cl},0}\Delta\Pi + \Delta\Pi F_{\mathrm{cl},0}^{\mathsf{T}},$$

and

$$F_{\mathrm{cl},0} = \check{F}_c(\alpha_0) - \Pi_0\check{H}^{\mathsf{T}}\check{R}_{c,0}^{-1}\check{H}.$$

Therefore, the output of the operator $G_\Pi^{-1}\Delta\Pi$ is the solution $X$ of the Lyapunov equation

$$F_{\mathrm{cl},0}X + XF_{\mathrm{cl},0}^{\mathsf{T}} = \Delta\Pi.$$

To sum up, the output of the operator $\Pi_{f_c}\Delta f_c$ is the solution $X$ of the Lyapunov equation

$$\begin{aligned} F_{\mathrm{cl},0}X + XF_{\mathrm{cl},0}^{\mathsf{T}} &= -G_{f_c}\Delta f_c \\ &= -2\left(\frac{1}{\sqrt{\alpha_0}} - 1\right)\Pi_0\Delta f_c - G_{\check{R}_c}\check{R}_{c,f_c}\Delta f_c. \end{aligned} \tag{4.11}$$

To interpret the derivative $\Pi_{f_c}\Delta f_c$, we let the solution $X = X_1 + X_2$ in (4.11), where $X_1$ is the solution of

$$F_{\mathrm{cl},0}X_1 + X_1F_{\mathrm{cl},0}^{\mathsf{T}} = -2\left(\frac{1}{\sqrt{\alpha_0}} - 1\right)\Pi_0\Delta f_c$$

and $X_2$ is the solution of

$$F_{\mathrm{cl},0}X_2 + X_2F_{\mathrm{cl},0}^{\mathsf{T}} = -G_{\check{R}_c}\check{R}_{c,f_c}\Delta f_c.$$

For $\Delta f_c \geq 0$, or by increasing communication rate $f_c$,

$$-2\left(\frac{1}{\sqrt{\alpha_0}} - 1\right)\Pi_0 \Delta f_c \leq 0,$$

and

$$-G_{\check{R}_c}\check{R}_{c,f_c}\Delta f_c \geq 0.$$

Since $(\check{F}_c, Q^{\frac{1}{2}})$ is stabilizable and $(\check{F}_c, H)$ is detectable, $F_{\text{cl},0}$ is stable [KSH00]. Therefore, as the solution of Lyapunov equation, we have $X_1 \geq 0$ and $X_2 \leq 0$. That is, by increasing the communication rate $f_c$, the increasing part of $\Pi$ is characterized by $X_1$ with the effect of $\check{F}_c$, and the decreasing part is given by $X_2$ with the effect of $R_k$. In other words, the communication update provides the information from $H_k^{\mathsf{T}} R_k^{-1} H_k$ with the cost that the original covariance is enlarged at the same time. This analysis not only substantiates the qualitative understanding, but also characterizes quantitative effect as well.

#### 4.3.2.2 The CI coefficient

The same analysis applied on the CI coefficient is direct. At $(\Pi_0, f_{c,0}, \alpha_0)$,

$$\Pi_\alpha \Delta\alpha = -G_\Pi^{-1} G_\alpha \Delta\alpha, \tag{4.12}$$

with

$$G_\alpha \Delta\alpha = -f_{c,0}\alpha_0^{-\frac{3}{2}}\Pi_0\Delta\alpha + G_{\check{R}_c}\check{R}_{c,\alpha}\Delta\alpha,$$

and

$$\check{R}_{c,\alpha}(f_{c,0}, \alpha_0)\Delta\alpha = \begin{bmatrix} 0 & \\ & \frac{1-\alpha_0+\alpha_0^2}{f_{c,0}(1-\alpha_0)^2} R_k \end{bmatrix} \Delta f_c.$$

Similarly, we can write $\Pi_\alpha \Delta\alpha = Y_1 + Y_2$ with

$$F_{\text{cl},0} Y_1 + Y_1 F_{\text{cl},0}^{\mathsf{T}} = f_{c,0}\alpha_0^{-\frac{3}{2}}\Pi_0\Delta\alpha$$

and

$$F_{\text{cl},0} Y_2 + Y_2 F_{\text{cl},0}^{\mathsf{T}} = -G_{\check{R}_c}\check{R}_{c,\alpha}\Delta\alpha.$$

With $\Delta\alpha \geq 0$, we have $Y_1 \leq 0$ and $Y_2 \geq 0$. Increasing $\alpha$ represents emphasizing the internal estimate over received data at the communication step, which decreases $\Pi$ with the effect of $\check{F}_c$ but increases $\Pi$ by that of $H_k^\top R_k^{-1} H_k$.

## 4.4 Optimal Scheduling Problems

To investigate localization accuracy as well as the associated costs, we formulate two optimization problems: to minimize the cost with predetermined covariance constraints, or to minimize the covariance with a resource limit. In the localization algorithm, one can design the observation rate $f_o$ and the communication rate $f_c$, as well as the CI parameter $\alpha$.

### 4.4.1 Cost Minimization Problem

We first consider the optimal scheduling problem that aims to minimize the overall cost of the localization algorithm, while maintaining a specified accuracy requirement.

$$\begin{aligned} \underset{f_o, f_c, \alpha}{\text{minimize}} \quad & \mu_o f_o + \mu_c f_c \\ \text{subject to} \quad & 2f_c\left(\frac{1}{\sqrt{\alpha}} - 1\right)\Pi + Q_c - \Pi \check{H}^\top \check{R}_c^{-1}(f_o, f_c, \alpha)\check{H}\Pi = 0 \\ & \mathsf{tr}(\Pi) \leq \pi_{\max} \\ & f_o \leq f_{o,\max} \\ & f_{c,\min} \leq f_c \leq f_{c,\max} \\ & 0 < \alpha < 1. \end{aligned} \tag{4.13}$$

In (4.13), $\mu_o$ and $\mu_c$ are the costs associated with the observation and communication rates, respectively. The uncertainty criterion is established with the first two constrains. The stationary estimation covariance $\Pi$ is characterized by the CARE (4.7), and the accuracy requirement is presented by the second constrain, where the trace of the stationary estimation covariance should be bounded by $\pi_{\max}$. The maximum observation rate $f_{o,\max}$ and the maximum communication rate $f_{c,\max}$ represent the physical limitation of the underlying operation. The last constraint is the requirement from CI fusion on $\alpha$.

The inequality $f_{c,\min} \leq f_c$ is imposed to account for the imperfection of the continuous-time approximation. According to the investigation in Section V, increasing the communication rate $f_c$ may lead to the increase of $\Pi$, and thus the optimal $f_c$ may be relatively small; this results in large amplitude steps in the covariance of the corresponding discrete-time setting. With this large sawtooth behavior in the discrete-time case, our continuous-time approximations, including the reordering of operations especially, become suspect.

### 4.4.2 Trace Minimization Problem

We can also establish the optimal scheduling problem by minimizing the localization uncertainty given an overall cost budget $\mu_{\max}$, as shown in the following:

$$\begin{aligned} \underset{f_o, f_c, \alpha}{\text{minimize}} \quad & \mathsf{tr}(\Pi) \\ \text{subject to} \quad & 2f_c\left(\frac{1}{\sqrt{\alpha}} - 1\right)\Pi + Q_c - \Pi \check{H}^\mathsf{T} \check{R}_c^{-1}(f_o, f_c, \alpha)\check{H}\Pi = 0 \\ & \mu_o f_o + \mu_c f_c \leq \mu_{\max} \\ & f_o \leq f_{o,\max} \\ & f_{c,\min} \leq f_c \leq f_{c,\max} \\ & 0 < \alpha < 1. \end{aligned} \tag{4.14}$$

Since the feasible set of parameters is bounded, the global optimum of both optimization problems can be obtained by exhaustive search. More sophisticated solving techniques may be explored using a further understanding of the properties of both optimization problems.

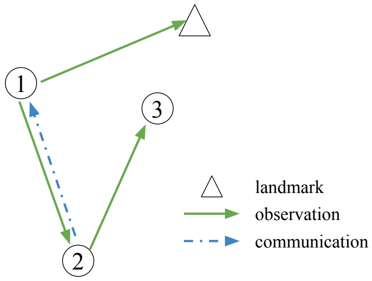


Figure 4.1: The topology of the simulated multirobot system. We will discuss the optimal scheduling on robot 1, since it has both observation and communication.

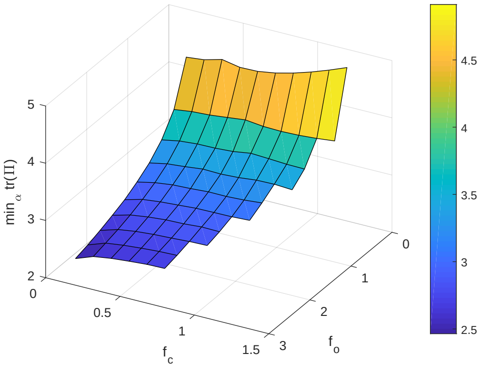


Figure 4.2: The plot of $\min_\alpha \mathsf{tr}(\Pi)$ in the region with feasible operation rates. We can see that the trace decreases with the increase of the observation rate $f_o$. The trace does not necessarily decreases with the communication rate $f_c$, as discussed.

