Range-aided Relative Pose Estimation and Formation Planning for Multi-robot Systems

Syed Shabbir Ahmed

Department of Mechanical Engineering
McGill University, Montreal

May 2024

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Masters of Science

Abstract

This thesis addresses the challenges associated with the estimation of the relative position and attitude between robots, collectively the *relative pose*, using ultra-wideband (UWB) radio. An accurate and precise relative pose estimate is essential for the coordination of robots in a variety of applications, such as search and rescue missions, environmental monitoring, surveillance, and collision avoidance. UWB transceivers, which are cheap and lightweight devices, offer a means to obtain distance or *range* measurements between mobile robots and fixed anchors in known locations. With fixed anchors, the localization of robots is generally precise. In infrastructure-free, and unexplored environments, with no fixed anchors, robots equipped with UWB transceivers can estimate their relative poses by exchanging range measurements. However, in the infrastructure-free scenario, there are challenges associated with the limited observability of the relative robot position and attitude states, which makes the localization problem, as well as any subsequent path planning, difficult.

The thesis meets two core objectives in using range-aided systems for multi-robot coordination in infrastructure-free environments. Firstly, this thesis addresses the problem of relative pose estimation of a multi-robot system with limited observability. This is done by using a Gaussian-sum filter (GSF) to estimate the relative pose between robots. The GSF is designed to exploit the ambiguous states that arise from the limited observability properties of the system in order to provide a consistent and accurate estimation of the relative pose states. Secondly, this thesis addresses the problem of path planning for multi-robot systems in formation with limited observability. This is done by minimizing a cost function that balances the observability of the relative pose states and any user-defined formation configuration. The thesis presents simulation and experimental results to demonstrate the effectiveness of the proposed approaches for the relative pose estimation and formation planning problems.

Résumé

Cette thèse explore les défis associés à l'estimation de la position et de l'orientation relatives entre des robots, c'est-à-dire leur état relatif, en utilisant la radio à bande ultra-large (UWB). Une estimation précise de leur position et orientation relative est essentielle pour la coordination des robots dans une variété d'applications, telles que les missions de recherche et de sauvetage, l'exploration de l'environnement, ainsi que la surveillance et l'évitement des collisions. Les capteurs UWB, qui sont bons marchés et légers, permettent d'obtenir des mesures de distance entre des robots mobiles et des ancrages fixes situés à des endroits connus. Avec des ancrages fixes, la localisation des robots est généralement précise. Dans les environnements inexplorés et sans infrastructure, sans ancrage fixe, les robots équipés d'émetteurs-récepteurs UWB peuvent estimer leur position relative en échangeant des mesures de distance. Cependant, dans le système sans infrastructure, des défis associés à l'observabilité limitée de la position et de l'orientation relative du robot apparaissent, ce qui rend difficile le problème de localisation et, par la suite, la planification de trajectoire.

Cette thèse répond à deux objectifs principaux de l'utilisation de systèmes employant des capteurs UWB pour la coordination multi-robot dans des environnements sans infrastructure. Premièrement, elle explore le problème de l'estimation de la position relative d'un système multi-robot avec une observabilité limitée. Un filtre à somme Gaussienne (FSG) est utilisé pour estimer la position relative entre les robots. Le FSG est conçu pour exploiter les états ambigus qui découlent des propriétés d'observabilité limitées du système afin de fournir une estimation cohérente et précise des états relatifs. Deuxièmement, cette thèse aborde le problème de la planification des trajectoires de formation pour les systèmes multi-robots à observabilité limitée. Cela se fait en minimisant une fonction qui équilibre l'observabilité des états relatifs et toute configuration de formation définie par l'utilisateur. Cette thèse présente des résultats de simulation et d'expérimentation afin de démontrer l'efficacité des approches proposées pour résoudre des problèmes d'estimation d'états relative et de planification de formations.

Acknowledgements

This thesis required the unwavering support of many individuals, and I would like to take this opportunity to express my gratitude to them. Firstly, I am immensely grateful to my supervisor, James R. Forbes, for his guidance over the past 2 years. He is an outstanding supervisor who has made my masters journey the most insightful and enjoyable. Notably, his contagious passion and enthusiasm for research have been a constant source of inspiration for me and a strong driving force behind my work.