## 4.5 An Example of Optimal Scheduling

We demonstrate the effectiveness of reducing the operation cost or achieving better estimation performance by solving the optimization problems on a simulated system with parameters listed in Table I. We consider three robots, indexed by 1, 2, and 3. Robot 1 can observe the landmark and robot 2; robot 2 can observe robot 3 and sends its information to robot 1, as the complete topology in Fig. 4.1. Based on the analysis in the previous chapter, the estimation covariance of robot 1 is bounded, even though that of robot 2 is not.

Table 4.1: Parameters of a real scenario for scheduling examples

| | | |
|---|---|---|
| observation cost | $\mu_o$ | 20 mW |
| maximum observation rate | $f_{o,\max}$ | 3 Hz |
| maximum observation distance | $d_{\max}$ | 4.8 m |
| the variance of range sensing | $\sigma_d^2$ | 0.0215 $\mathrm{m}^2$ |
| the variance of bearing sensing | $\sigma_\phi^2$ | 0.01 $\mathrm{rad}^2$ |
| communication cost | $\mu_c$ | 100 mW |
| maximum communication rate | $f_{c,\max}$ | 10 Hz |
| minimum communication rate | $f_{c,\min}$ | 0.2 Hz |
| propagation rate | $f_p$ | 10 Hz |

A typical example scenario is presented as a baseline with $f_c = 1$ Hz, $f_o = 2$ Hz. The total cost is $\mu = 140$ mW, and the trace of stationary covariance is $\mathsf{tr}(\Pi) = 2.0918\ \mathrm{m}^2$ with an optimized $\alpha = 0.90$. With this baseline example, we can consider both the cost minimization problem (4.13) with the same performance criterion, and the trace minimization problem (4.14) with the same amount of cost budget $\mu_{\max} = 140$ mW, respectively.

We plot the trace of the stationary covariance with optimized $\alpha$ in Fig. 4.2, and we arrive the optimal solution for each optimization problem. The optimized parameters and the optimization result of two scenarios are listed in Table 4.2 for comparison. After solving the cost minimization problem, the cost reduction is around 64% for the same error performance, achieved by reducing the communication rate $f_c$ and CI parameter $\alpha$ jointly. In the trace minimization problem, we can instead reduce 15% of the stationary covariance trace within the same original resource budget, in fact simultaneously decreasing cost by 43%.

With the parameters obtained in continuous-time approximation, we can apply them in the original discrete-time setting. The simulation with baseline scenario, cost minimization and trace minimization is presented in Fig. 4.3, where the operation parameters are obtained from Table II. The estimation error is defined as $||\hat{s}^1 - s||_2$. We plot an average over 10

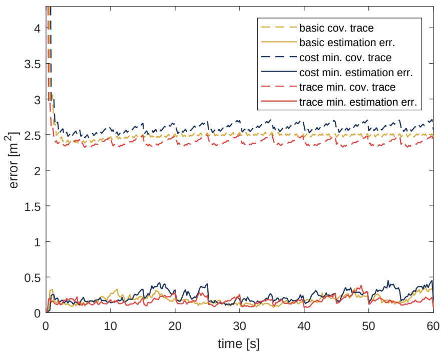


Figure 4.3: The discrete-time simulation with parameters from optimal scheduling result. This simulation shows that even though we are optimizing the covariance upper bound, the real covariance as well as the position error show similar trend.

Table 4.2: Results for scheduling examples in continuous-time approximation

| scenario | $f_o$ [Hz] | $f_c$ [Hz] | $\alpha$ | $\mu$ [mW] | $\mathsf{tr}(\Pi)$ [m$^2$] |
|---|---|---|---|---|---|
| baseline | 2 | 1 | 0.90 | 140 | 2.9018 |
| cost min (4.13) | 1.52 | 0.2 | 0.63 | 50.4 | 2.9003 |
| trace min (4.14) | 3 | 0.2 | 0.47 | 80 | 2.4617 |

sampled curves in Fig. 4.3 and present the result in Table 4.3, where the estimation error and the covariance trace are averaged over the operation period. From Fig. 4.3, as $f_c = 0.2$ in both optimized case, we can observe that the error curves in these two cases follow a sawtooth profile in accordance with the communication period. This motivates the constraint $f_{c,\min}$ in the optimization problems (4.13) and (4.14). Also, the discrete-time realization reflects the design goal in the continuous-time counterpart. The two examples show the desired improvement in terms of the cost and of the stationary trace by considering the optimal scheduling problems.

In the trace optimization problem, the communication rate is small comparing to the baseline case. The saving of communication operation with the benefit of trace improvement may seem confusing at first. However, since the communication update applies covariance intersection, the multiplication of $\alpha$ does increase the covariance. This phenomenon can also be explained by the discussion in Section V. With the analysis of the Fréchet derivative, from $f_c = 0.2$ to $f_c = 1$, the contribution of communication incoming information is compensated by the uncertainty introduced by CI algorithm. This example not only demonstrates the improvement by simply engineering the parameters, it also shows the necessity of optimizing operation parameters in such scheme.

While this example presents a minimum topology to clearly demonstrate our algorithm, the extension to general topologies is direct. Equation (4.3) can account for arbitrary observation topology without difficulty via equations (4.1) and (4.2). For the communication topology, two types of generalization are considered for completeness. If the incoming information contains other communicated data, (4.3) can simply be extended to include the additional terms. If there are multiple incoming communication paths, from several robots transmitting their data to the receiving node, CI will use a set of convex coefficients instead of a single coefficient $\alpha$ for such data fusion. In this case, the number of optimization variables will increase, but the formulation remains unaltered.

One should note that the operation periods, for example $T_o$ and $T_c$, may not have a common divisor in general, which makes discrete-time analysis intractable. Nevertheless, as demonstrated by this example, we show that it is straightforward to design and to analyze these operation periods in a corresponding continuous-time limiting approximation. The tightness of the upper bound mainly depends on the knowledge of odometry and observation error. One can further improve those bounds with more available information, and optimize the localization algorithm following the procedure in this paper without changing any step.

Table 4.3: Results for scheduling examples in discrete-time simulation

| scenario | estimation error [m$^2$] | cov. trace [m$^2$] | power [mW] |
|---|---|---|---|
| baseline | 0.1894 | 2.5051 | 140 |
| cost min (4.13) | 0.2365 | 2.6251 | 50.4 |
| trace min (4.14) | 0.1615 | 2.4070 | 80 |

# CHAPTER 5

# Block Online EM SLAM

Simultaneous localization and mapping (SLAM) enables a robot to obtain such spatial relationships intelligently and autonomously [CCC16, BAY17, SRD21]. After 30 years of extensive study, we now have sophisticated SLAM algorithms for visual-inertial systems that achieve unprecedented performance, for example ORB SLAM series [MMT15, MT17b, MT17a] and VINO-Mono [QLS18].

Typically, a visual-inertial SLAM system is divided into a frontend and a backend. The frontend abstracts sensor data, and gives a local trajectory estimation. While the trajectory from the frontend has inevitable drift, the backend takes the results from the frontend, and computes a globally consistent trajectory and map with loop closure. In fact, the SLAM backend is the computation bottleneck for obtaining globally consistent trajectory and map, and it often operates offline from the frontend process [CCC16].

The reason that most of the current backend algorithms are offline mainly lies in the fact that they solve the SLAM problem as a maximum likelihood (ML) problem of the trajectory. This ML problem is formulated as a nonlinear least squares (NLS) problem, and its computational cost and storage requirements grow significantly with respect to the time duration. The NLS formulation does not have sound methods to remove the historical data. Existing solutions try to downsize the optimization problem by pruning out or culling less informative nodes, but the overall formulation remains offline [CCC16]. In this paper, we propose an online backend algorithm that can summarize the past data into sufficient statistics, and then are no longer needed, which will greatly improve the efficiency and thus the scalability of the entire SLAM systems.

To develop online backend algorithms, instead of solving the SLAM problem as a pure optimization problem, we first follow the EM SLAM that models the SLAM problem as a ML problem of landmark positions with the trajectory as latent variables [SSG17]. Furthermore, by acknowledging that the SLAM problem follows a general state hidden Markov model (HMM), we can apply the block online EM (BOEM) algorithm for the ML problem [LF13], and formulate BOEM SLAM. In essence, the historical data are summarized into the landmark estimates and the most recent state distribution, and each iteration only takes place within non-overlapping blocks. Since past statistics can be summarized and discarded, the BOEM SLAM is online. The authors of the BOEM algorithm used the algorithm in a simple 2D scenario [LFM12] and in wireless sensor networks [DL13, DL14], but the systems they considered are fairly simple. Incorporating visual input, we are the first to extend the BOEM SLAM in 3D visual-inertial systems to the best of our knowledge.

In particular, we apply the BOEM SLAM in visual-inertial systems with a monocular camera and inertial measurement unit (IMU). Even though the EM SLAM lays out the framework for our BOEM SLAM, several critical implementation issues in visual-inertial systems are barely addressed in the original EM SLAM paper [SSG17]. First, as the E-step contains the Kalman filter and the RTS smoother, it is necessary to expand the filtering methods to consider rotation estimation, which requires a Lie group representation. Second, visual measurements occasionally contain outliers, which greatly deteriorate SLAM performance. Similar to the robust cost function in optimization-based methods, we propose the outlier removal approach to reject spurious measurements.

We compare SLAM backend algorithms in both synthetic data and the EuRoC datasets, and show that BOEM SLAM achieves comparable estimation accuracy with a significantly lower computational cost compared to other backend solutions. In simulation, we highlight the relationship between the increasing time interval and the resulting computational cost. The realistic scenario considers the fact that data keep arriving in robotic applications, and therefore online SLAM algorithms are essential for better efficiency and scalability. We further run the BOEM SLAM on EuRoC datasets to show that the BOEM SLAM algorithm is applicable in real visual-inertial systems.

## 5.1 Review on SLAM Algorithms

In order to understand the offline nature of SLAM algorithms, we systematically review various SLAM backend algorithms. As these algorithms directly estimate globally consistent trajectory, they are referred to the SLAM algorithm before the clear separation of frontend and backend in visual-inertial systems. For readers that are interested in the entire SLAM development, please refer to SLAM survey papers [CCC16, BAY17, SRD21].

Modern SLAM algorithms and systems have converged to a common underlying model [CCC16]. In particular, landmarks are treated as parameters in the modern model. While these algorithms admit the same model, we classify them by two criteria:

- whether the trajectory is solved as parameters or as distributions, and
- whether the map is explicitly obtained or not.

### 5.1.1 Trajectory as Parameters

This category directly solves the ML problem of the trajectory in an optimization problem. With noise assumed Gaussian, the optimization problem leads to a NLS problem, and there are several numerical tools designed to solve NLS problems, including g$^2$o [KGS11] and Ceres [AMO]. Therefore, this category is often referred as the *optimization-based* algorithms. We can further classify the algorithms in this category by the second criterion on map.

#### 5.1.1.1 Explicit Map

When solving the map explicitly, the landmark positions and the trajectory are jointly formulated as the parameters in the optimization problem. The $\sqrt{\text{SAM}}$ is considered one of the first successful implementations in this category [DK06]. Due to the sparsity of the optimization problem, the $\sqrt{\text{SAM}}$ can solve the optimization problem efficiently by factorizing matrices into square root form. Built on $\sqrt{\text{SAM}}$, the iSAM is later proposed to improve the efficiency by using QR factorization [KRD08]. The Bayes tree in introduced in iSAM2 to further enhance the efficiency, especially on the incremental factorization [KJR11].

These algorithms are able to do incremental update in the exploration stage without loop closure. In other words, only the affected parameters are updated but not all the parameters. However, the underlying optimization problem remains unchanged, and thus these algorithms are still offline. Particularly, the entire trajectory has to be stored, and is updated all together after loop closure.

#### 5.1.1.2 Implicit Map

Instead of using the exteroceptive information directly, there are ways to formulate an optimization problem without landmark parameters. Observations with overlapping landmarks can be transformed into constraints among states that take those observations. In other words, the exteroceptive information is turned into the constraints among states, and the resulting optimization problem only has the trajectory as parameters.

The entire optimization problem can be regarded as a graph with states in the trajectory as nodes, and constraints as edges. Since the transformed constraints are imposed between nodes, this category is also known as the *pose graph SLAM*. This concept is introduced very early in the seminal paper by [LM97], and is later improved in [TM06] and [GKS10]. Visual-inertial systems also heavily rely on the pose graph methods to refine the trajectory estimate after loop closures.

### 5.1.2 Trajectory as Distributions

SLAM algorithms can also solve the trajectory as distributions, and accordingly they have to use filtering techniques. Similarly, we can further separate the algorithms in this category by whether the landmarks are explicitly solved or treated as relative constraints.

#### 5.1.2.1 Explicit Map

As landmark positions are treated as unknown parameters in the system, optimizing these parameters with latent variables can be considered as an ML problem, and can be solved by the EM algorithm. Following this idea, [SSG17] propose the EM SLAM. In particular, the E-step uses filtering methods to estimate the trajectory while the M-step optimizes the landmark positions.[1] Since the EM algorithm is an offline algorithm, EM SLAM is also offline. The proposed BOEM SLAM relies on this EM framework, but takes advantage of the HMM structure to enable an online solution.

#### 5.1.2.2 Implicit Map

The similar concept of turning observation into constraints is also applicable for this category, but constraints are used in the observation update in the Kalman filter instead. This approach is first proposed in [ESL06], and is called the exactly sparse delayed state filter (ESDSF). An extension to Lie group is later presented in [LCM18]. When establishing constraints between states, the ESDSF encounters the same issue as mentioned in the pose graph SLAM where old states should be maintained. As a consequence, even though the Kalman filter is an online algorithm, the overall implementation of the ESDSF is offline.

[1] We avoid the traditional dichotomy between filtering-based and optimization-based algorithms, since EM SLAM naturally integrates both.

### 5.1.3 Discussion

There are two reasons that all aforementioned algorithms are offline. First, for the algorithms with implicit landmarks, they need to store the historical state in order to realize the relative constraints. Even for the algorithms with explicit landmarks, the existing algorithms use offline methods, including the NLS optimization and the EM algorithm. In other words, there is no way that the old states can be discarded. The proposed BOEM SLAM belongs to the same category of the EM SLAM, but it solves the SLAM problem online by leveraging the HMM structure.

## 5.2 Parameter Inference in HMMs

We summarize the parameter inference problem by EM algorithms in this section. The thorough treatment of the EM algorithm and the block online EM algorithm can be found in their original papers [DLR77] and [LF13], respectively. Formally, we consider a HMM with latent states $\{\mathbf{s}_t\}_{t\geq 0}$, and the observation $\{\mathbf{o}_t\}_{t\geq 1}$. The unknown parameter $\theta$, which is in the state transition model and/or the observation model, will be inferred by the EM algorithm.

### 5.2.1 Standard EM Algorithm

The complete-data log-likelihood function of the HMM is

$$\log p(s_{0:T}, o_{1:T}; \theta) = \log p(s_0) + \sum_{t=1}^{T} \log p(s_t, o_t | s_{t-1}; \theta).$$

The EM algorithm computes the ML solution of the unknown parameter $\theta$ by two iterative steps: The E-step constructs the function $Q(\theta)$ and the M-step finds the $\theta$ that optimizes $Q(\theta)$. Typically, the function $Q(\theta)$ is the conditional expectation of the complete data log-likelihood given the observations. To ease the transition to the online setting, we write the

E-step by normalizing the conditional expectation with $T$ as

$$Q_k(\theta) = \frac{1}{T} \mathsf{E}_{\hat{\theta}_{k-1}} \left[ \sum_{t=1}^{T} \log p(\mathbf{s}_t, \mathbf{o}_t | \mathbf{s}_{t-1}; \theta) \middle| \mathbf{o}_{1:T} \right]. \tag{5.1}$$

The M-step directly solves the optimal $\theta$ by

$$\hat{\theta}_k = \arg\max_{\theta} Q_k(\theta). \tag{5.2}$$

### 5.2.2 Block Online EM Algorithm (BOEM) for HMMs

The BOEM algorithm performs the EM algorithm in non-overlapping blocks. The only historical information needed in each block is the distribution of the previous state, while the rest of the past statistics are not required. This property makes the BOEM algorithm an online algorithm in essence.

The BOEM algorithm requires the block size to grow due to the dependency in the Markov chain. In the original paper, the block size $\{\tau_n\}_{n\geq 1}$ is taken as $\tau_n = \lfloor cn^a \rfloor$, with $c > 0$ and $a > 1$. For convenience, we also define $T_n$ as the end time of the block $n$, or $T_n = \sum_{m=1}^{n} \tau_m, \quad T_0 = 0$. We now present the BOEM algorithm in block $n$. The E-step computes a function similar to (5.1), but only takes the states and the observations within the block.

$$\bar{Q}_n(\theta) = \frac{1}{\tau_n} \mathsf{E}_{\hat{\theta}_{n-1}}^{\chi_{n-1}} \left[ \sum_{t=T_n+1}^{T_{n+1}} \log p(\mathbf{s}_t, \mathbf{o}_t | \mathbf{s}_{t-1}; \theta) \middle| \mathbf{o}_{T_n+1:T_{n+1}} \right]. \tag{5.3}$$

The initial distribution of this block, or the distribution of $\mathbf{s}_{T_n}$ is given by $\chi_{n-1}$. The M-step remains the same:

$$\hat{\theta}_n = \arg\max_{\theta} \bar{Q}_n(\theta). \tag{5.4}$$

To finish the calculation in this block, we need to provide $\chi_n$ for the next block. While $\chi_n$ represents the distribution of $\mathbf{s}_{T_{n+1}}$ with the current parameter estimate $\hat{\theta}_n$, $\chi_n$ can by obtained by starting with $\chi_{n-1}$ and calculating the distribution of $\mathbf{s}_{T_{n+1}}$ conditioned on $\mathbf{o}_{T_n+1:T_{n+1}}$.

The convergence of the BOEM algorithm is discussed in the original paper [LF13] under fairly restrictive conditions. The strong mixing condition requires the distribution function to be larger than a positive constant, which is too strong even for the linear Gaussian model. Even the convergence of the BOEM algorithm in the linear Gaussian model is shown numerically without proof in [LF13]. While most SLAM works use Gaussian noise for modeling, the current convergence analysis is not ready for SLAM applications, including [LFM12] and this work. We expect the analysis to be more complete and general in the future.