Secondly, special thanks to Mohammed A. Shalaby, and Charles C. Cossette who have been instrumental in my research journey. Their constant support and guidance have been invaluable, and they have made my masters journey a truly memorable one. The valuable technical discussions and insights they have provided have been instrumental in shaping my research direction. Alongside, every member of the DECAR team has helped me in numerous ways, and I am grateful for their support.

I am immensely grateful to my mother and father for their unconditional support and endless sacrifices in making my journey in McGill University a reality. Their pride in my work, and constant eagerness to learn about my projects, has always pushed me to do better.

My friends and colleagues are the pillars who have always lent me their support and encouragement, and made my time at McGill absolutely unforgettable.

Finally, special credit goes to the NSERC Alliance Grant program, the NSERC Discovery Grant program, and the Canadian Foundation for Innovation (CFI) program for supporting the work in this thesis.

Table of Contents

List of Figures

Figure

List of Tables

Table

List of Abbreviations

EKF extended Kalman filter

GSF Gaussian-sum filter

IMU inertial measurement unit

NEES normalized estimation error squared

NIS normalized innovation squared

PF particle filter

RMSE root mean squared error

SLAM simultaneous localization and mapping

UAV unmanned aerial vehicle

UWB ultra-wideband

List of Symbols

$\mathbb{R}^n$	The vector space of real n-dimensional vectors
$\mathbb{R}^{m\times n}$	The vector space of real $m \times n$-dimensional matrices
$\lVert \cdot \rVert$	The Euclidean norm of a physical vector
$\mathbf{0}$	Zero matrix
$\mathbf{1}$	Identity matrix
$\operatorname{diag}(\cdot)$	Produces a block-diagonal matrix, such that $\operatorname{diag}(\mathbf{A}_1, \dots, \mathbf{A}_n)$ contains $\mathbf{A}_1, \dots, \mathbf{A}_n$ on the main diagonal and $\mathbf{0}$ elsewhere
$(\cdot)^{\mathsf{T}}$	Transpose
$(\cdot)^{-1}$	Inverse
$(\cdot)^{\times}$	Skew-symmetric cross operator
$\check{(\cdot)}$	A prior estimate
$\hat{(\cdot)}$	A posterior estimate
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian probability density with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\underrightarrow{r}^{zw}$	The physical vector expressing the position of point z relative to point w
$\mathcal{F}_a$	Reference frame a
$\mathbf{C}_{ab}$	A DCM parameterizing the attitude of $\mathcal{F}_a$ relative to $\mathcal{F}_b$.
$\mathbf{r}_a^{zw}$	The physical vector $\underrightarrow{r}^{zw}$ resolved in reference frame $\mathcal{F}_a$.
G	A matrix Lie group
$\mathfrak{g}$	A matrix Lie algebra, $\mathfrak{g} = T_{\mathbf{1}}G$
$\exp(\cdot)$	The exponential map for matrix Lie groups, $\exp(\cdot) : \mathfrak{g} \to G$
$\log(\cdot)$	The logarithmic map for matrix Lie groups, $\log(\cdot) : G \to \mathfrak{g}$
$(\cdot)^{\vee}$	The contraction operator for matrix Lie groups, $(\cdot)^{\vee} : \mathfrak{g} \to \mathbb{R}^d$
$(\cdot)^{\wedge}$	The expansion operator for matrix Lie groups, $(\cdot)^{\wedge} : \mathbb{R}^d \to \mathfrak{g}$
$p(\mathbf{a})$	A probability density function of $\mathbf{a}$
$p(\mathbf{a} \vert \mathbf{b})$	The probability density function of $\mathbf{a}$ conditioned on $\mathbf{b}$

Preface

The contribution of this thesis that, to the author's knowledge, are original are as follows:

- Chapter 4
  - The development of a Gaussian-sum filter for range-based 3D relative pose estimation in the presence of ambiguities.
- Chapter 5
  - The development of a cost function for optimal robot formations that balances range-based observability and user-defined configurations.

Another contribution of this thesis is the evaluation of all the proposed algorithms in simulation and experimental scenarios. Hours of experimental flights were conducted to validate the proposed algorithms.

Special credit is given to Mohammed A. Shalaby and Charles C. Cossette for their part in making the experiments in this thesis a reality and for the insightful discussions that led to many of the ideas presented in this thesis. Nonetheless, the contributions claimed in this thesis are solely developed by the author of this thesis.

The topic addressed in this thesis has a wide range of potential applications, and the author hopes it represents a step towards making cheap, lightweight, and reliable multi-robot systems a reality, with the sole purpose of making the world a better place.

All texts, figures, and tables in this thesis are original, except where otherwise indicated.

Chapter 1

Introduction

The goal of the thesis is to develop estimation and planning algorithms that ensure the range-based relative position and attitude estimation, referred to as relative pose estimation, between a team of robots is accurate and precise. A popular choice for obtaining range measurements is the use of ultra-wideband (UWB) radio transceivers, which are also referred to as UWB *tags*. UWB tags are cheap, lightweight, and low-power sensor that are capable of providing 10 cm accurate range measurements between a pair of transceivers at a high frequency. UWB sensors are highly suitable for small robots with limited computational power and payload capacity. UWB tags are also useful for localization in *Global Position System* (GPS)-denied environments, where GPS signals are either unavailable or unreliable. A high localization accuracy is vital for applications such as autonomous surveillance, coverage and exploration, infrastructure and mine inspection, collaborative simultaneous localization and mapping (SLAM), and formation control. UWB-aided robotic systems have only recently been studied in academic research [1–4], and still requires further research to be able to provide accurate localization in real-world scenarios.

UWB tags are oftentimes static anchors with known locations and are then used to localize tags placed on mobile robots [5–10]. This research is a step forward towards improving localization of robots in infrastructure-free setups, where the UWB tags are placed on the mobile robots themselves. This would result in a fully self-contained localization system that is not dependent on external infrastructure. However, in such systems the estimation task is more difficult due to no information being available about the absolute position of the robots, from fixed anchors or GPS.

A highly challenging issue in UWB-aided systems is the presence of ambiguous poses, which result from observability limitations in the range-based relative pose estimation. For instance, in a 3D scenario, the true pose of the robots is not uniquely determined by the range measurements. This is because the range measurements only provide the distance between

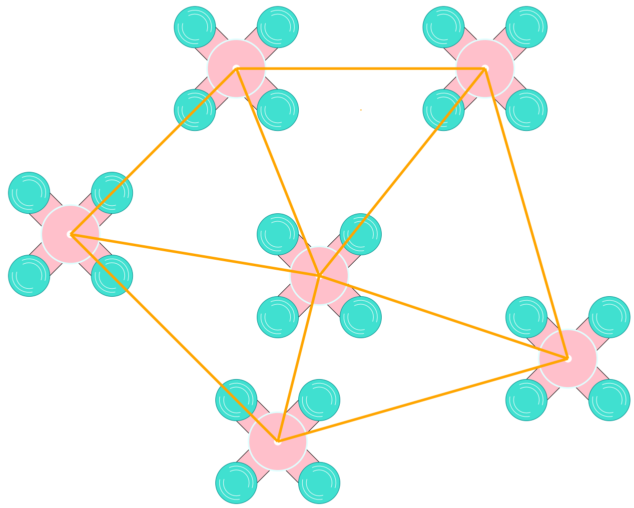


Figure 1.1: Concept diagram of a self-localizing formation of robots that use one UWB tag per robot for range-based relative pose estimation.

the robots, and no information about the direction from which the range measurements are taken. This results in multiple possible relative poses between robots that are consistent with the range measurements, and the true pose is one of these possible poses. The presence of ambiguous poses can lead to inaccurate and imprecise relative pose estimates, which in turn can lead to inaccurate and imprecise formation planning and control. The thesis aims to develop estimation and planning techniques that can mitigate the effects of ambiguous poses and ensure that the relative pose estimates are accurate and precise.

1.1 Background and Related Work

UWB-based relative pose estimation in the absence of fixed UWB anchors at known locations has many challenges. A concept diagram of a self-localizing formation of robots that use one UWB tag per robot for range-based relative pose estimation is shown in Figure 1.1. In this setup, with a range-based approach, there is an infinite number of relative positions between the robots that result in the same range measurements [11]. In fact, if the entire team rotates, flips, and if each robot rotates freely about its center, the range measurements will remain the same. The multiple possible ambiguous poses may cause estimators to converge to the wrong pose.

A common strategy used to obtain an observable system is by ensuring that the robots are constantly moving relative to each other, which satisfies the *persistency of excitation* condition [1–3, 12, 13]. This condition ensures that the range measurements are constantly taken from different directions, which in turn ensures that the relative pose is observable.