## 5.3 Block Online EM (BOEM) SLAM

In this section, we apply the BOEM algorithm to solve the SLAM problem of a visual-inertial system. It is not difficult to recognize the HMM structure in the SLAM problem, but we need to take care of the manifold nature of the SLAM problem, which will be emphasized in this section.

### 5.3.1 System Model

A mobile agent can be described by the state space model,

$$\mathbf{s}_{t+1} = f(\mathbf{s}_t, u_t) + \mathbf{w}_t, \tag{5.5}$$

$$\mathbf{o}_t = h(\mathbf{s}_t, \lambda) + \mathbf{v}_t. \tag{5.6}$$

As the noises $\mathbf{w}_t$ and $\mathbf{v}_t$ are independent, the states $\{\mathbf{s}_t\}_{t\geq 0}$ together with the observations $\{\mathbf{o}_t\}_{t\geq 1}$ admits a HMM.

We now specify the model in (5.5) and (5.6) for a visual-inertial system. The state $s_t$ includes the rotation, velocity and position, or $s_t^\mathsf{T} = [q_t^\mathsf{T}, v_t^\mathsf{T}, p_t^\mathsf{T}]$. The rotation $q_t \in S^3$ is represented as a unit quaternion. For rotation disturbance, we use the right-plus $\oplus$

convention in [SDA20], which leads to

$$\mathbf{q}_t = \bar{q}_t \oplus \mathbf{e}_t = \bar{q}_t \circ \exp_{\mathsf{q}}(\mathbf{e}_t). \tag{5.7}$$

This definition decomposes the rotation $\mathbf{q}_t$ into the deterministic term $\bar{q}_t$ and the random term $\mathbf{e}_t$ in the tangent space.

The time dynamic model (5.5) describes the state transition with IMU measurements $u_t$, including gyroscopes and accelerometers. With fairly common assumptions [KHS17], we use the model:

$$\begin{aligned}
\mathbf{q}_{t+1} &= \mathbf{q}_t \circ \exp_{\mathsf{q}}\left(\Delta t(u_{\omega,t} - \delta_{\omega,t}) + \mathbf{w}_{q,t}\right), \\
\mathbf{v}_{t+1} &= \mathbf{v}_t + \Delta t\left(R_{\mathsf{q}}(\mathbf{q}_t)(u_{a,t} - \delta_{a,t}) + g\right) + \mathbf{w}_{v,t}, \\
\mathbf{p}_{t+1} &= \mathbf{p}_t + \Delta t\, \mathbf{v}_t + \frac{\Delta t^2}{2}\left(R_{\mathsf{q}}(\mathbf{q}_t)(u_{a,t} - \delta_{a,t}) + g\right) + \mathbf{w}_{p,t},
\end{aligned} \tag{5.8}$$

where $R_{\mathsf{q}}(\cdot)$ maps a quaternion to the corresponding rotation matrix. This model includes the IMU biases $\delta_{\omega,t}$ and $\delta_{a,t}$. The zero-mean Gaussian noise $\mathbf{w}_t^{\mathsf{T}} = [\mathbf{w}_{q,t}^{\mathsf{T}}, \mathbf{w}_{v,t}^{\mathsf{T}}, \mathbf{w}_{p,t}^{\mathsf{T}}]$ can be derived from the IMU measurement noises, and we denote the covariance matrix of $\mathbf{w}_t$ by $Q$.

For the observation model (5.6), the monocular camera is used as the exteroceptive sensor [MSK03]. We use the standard pinhole camera model, and the normalized camera measurement of landmark $m$ at time $t$ is given by

$$\mathbf{o}_{m,t} = \pi(T^{cn}(\lambda_m)) + \mathbf{v}_{m,t}. \tag{5.9}$$

The projection $\pi$ is defined as $\pi([x, y, z]^{\mathsf{T}}) = [x/z, y/z]^{\mathsf{T}}$, and $T^{cn}$ is the transformation from the navigation frame to the camera frame. Let $M_t$ denote the sets of landmarks observed at time $t$. The entire observation at time $t$ is

$$\mathbf{o}_t = \begin{bmatrix} \vdots \\ \mathbf{o}_{m,t} \\ \vdots \end{bmatrix}_{m \in M_t} = \begin{bmatrix} \vdots \\ \pi(T^{cn}(\lambda_m)) \\ \vdots \end{bmatrix}_{m \in M_t} + \mathbf{v}_t, \tag{5.10}$$

where $\mathbf{v}_t$ is zero-mean Gaussian with covariance matrix $R_t$.

### 5.3.2 Algorithm

When solving the SLAM problem in the EM framework, we aim to optimize the parameter $\lambda$ given the available observation data $\{o_t\}_{t\geq 1}$. While the application of the BOEM algorithm is direct, we take a closer look at the calculation procedure for the BOEM SLAM.

By combining the E-step (5.3) and the M-step (5.4) in the BOEM algorithm, we have

$$\begin{aligned}
\hat{\lambda}_n &= \arg\max_{\lambda} \mathsf{E}_{\hat{\lambda}_{n-1}}^{\chi_{n-1}} \left[ \sum_{t=T_n+1}^{T_{n+1}} \log p(\mathbf{s}_t, \mathbf{o}_t | \mathbf{s}_{t-1}; \lambda) \middle| \mathbf{o}_{T_n+1:T_{n+1}} \right] \\
&= \arg\min_{\lambda} \mathsf{E}_{\hat{\lambda}_{n-1}}^{\chi_{n-1}} \left[ \sum_{t=T_n+1}^{T_{n+1}} \|\mathbf{o}_t - h(\mathbf{s}_t, \lambda)\|_{R_t}^2 \middle| \mathbf{o}_{T_n+1:T_{n+1}} \right] \\
&\approx \arg\min_{\lambda} \sum_{t=T_n+1}^{T_{n+1}} \left\| o_t - h\left( \mathsf{E}_{\hat{\lambda}_{n-1}}^{\chi_{n-1}} \left[ \mathbf{s}_t \middle| \mathbf{o}_{T_n+1:T_{n+1}} \right], \lambda \right) \right\|_{R_t}^2 .
\end{aligned}$$

The notation $\|x\|_M^2 = x^\mathsf{T} M^{-1} x$ is used for the squared Mahalanobis distance with covariance matrix $M$, and the approximation in the last equation is taken by interchanging the expectation and the entire function. In each block, the BOEM SLAM first calculates the conditional expectation $\mathsf{E}_{\hat{\lambda}_{n-1}}^{\chi_{n-1}} \left[ \mathbf{s}_t \middle| \mathbf{o}_{T_n+1:T_{n+1}} \right]$ by the Kalman filter and the RTS smoother [RTS65]. In the following, obtaining $\hat{\lambda}_n$ is exactly to solve a NLS problem, thanks to the Gaussian noise assumption. The BOEM SLAM algorithm can be summarized as

- E-step: calculate $\mathsf{E}_{\hat{\lambda}_{n-1}}^{\chi_{n-1}} \left[ \mathbf{s}_t \middle| \mathbf{o}_{T_n+1:T_{n+1}} \right]$ with the Kalman filter and the RTS smoother.
- M-step: solve the NLS problem to get $\hat{\lambda}_n$.
- End of the block: obtain $\chi_n$ for the next block.

Overall, the EM framework naturally approaches the SLAM problem by tacking two sub-problems separately. As the E-step keeps the landmark parameters constant, it can be regarded as the localization step to improve the trajectory estimate; meanwhile, the M-step can be considered as the mapping step to optimize the landmark parameters. In BOEM SLAM, the past information is not explicitly used, but is implicitly incorporated in $\chi_{n-1}$

and also in $\hat{\lambda}_{n-1}$. When the past information can be summarized and discarded, the BOEM SLAM algorithm greatly decreases the amount of data required in calculation, and provides an online solution.

### 5.3.3 Implementation Details in the Visual-Inertial System

We now turn to the implementation aspects of the BOEM SLAM. The calculation of the conditional expectation in the E-step is typical in filtering and smoothing problems, which are typically solved by the Kalman filter and the RTS smoother. One of the main challenge is that we have to take the manifold structure of the rotation into consideration. The Kalman filer on Lie group is well-developed [BMG13], while the RTS smoother on Lie group receives little discussion. We then follow the derivatives on Lie groups in [SDA20] to design the RTS smoother on the Lie group.

The state space $\mathbb{S} = S^3 \times \mathbb{R}^3 \times \mathbb{R}^3$ is the direct product of three Lie groups, and thus it is also a Lie group. The group multiplication, the exponential and logarithm maps, and even the plus and minus operations, are all well-defined in $\mathbb{S}$ by inheriting from individual Lie groups. We use $\bar{s}_{t|t'}$ for conditional expectation of $s_t$ given $o_{1:t'}$, and $\Sigma_{t|t'}$ for the associated covariance matrix. With the derivatives on Lie groups, we can then use the following iterative equations

$$\bar{s}_{t|T} = \bar{s}_{t|t} \circ \exp\left(C_t\left(\bar{s}_{t+1|T} \ominus f(\bar{s}_{t|t}, u_t)\right)\right), \tag{5.11}$$

and

$$C_t = \Sigma_{t|t} F^{\mathsf{T}} (F \Sigma_{t|t} F^{\mathsf{T}} + Q)^{-1}, \tag{5.12}$$

for the estimate update in the RTS smoother, while the covariance update remains identical to the original RTS smoother.

Another key factor in the SLAM performance in the visual feature outlier detection and removal. In the optimization-based approaches, robust cost functions are often employed to reduce the effect of outliers, for example the Huber loss function. In our experiments, we implement the similar detection scheme to remove the outliers. We reject the visual measurements that are far from the predicted measurements. We also reject the observation update in the Kalman filter that changes the trajectory abruptly. While the current methods are straightforward, we believe that more sophisticated outlier removal methods can be proposed with more investigation.

## 5.4 Results

In this section, we compare the accuracies and the computational costs of different SLAM backend algorithms. We focus on the explicit-map algorithms, which simultaneously estimate the trajectory and the landmarks. In particular, we consider three algorithms: the optimization-based SLAM (opt.), the EM SLAM and the proposed BOEM SLAM.[2]

To ensure fair comparison of processing times, the simulations and the experiments are all performed on a single computer equipped with an Intel i5-8250U CPU @ 1.60 GHz and 8 GB of DDR4 RAM, and all optimization problems are solved by Ceres [AMO].

### 5.4.1 Simulations

We test all three SLAM algorithms on a circular trajectory with discrete time interval 30 ms as in Fig. 5.1. The camera on the robot is constrained to face outward to observe landmarks on the walls throughout the trajectory. We then generate the IMU measurements and camera observations for the robot with random noise. All the SLAM results together with the dead reckoning trajectory are shown in Fig. 5.1.

To verify the effectiveness of the BOEM SLAM, we plot the rotation and the position estimation errors of all SLAM algorithms over 50 sec in Fig. 5.2. All SLAM algorithms maintain reasonable estimation errors throughout this interval. The only noticeable differ-

[2] The source code and the parameters of the simulation and the experiment is available at `https://github.com/tsangkai/slam_demo`.

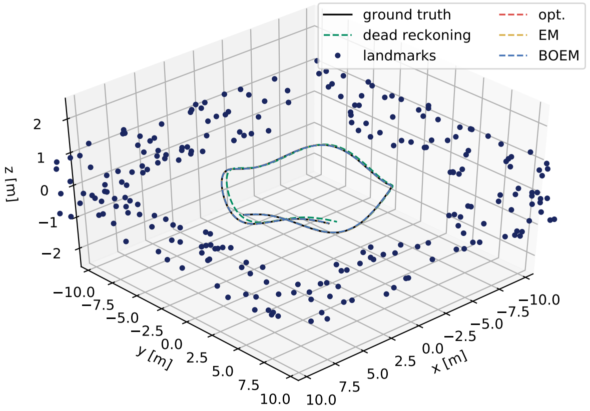


Figure 5.1: The first 16 sec of the simulated trajectories. Three SLAM algorithms are presented in dashed lines, including optimization-based algorithm (opt.), the EM SLAM and the BOEM SLAM. 200 landmarks are randomly generated on the walls of a $7.5 \times 7.5$ m box.

ence between BOEM SLAM and other algorithms is that the estimation trajectories from BOEM SLAM have larger variance. Since the BOEM SLAM processes less data in each time block, we expect that it needs more data for convergence compared to other offline algorithms. Overall, the estimation accuracy of these 3 algorithms are comparable.

To emphasize the importance of the online feature for robotic applications, we consider the computation time with various durations. The duration of the trajectory is increased by 15 sec, and the performance analyzing tool `perf` is used to calculate the processing time, with the results shown in Fig. 5.3. Even though all algorithms achieve similar estimation accuracy, the processing times are significantly different. While the NLS optimization procedure dominates the computation cost, an improvement of processing time can be achieved by decreasing the size of the optimization procedure, which makes the EM SLAM more efficient over the optimization-based SLAM. Furthermore, as an online algorithm, BOEM SLAM can effectively summarize and discard past data, and thus has the least computational cost. For the duration of 150 sec, while optimization SLAM needs more than 1000 sec, BOEM SLAM

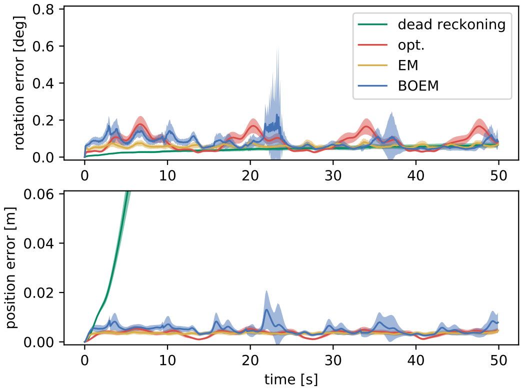


Figure 5.2: The rotation and the position errors of the SLAM algorithms. The shaded areas show 1 standard deviation error bar over 50 trials.

only takes around 20 sec to complete the task.

### 5.4.2 EuRoC Datasets

We then test the algorithms on the EuRoC datasets, which provide images, IMU measurements and ground truth data [BNG16]. Three SLAM backend algorithms use the output from the common frontend of monocular camera vision, which includes the feature measurements and the estimated trajectory from visual-inertial odometry (VIO). The data association of the feature measurements is also provided by the frontend. Therefore, the SLAM algorithms can close the loop when the same landmarks are observed at both ends of the loop. The VIO trajectory is provided by the open-source okvis project [LLB15]. As for the IMU measurements that are obtained at a higher rate, we use the preintegration method in [FCD17] to combine IMU measurements between two keyframes.

In EM-based SLAM algorithms, the forward Kalman filtering in the E-step only uses IMU odometry data, which gives a very inaccurate result over a longer period of time. In a visual-inertial system, the visual-inertial odometry (VIO) normally gives a better initially estimated trajectory. Therefore, we take the weighted average between the IMU propagated

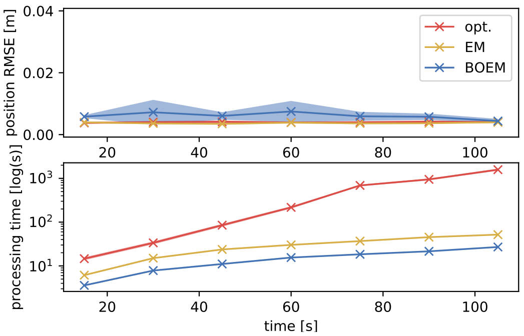


Figure 5.3: The trajectory estimation accuracy and the processing time with increasing time intervals. The shaded areas show 1 standard deviation error bar over 20 trials. As an online algorithm, the BOEM SLAM can effectively discard information, and thus it has a lower processing time.

states and the VIO states in the forward Kalman filtering. As for optimization SLAM, the VIO trajectory is used to initialize the optimization process.

The MH 04 dataset is demonstrated as an example to show the estimation trajectories in Fig. 5.4 and the trajectory errors in Fig. 5.5. Overall, the SLAM algorithms provide more accurate estimated trajectories than the VIO result, since they can improve the estimation with loop closure. We can also see the error drop in the last 20 sec of BOEM SLAM due to the loop closure, which shows the effectiveness of the BOEM SLAM.

We then extend to all EuRoC datasets, and the trajectory estimation errors are presented in Fig. 5.6. While all SLAM algorithms bring down the VIO trajectory error, the optimization SLAM has the most consistent performance, but overall the performances of all SLAM algorithms are comparable. We attribute the prominent performance of optimization SLAM to two possible explanations: First, this algorithm can incorporate robust cost functions easily to remove outliers of visual measurement. Even though we implement the similar mechanism in the EM-based algorithms, we believe that there is still room for improvement with further investigation. Second, since the EM-based algorithms perform filtering

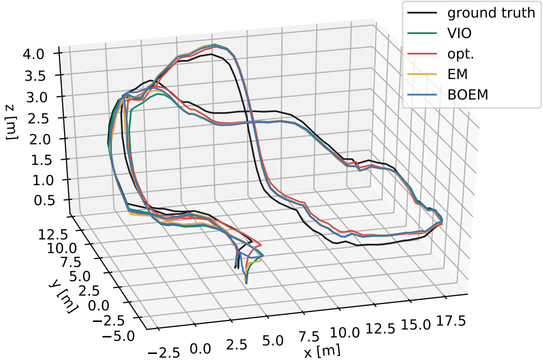


Figure 5.4: The estimated trajectories of SLAM algorithms on dataset MH 04. The visual-inertial odometry (VIO) is obtained from the okvis project.

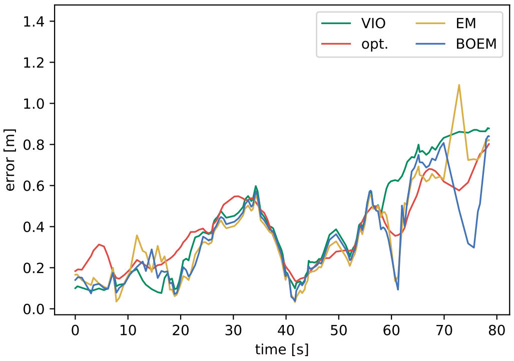


Figure 5.5: The trajectory errors of SLAM algorithms on dataset MH 04.

and smoothing in the E-step, they are more susceptible to linearization issues. The performance difference between the EM SLAM and the BOEM SLAM is not significant, which demonstrates that the block online structure does not sacrifice the estimation accuracy.