However, in practice, it is difficult to ensure that the robots are constantly moving relative to each other, and the persistency of excitation condition is not always satisfied. For instance, if a group of robots is going in formation from one point to another, or are not moving, there is no relative motion between the robots. The authors in [14] propose a two-dimensional solution, where the robots are collecting velocity measurements from an onboard optical flow sensor and sharing this data among themselves to ensure that the relative pose is observable. Furthermore, controllers are designed in [15] to ensure that the robots are constantly moving relative to each other.

Another approach is installing more than one UWB tag on each robot [11, 16–20]. In [11], the authors demonstrate the efficacy of utilizing two UWB tags per robot. This approach enables precise and accurate estimation of the relative position among robots in 3D space without persistency of excitation. However, the authors do not address global observability issues, which can still occur in multi-robot systems with two UWB tags per robot. In [21], the authors partially address this issue by proposing a cost function, the minimization of which would provide optimal multi-robot formations that have good observability properties. In these formations, it is shown that the relative pose estimates are accurate and precise. However, the formations achievable by this method are restricted to a specific set of formations, which are unsuitable for many applications such as coverage and exploration.

1.2 Objective and Outline

The work of [11] and [21] build strong foundation towards achieving reliable relative pose estimation capability in multi-robot systems in real world scenarios. Inspired by [11] and [21], each robot is equipped with two UWB tags throughout this thesis. This thesis gives a detailed account of the observability issues in multi-robot systems with two UWB tags per robot, and presents a solution to this problem. The solution consists of two main components: 1) the Gaussian-sum filter for range-based 3D relative pose estimation in the presence of ambiguities, and 2) optimal robot formations that balance range-based observability and user-defined configurations, and starts with a review of the mathematical tools and concepts used in the thesis in Chapter 2.

Chapter 3 is dedicated to the derivation of the Bayesian filtering equations that are used to estimate the relative pose between a team of robots. After introducing the Bayesian filtering equations, Extended Kalman Filter (EKF) is derived. This derivation is carried forward to the Gaussian-sum filter. Finally, the chapter introduces the Particle Filtering (PF) algorithm.

Chapter 4 presents the Gaussian-sum filter suitable for range-based relative pose estimation.

In this chapter, it is shown that, multi-robot systems with two UWB tags per robot can have multiple possible relative poses that correspond to the same set of range measurements. The chapter firstly determines the ambiguous poses using a least-squares estimator that is initialized using a geometric method. The chapter then models these ambiguous poses as a Gaussian mixture model, which is fed into a Gaussian-sum filter to estimate the relative pose between the robots. The chapter also presents the simulation and experimental results that show the effectiveness of the Gaussian-sum filter in estimating the relative pose between the robots.

Chapter 5 presents the optimal robot formations that balance range-based observability and user-defined formation configurations. In this chapter, a cost function is proposed, which is then used to determine the optimal robot formations, designed to balance the observability of the relative pose between the robots and any formation constraints that the user may have. The chapter then presents the simulation and experimental results showing that user-sought goals can be achieved much more effectively using the optimal robot formations that are derived from the proposed cost function, while ensuring that the relative pose estimates between the robots are accurate and precise.

Chapter 2

Preliminaries

2.1 Summary

Before introducing the UWB-aided state estimation and planning problems, this chapter presents the mathematical preliminaries required for the explanation, and derivation of the majority of contributions presented in this thesis. Specific mathematical preliminaries relevant to individual chapters will be described within the respective chapters. Therefore, an expert reader familiar within basic probability, estimation, and matrix Lie groups can move ahead to following chapters.

2.2 Notation

A column vector is denoted with a lower-case bolded $\mathbf{y} \in \mathbb{R}^n$, and a matrix is denoted with an upper-case bolded $\mathbf{Y} \in \mathbb{R}^{m \times n}$. An arbitrary reference frame 'p' is denoted $\mathcal{F}_p$. Physical vectors resolved in $\mathcal{F}_p$ are denoted $\mathbf{v}_p$. The same physical vector resolved in a different frame $\mathcal{F}_q$ is denoted $\mathbf{v}_q$, and is related by a direction cosine matrix (DCM), $\mathbf{C}_{pq} \in SO(3)$, such that $\mathbf{v}_p = \mathbf{C}_{pq}\mathbf{v}_q$ [22], where $SO(3)$ is the Special Orthogonal group in 3D.