We now turn to the processing time across all datasets as shown in Fig. 5.7. Throughout all datasets, the BOEM SLAM uses less than 40 sec consistently to reach comparable SLAM performance. In general, the processing time increases with the duration of the trajectory, and the processing times of EM SLAM and BOEM SLAM reflect this trend. For optimization SLAM, the small processing times in datasets MH 02 and MH 03 in particular come from

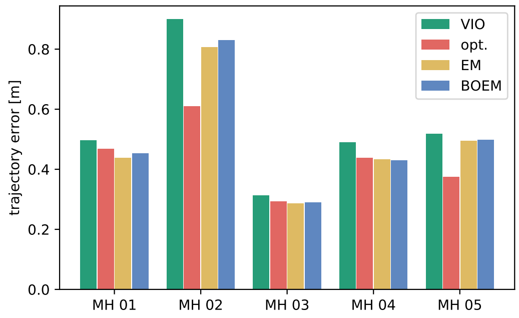


Figure 5.6: The trajectory estimation error of SLAM algorithms on EuRoC datasets.

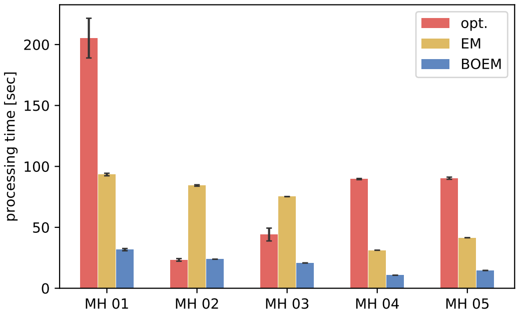


Figure 5.7: The processing time of SLAM algorithms on EuRoC datasets. Each error bar shows the standard deviation over 10 trials. The BOEM SLAM algorithm shows a significantly lower processing time compared to the other two algorithms.

the fact that the initial estimated trajectory leads to convergence early. However, for other datasets, the optimization SLAM takes the longest processing time, which is about 5 times of our BOEM SLAM.

# CHAPTER 6

# Conclusions

This dissertation demonstrates two important spatial autonomy scenarios. In the first multirobot scenario, I proposed a new cooperative localization algorithm that uses less communication and is more robust to communication failures. Second, an online EM SLAM algorithm is designed and implemented for visual-inertial systems that can discard past information efficiently.

In this section, I will discuss further topics in robotics with the foundations built in previous chapters. First, I will propose an algorithm for multirobot mapping. With the experience in multirobot localization, the proposed algorithm highlights the robustness nature in the multirobot systems. I will also discuss the exciting topics revealed from my research. I will talk about the principle of handling data in robotic scenarios: sharing data and forgetting data. Even though the concept is straightforward, the questions for researchers and engineers lie in how to find a good algorithm.

## 6.1 Multirobot Mapping

Multirobot mapping considers an extension of single-robot SLAM problem to a multirobot scenario. Several robots do not have any pre-existing knowledge to an environment, and they want to construct a map of this environment cooperatively. The coverage of multiple robots is larger than that of a single robot, which suggests that the multirobot mapping scenarios will be more efficient. However, the information exchange among robots pose engineering

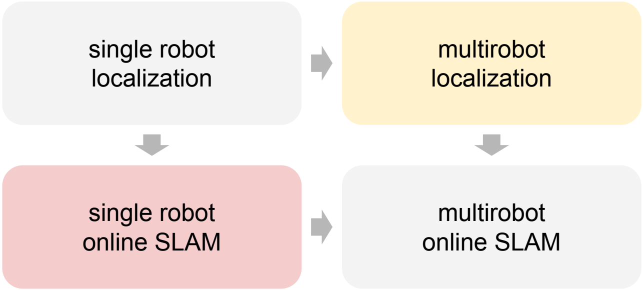


Figure 6.1: The roadmap of proposing the multirobot mapping algorithm. The proposed algorithm maintains robustness from the multirobot localization algorithm, and it uses the EM framework as for the online EM algorithm.

challenges to realize this system.

Current methods typically consist of two individual steps: first, each robot builds its own map, and then all individual maps are combined together. One major technical challenge is to determine the transformation between individual maps before merging them. Even though this two-step framework can construct a map with cooperative robots, its simplicity limits its potential in reality. First, while the entire map is constructed in the end, it prevents robots to change its behavior according to the map. For example, robots are not able to determine the unvisited territory on the fly. Second, the fusion is done on a centralized robot that collects all the map, which poses a vulnerability issue is a distributed system.

This dissertation suggests a novel way to design a multirobot mapping algorithm. The EM-based SLAM algorithm for a single robot provides a framework that separates localization and mapping, where the mapping is viewed as an optimization problem. For multirobot system, we can extend this EM framework, and uses distributed optimization algorithm for the cooperative mapping problem. We follow the notation in the SLAM section. In particular, the E-step takes place in each robot, and robot $n$ computes the conditional expectation as

$$Q_k^n(\lambda) = \frac{1}{T}\,\mathsf{E}_{\hat{\lambda}_{k-1}}\left[\sum_{t=1}^{T} \log p(\mathbf{s}_t^n, \mathbf{o}_t^n | \mathbf{s}_{t-1}^n; \lambda) \middle| \mathbf{o}_{1:T}^n\right].$$

As the E-step is interpreted as the localization step, each robot reevaluates its own trajectory based on the current landmark estimate $\hat{\lambda}_{k-1}$. In M-step, all robots collectively solves

$$\hat{\lambda}_k = \arg\max_{\lambda} \sum_{n=1}^{N} Q_k^n(\lambda).$$

This is a distributed optimization problem, and robots can use distributed optimization algorithms to solve this problem, EXTRA for example. For this algorithm, robots only need to communicate with its neighbors, which preserves the distributed nature of multirobot systems.

In the traditional framework, the transformation of maps in different robot is unknown, and the centralized unit has to spend a lot of effort to recover this. In other words, the traditional framework solves two problems at once: finding the transformation and combining the map. The proposed framework instead requires that robots solve the transformation configuration first, and then solve the map fusion on the fly. The transformation configuration can be solved at the beginning of the robot deployment, especially when robots are deployed nearby. By separating these two main problems, the overall multirobot mapping framework can be more efficient and more robust.

There are 3 advantages of this new framework.

- The traditional framework only combines the map in the end of the entire process. In this case, the updated map cannot be utilized for robots to further explore the space. On the contrary, in the proposed framework, the map is constructed along the process rather than in the end. Therefore, robots can use the updated map to further explore the unmapped territory.

- The proposed framework does not rely on a centralized unit. Therefore, the proposed framework has better robustness compared to the original centralized framework. In particular, the main distributed component in the proposed framework is the distributed optimization problem, while the rest of the computation remains local.

- In the proposed framework, all robots has the updated map from the algorithm, and no extra step is required to broadcast the updated map. On the contrary, the traditional framework computed the entire map in a centralized unit, and has to broadcast the map in the end.

Even though I do not directly proposed the online multirobot SLAM algorithm, the proposed multirobot mapping algorithm can be easily extended. In the multirobot mapping, each robot localizes itself on its own. We can further impose the multirobot localization mapping algorithm in this dissertation to realize a fully multirobot SLAM algorithm, thanks to the EM framework that separates the localization and the mapping. On the other hand, the proposed multirobot mapping algorithm can also be extended to the online implementation. Once we recognize that multirobot mapping is still an inference problem over a hidden Markov model, the online EM algorithm can then be applied. However, the interesting question here is the general distributed and online EM algorithm. Once we can formalize it, this algorithm can be applied beyond robotics, since the hidden Markov model is very common in science and engineering.

I am very excited to see this framework to be implemented in visual-inertial systems. More challenges and possibilities will show up if we consider heterogeneous system. For example, a system with ground robots and aerial robots.

## 6.2 Sharing Information

As the manufacture of robots becomes more democratized and accessible, we are expect to see more robots in daily lives. The challenge does not only lie in the manufacturing robots, but also the algorithm design in this new scenario. While each robot has different sensors and actuators, robots can cooperate to jointly achieve their own goals. The multirobot localization project in this dissertation provides the framework for general information sharing, with a focus on the inter-robot correlation. When we move on to the general cooperative scenario in multirobot systems, the information sharing has new challenges.

The first challenge is how one robot should use the shared information. As shown in the localization case, massive communication is required to obtain the inter-robot correlation, without which the obtained information is not very useful. Most of the time, robots will combine the information from other robots, and the theory is provided in this dissertation from a barycenter perspective. However, the information shared in the multirobot system is not limited to this case. As for general information sharing, an easy and efficient information usage is essential for every case.

Another challenge is that the algorithm for multirobot system should be built on its distributed nature. While some algorithms have the centralized-equivalent performance, the price to pay is not only the massive information requirement, but also the loss of robustness. In terms of information usage, the information may not be considered as exact or fully characterized in a distributed system. Therefore, different information usage schemes should be investigated for distributed systems to ensure its robustness.

The challenge of realizing multirobot system is mostly algorithmic. If we can not design the algorithm coordinating how robots interact, the so-called multirobot systems will be just several single robot system without any further potential. As the number of robots keep increasing, we would also expect something like “culture” and “social norm” that shows up in our own society. In that case, we not only have to design the interaction between robots, but we also have to keep an eye on the global behavior emerging from these local interactions.

## 6.3 Forgetting Information

When robots are deployed in realistic environment, data from sensors will keep arriving, since we are living in a dynamic world. A robot not only has to know how to process the data, but summarizing and discarding data is also important to avoid information overload. The online SLAM project shows the effort to take this challenge into consideration. This question is related to both the algorithm and the representation of the spatial autonomy that we mentioned in the Introduction. While robots are able to “forget” unimportant information and keep essential information in an efficient representation, robots can operate in the dynamic environment for a long period of time.

As robots use less communication and computation to acquire spatial autonomy, robots have more resource for all high-level tasks, including planning, exploration, navigation, or even interaction with human. Thus, this dissertation serves as a block to the road for a future with abundant robots. With more understanding spatial autonomy by construction, we might be able to decipher the mystery of the human spatial autonomy one day.

## 6.4 The Representation of Spatial Autonomy

While this dissertation is about obtaining spatial autonomy, the spatial autonomy is usually not the end goal of a robot. To put this concretely, we never talk about how robots move or how robots determine the trajectories. How does the study of spatial autonomy inform the upper layer decision making?

There are at least two ways for decision making. If we obtain the control inputs directly, this can be considered a planning problem. If we obtain the policy, or a mapping from the state to the action, it should be considered a reinforcement learning problem. In both way, the state is often considered exact while no uncertainty involved. More over, when the state space is large or even continuous, the representation of the state is not trivial but has to be given or learned. This problem refers back to the three-stage overview of the spatial autonomy. However, instead of viewing from the reinforcement learning, we provide a broader interpretation of the representation as the bridge between lower level spatial autonomy and the upper layer decision making.

The decision making should integrate the uncertainty smoothly. The first simple scheme is to let the robot randomly walk around the environment. Once the location and the map are accurate enough, the decision making based on exact information can take over. However, a more complete scheme can be designed that take the uncertainty in the spatial autonomy into account. This is the way the build the entire autonomy of robots from the bottom, and with more research on this, we can expect the real autonomy in robots to be realized on day.

# REFERENCES

[AMO] S. Agarwal, K. Mierle, and Others. "Ceres Solver." http://ceres-solver.org.

[ARC09] M. Azmani, S. Reboul, J.-B. Choquel, and M. Benjelloun. "A recursive fusion filter for angular data." In *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 882–887, December 2009.

[ARM01] P. O. Arambel, C. Rago, and R. K. Mehra. "Covariance intersection algorithm for distributed spacecraft state estimation." In *Proceedings of the American Control Conference*, volume 6, pp. 4398–4403, Arlington, VA, USA, 2001.

[BAY17] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser. "Simultaneous localization and mapping: A survey of current trends in autonomous driving." *IEEE Transactions on Intelligent Vehicles*, **2**(3):194–220, 2017.

[BC14] G. Battistelli and L. Chisci. "Kullback-Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability." *Automatica*, **50**(3):707–718, March 2014.

[BDG05] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra. "Clustering on the unit hypersphere using von Mises-Fisher distributions." *Journal of Machine Learning Research*, **6**(46):1345–1382, 2005.

[BJA12] T. Bailey, S. Julier, and G. Agamennoni. "On conservative fusion of information with unknown non-Gaussian dependence." In *Proceedings of the International Conference on Information Fusion (FUSION)*, pp. 1876–1883, July 2012.

[BMG13] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu. "Discrete extended Kalman filter on Lie groups." In *Proceedings of the European Signal Processing Conference*, pp. 1–5, 2013.

[BNG06] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. "Consistency of the EKF-SLAM algorithm." In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3562–3568, Beijing, China, October 2006.

[BNG16] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. "The EuRoC micro aerial vehicle datasets." *International Journal of Robotics Research*, **35**(10):1157–1163, 2016.

[BWL09] A. Bahr, M. R. Walter, and J. J. Leonard. "Consistent cooperative localization." In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3415–3422, Kobe, Japan, May 2009.

[CAM02] L. Chen, P. O. Arambel, and R. K. Mehra. “Estimation under Unknown Correlation: Covariance Intersection Revisited.” *IEEE Transactions on Automatic Control*, **47**(11):1879–1882, November 2002.

[CCC16] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age.” *IEEE Transactions on Robotics*, **32**(6):1309–1332, 2016.

[CCM20] T.-K. Chang, S. Chen, and A. Mehta. “Multirobot cooperative localization algorithm with explicit communication and its topology analysis.” In N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, editors, *Robotics Research*, Springer Proceedings in Advanced Robotics, pp. 643–659, Cham, 2020. Springer International Publishing.

[CCM22] T.-K. Chang, K. Chen, and A. Mehta. “Resilient and consistent multirobot cooperative localization with covariance intersection.” *IEEE Transactions on Robotics*, **38**(1):197–208, February 2022.

[CLK19] G. Chen, Y. Lu, J. A. King, F. Cacucci, and N. Burgess. “Differential influences of environment and self-motion on place and grid cell firing.” *Nature Communications*, **10**(1):630, February 2019.

[CM18a] T.-K. Chang and A. Mehta. “Control-theoretical and topological analysis of covariance intersection based distributed Kalman filter.” *IEEE Control Systems Letters*, **2**(4):665–670, October 2018.

[CM18b] T.-K. Chang and A. Mehta. “Optimal scheduling for resource-constrained multirobot cooperative localization.” *IEEE Robotics and Automation Letters*, **3**(3):1552–1559, July 2018.

[CNG13] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis. “Decentralized multi-robot cooperative localization using covariance intersection.” In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1412–1417, Tokyo, Japan, November 2013.

[CYG20] S. Claici, M. Yurochkin, S. Ghosh, and J. Solomon. “Model fusion with Kullback-Leibler divergence.” In *Proceedings of the International Conference on Machine Learning*, pp. 2038–2047, November 2020.

[DFC18] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak. “Hyperspherical variational auto-encoders.” In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 856–865, January 2018.

[DK06] F. Dellaert and M. Kaess. “Square Root SAM: Simultaneous localization and mapping via square root information smoothing.” *International Journal of Robotics Research*, **25**(12):1181–1203, 2006.

[DL13] T. Dumont and S. Le Corff. “Online EM for indoor simultaneous localization and mapping.” In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6431–6435, May 2013. ISSN: 2379-190X.

[DL14] T. Dumont and S. Le Corff. “Simultaneous localization and mapping in wireless sensor networks.” *Signal Processing*, **101**:192–203, August 2014.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum likelihood from incomplete data via the EM algorithm.” *Journal of the Royal Statistical Society, Series B*, **39**(1):1–38, 1977.

[DNC01] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. “A solution to the simultaneous localization and map building (SLAM) problem.” *IEEE Transactions on Robotics and Automation*, **17**(3):229–241, June 2001.

[ESL06] R. M. Eustice, H. Singh, and J. J. Leonard. “Exactly sparse delayed-state filters for view-based SLAM.” *IEEE Transactions on Robotics*, **22**(6):1100–1114, 2006.

[FBK00] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. “A probabilistic approach to collaborative multi-robot localization.” *Autonomous Robots*, **8**(3):325–344, June 2000.

[FCD17] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. “On-manifold preintegration for real-time visual-inertial odometry.” *IEEE Transactions on Robotics*, **33**(1):1–21, 2017.

[FCM20] C. Fan, T.-K. Chang, and A. Mehta. “Kullback-Leibler average of von Mises distributions in multi-agent systems.” In *Proceedings of the IEEE Conference on Decision and Control*, pp. 2411–2417, December 2020. ISSN: 2576-2370.

[FI01] G. Freiling and V. Ionescu. “Monotonicity and convexity properties of matrix Riccati equations.” *IMA Journal of Mathematical Control and Information*, **18**(1):61–72, March 2001.

[GBC20] L. Gao, G. Battistelli, and L. Chisci. “Multiobject fusion with minimum information loss.” *IEEE Signal Processing Letters*, **27**:201–205, 2020.

[GK01] R. Grabowski and P. Khosla. “Localization techniques for a team of small robots.” In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pp. 1067–1072, October 2001.

[GKS10] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. “A tutorial on graph-based SLAM.” *IEEE Intelligent Transportation Systems Magazine*, **2**(4):31–43, 2010.

[HBM17] Md Abul Hasnat, J. Bohn, J. Milgram, S. Gentric, and L. Chen. “von Mises-Fisher mixture model-based deep learning: Application to face verification.” *arXiv:1706.04264 [cs]*, December 2017. arXiv: 1706.04264.

[HD07] S. Huang and G. Dissanayake. “Convergence and consistency analysis for extended Kalman filter based SLAM.” *IEEE Transactions on Robotics*, **23**(5):1036–1049, October 2007.

[HFM05] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser. “Microstructure of a spatial map in the entorhinal cortex.” *Nature*, **436**(7052):801–806, August 2005.

[HMS03a] A. Howard, M. J. Mataric, and G. S. Sukhatme. “Localization for mobile robot teams: A distributed MLE approach.” In *Experimental Robotics VIII*, pp. 146–155, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[HMS03b] A. Howard, M. J. Mataric, and G. S. Sukhatme. “Putting the ‘I’ in ‘team’: An ego-centric approach to cooperative localization.” In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 868–874, Taipei, Taiwan, September 2003.

[HTM11] G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis. “Observability-based consistent EKF estimators for multi-robot cooperative localization.” *Autonomous Robots*, **30**(1):99–122, January 2011.

[IGR12] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. “Graph-based distributed cooperative navigation for a general multi-robot measurement model.” *International Journal of Robotics Research*, May 2012.

[JU97] S. J. Julier and J. K. Uhlmann. “A Non-Divergent Estimation Algorithm in the Presence of Unknown Correlations.” In *Proceedings of the American Control Conference*, volume 4, pp. 2369–2373, Albuquerque, NM, USA, June 1997.

[JU01] S. J. Julier and J. K. Uhlmann. “A counter example to the theory of simultaneous localization and map building.” In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pp. 4238–4243, Seoul, South Korea, May 2001.

[KAC19] J. Klingner, N. Ahmed, and N. Correll. “Fault-tolerant covariance intersection for localizing robot swarms.” In *Distributed Autonomous Robotic Systems*, pp. 485–497. Springer Int. Publishing, 2019.

[KCA06] N. Karam, F. Chausse, R. Aufrère, and R. Chapuis. "Localization of a group of communicating vehicles by state exchange." In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 519–524, Beijing, China, October 2006.

[KGS11] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. "g$^2$o: A general framework for graph optimization." In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3607–3613, 2011.

[KH00] R. Kurazume and S. Hirose. "An experimental study of a cooperative positioning system." *Autonomous Robots*, **8**(1):43–52, January 2000.

[KHG18] S. S. Kia, J. Hechtbauer, D. Gogokhiya, and S. Martínez. "Server-Assisted Distributed Cooperative Localization Over Unreliable Communication Links." *IEEE Transactions on Robotics*, **34**(5):1392–1399, October 2018.

[KHS17] M. Kok, J. D. Hol, and T. B. Schön. "Using inertial sensors for position and orientation estimation." *Foundations and Trends in Signal Processing*, **11**(1-2):1–153, 2017.

[KJR11] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering." In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3281–3288, 2011.

[KNH94] R. Kurazume, S. Nagata, and S. Hirose. "Cooperative positioning with multiple robots." In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1250–1257, May 1994.

[KRD08] M. Kaess, A. Ranganathan, and F. Dellaert. "iSAM: Incremental smoothing and mapping." *IEEE Transactions on Robotics*, **24**(6):1365–1378, 2008.

[KSH00] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation.* Pearson, 2000.

[LCM18] K. Lenac, J. Ćesić, I. Marković, and I. Petrović. "Exactly sparse delayed state filter on Lie groups for long-term pose graph SLAM." *International Journal of Robotics Research*, **37**(6):585–610, 2018.

[LF13] S. Le Corff and G. Fort. "Online expectation maximization based algorithms for inference in hidden Markov models." *Electronic Journal of Statistics*, **7**:763–792, 2013.

[LFM12] S. Le Corff, G. Fort, and E. Moulines. "New online EM algorithms for general hidden Markov models. Application to the SLAM problem." In F. Theis, A. Cichocki, A. Yeredor, and M. Zibulevsky, editors, *Latent Variable Analysis and Signal Separation*, Lecture Notes in Computer Science, pp. 131–138, Berlin, Heidelberg, 2012. Springer.

[LHB11] K. Leung, Y. Halpern, T. D. Barfoot, and H. Liu. "The UTIAS multi-robot cooperative localization and mapping dataset." *International Journal of Robotics Research*, **30**(8):969–974, 2011.

[LLB15] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. "Keyframe-based visual-inertial odometry using nonlinear optimization." *International Journal of Robotics Research*, **34**(3):314–334, 2015.

[LM97] F. Lu and E. Milios. "Globally consistent range scan alignment for environment mapping." *Autonomous Robots*, **4**(4):333–349, 1997.

[LNY13] H. Li, F. Nashashibi, and M. Yang. "Split covariance intersection filter: Theory and its application to vehicle localization." *IEEE Transactions on Intelligent Transportation Systems*, **14**(4):1860–1871, December 2013.

[LSR18] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard. "Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication." *International Journal of Robotics Research*, **37**(10):1152–1167, 2018.

[MMT15] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. "ORB-SLAM: A versatile and accurate monocular SLAM system." *IEEE Transactions on Robotics*, **31**(5):1147–1163, 2015.

[MR06] A. I. Mourikis and S. I. Roumeliotis. "Performance analysis of multirobot cooperative localization." *IEEE Transactions on Robotics*, **22**(4):666–681, August 2006.

[MSK03] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer-Verlag, 2003.

[MT17a] R. Mur-Artal and J. D. Tardós. "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras." *IEEE Transactions on Robotics*, **33**(5):1255–1262, 2017.

[MT17b] R. Mur-Artal and J. D. Tardós. "Visual-inertial monocular SLAM with map reuse." *IEEE Robotics and Automation Letters*, **2**(2):796–803, 2017.

[NRM09] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli. "Distributed maximum a posteriori estimation for multi-robot cooperative localization." In *Proceeding of the IEEE International Conference on Robotics and Automation*, pp. 1402–1409, Kobe, Japan, May 2009.

[OD71] J. O'Keefe and J. Dostrovsky. "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat." *Brain Research*, **34**(1):171–175, November 1971.

[PBM12] A. Prorok, A. Bahr, and A. Martinoli. “Low-cost collaborative localization for large-scale multi-robot systems.” In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4236–4241, Saint Paul, MN, USA, May 2012.

[QLS18] T. Qin, P. Li, and S. Shen. “VINS-Mono: A Robust and Versatile monocular visual-inertial state estimator.” *IEEE Transactions on Robotics*, **34**(4):1004–1020, 2018.

[RB02] S. I. Roumeliotis and G. A. Bekey. “Distributed multirobot localization.” *IEEE Transactions on Robotics and Automation*, **18**(5):781–795, October 2002.

[RDM98] I. M. Rekleitis, G. Dudek, and E. E. Milios. “On multiagent exploration.” In *in Proc. Vision Interface*, pp. 455–461, 1998.

[RKW13] P. Ravassard, A. Kees, B. Willers, D. Ho, D. A. Aharoni, J. Cushman, Z. M. Aghajan, and M. R. Mehta. “Multisensory control of hippocampal spatiotemporal selectivity.” *Science*, **340**(6138):1342–1346, June 2013.

[RNA15] M. Reinhardt, B. Noack, P. O. Arambel, and U. D. Hanebeck. “Minimum Covariance Bounds for the Fusion under Unknown Correlations.” *IEEE Signal Processing Letters*, **22**(9):1210–1214, September 2015.

[RTS65] H. E. Rauch, F. Tung, and C. T. Striebel. “Maximum likelihood estimates of linear dynamic systems.” *AIAA Journal*, **3**(8):1445–1450, 1965.

[SDA20] J. Solà, J. Deray, and D. Atchuthan. “A micro Lie theory for state estimation in robotics.” *arXiv:1812.01537 [cs]*, 2020.

[SMG88] M. Salgado, R. Middleton, and G. C. Goodwin. “Connection between continuous and discrete Riccati equations with applications to Kalman filtering.” *IEE Proceedings D - Control Theory and Applications*, **135**(1):28–34, January 1988.

[SRA14] G. Stienne, S. Reboul, M. Azmani, J. B. Choquel, and M. Benjelloun. “A multitemporal multi-sensor circular fusion filter.” *Information Fusion*, **18**:86–100, July 2014.

[SRD21] M. Servières, V. Renaudin, A. Dupuis, and N. Antigny. “Visual and visual-inertial SLAM: State of the art, classification, and experimental benchmarking.” *Journal of Sensors*, **2021**:e2054828, February 2021.

[SSG17] Z. Sjanic, M. A. Skoglund, and F. Gustafsson. “EM-SLAM with inertial/visual applications.” *IEEE Transactions on Aerospace and Electronic Systems*, **53**(1):273–285, 2017.

[Sun04] S.-l. Sun. "Multi-Sensor Optimal Information Fusion Kalman Filters with Applications." *Aerospace Science and Technology*, **8**(1):57–62, January 2004.

[TM06] S. Thrun and M. Montemerlo. "The Graph SLAM algorithm with applications to large-scale mapping of urban structures." *International Journal of Robotics Research*, **25**(5-6):403–429, 2006.

[TZZ10] N. Trawny, X. S. Zhou, K. Zhou, and S. I. Roumeliotis. "Interrobot transformations in 3-D." *IEEE Transactions on Robotics*, **26**(2):226–243, April 2010.

[YBD18] G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield, B. J. Nelson, B. Scassellati, M. Taddeo, R. Taylor, M. Veloso, Z. L. Wang, and R. Wood. "The grand challenges of Science Robotics." *Science Robotics*, January 2018.

[Zei95] E. Zeidler. *Applied Functional Analysis: Applications to Mathematical.* Springer-Verlag New York, 1995.

[ZR13] X. S. Zhou and S. I. Roumeliotis. "Determining 3-D Relative Transformations for Any Combination of Range and Bearing Measurements." *IEEE Transactions on Robotics*, **29**(2):458–474, April 2013.

[ZR19] P. Zhu and W. Ren. "Multi-robot Joint Localization and Target Tracking with Local Sensing and Communication." In *Proceedings of the American Control Conference*, pp. 3261–3266, Philadelphia, PA, USA, July 2019.