The special matrices $\mathbf{1}$ and $\mathbf{0}$ denote appropriately-sized identity and zero matrices, respectively. Subscripts such as $\mathbf{1}_{2\times 2}$ and $\mathbf{0}_{2\times 1}$ may be used to explicitly indicate dimensions.

2.3 Probability Theory

Sensor data in real-world scenarios is noisy. Proper characterization of the noise from sensors is crucial for tasks such as navigation, and mapping. Any estimate of the state of a robot is typically conditioned on noisy sensor measurements, which result in uncertainty in the estimate. Probability theory provides a framework to model this uncertainty and evaluate the confidence in the estimate.

2.4 Probability Density Functions

A continuous random variable $\mathbf{x} \in \mathbb{R}^n$ is assumed to have a probability density function (PDF) $p(\mathbf{x})$ [22]. A continuous PDF is a function $p : \mathbb{R}^n \to \mathbb{R} \geq 0$ that satisfies the *axiom of total probability*,

$$\int_{\mathbf{a}}^{\mathbf{b}} p(\mathbf{x})\ \mathrm{d}\mathbf{x} = 1. \tag{2.1}$$

If the random variable $\mathbf{x} \in [\mathbf{a}, \mathbf{b}]$ is distributed according to the PDF $p(\mathbf{x})$, it is written as $\mathbf{x} \sim p(\mathbf{x})$.

A joint probability density can always be factored into a conditional and an unconditional factor, such that

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y})p(\mathbf{x}|\mathbf{y}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x}). \tag{2.2}$$

Rewriting the joint PDF in terms of the conditional PDF, the Bayes' rule can be expressed as

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})}. \tag{2.3}$$

The marginalization of a joint PDF $p(\mathbf{x}, \mathbf{y})$ with respect to some of the variables, such as $\mathbf{x}$ is defined as

$$\begin{aligned}
\int_{-\infty}^{\infty} p(\mathbf{x}, \mathbf{y})\ \mathrm{d}\mathbf{x} &= \int_{-\infty}^{\infty} p(\mathbf{x}|\mathbf{y})p(\mathbf{y})\ \mathrm{d}\mathbf{x} && (2.4)\\
&= \int_{-\infty}^{\infty} p(\mathbf{y}|\mathbf{x})p(\mathbf{x})\ \mathrm{d}\mathbf{x} && (2.5)\\
&= p(\mathbf{y}) \underbrace{\int_{-\infty}^{\infty} p(\mathbf{x}|\mathbf{y})\ \mathrm{d}\mathbf{x}}_{=1} && (2.6)\\
&= p(\mathbf{y}). && (2.7)
\end{aligned}$$

Note that, the indefinite integral is taken to be from $-\infty$ to ∞ throughout this thesis. Subsequently,

$$\begin{aligned}
p(\mathbf{x}|\mathbf{y}) &= \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} && (2.8)\\
&= \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{\int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})\ \mathrm{d}\mathbf{x}} && (2.9)\\
&\triangleq \frac{1}{\eta} p(\mathbf{y}|\mathbf{x})p(\mathbf{x}). && (2.10)
\end{aligned}$$

2.4.1 Gaussian Distributions

A special case of the PDF is the Gaussian distribution, and is widely used distribution in estimation theory. The Gaussian distribution is unimodal, and it has mathematical properties that allow for computationally efficient solutions to many estimation problems. The n-dimensional Gaussian distribution is defined by the mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$, and is denoted as $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. The covariance matrix is symmetric positive definite, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^\mathsf{T} > \mathbf{0}$, and therefore invertible. The PDF of the Gaussian distribution is given by

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right). \tag{2.11}$$

The expectation operator is defined as

$$E[\mathbf{x}] = \int \mathbf{x} p(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \boldsymbol{\mu}, \tag{2.12}$$

where $E[\cdot]$ denotes the expectation operator.

2.4.2 Passing a Gaussian through a Nonlinearity

Let, $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$ be a nonlinear function. Consider $\mathbf{y} = \mathbf{g}(\mathbf{x}) + \boldsymbol{\eta}$ where $\mathbf{x} \sim p(\mathbf{x})$, and $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ represents sensor with the noisy output $\mathbf{y}$ corrupted by zero-mean Gaussian noise with covariance $\mathbf{R}$. The PDF of the output $\mathbf{y}$ is

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \tag{2.13}$$

where,

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mathbf{g}(\mathbf{x}), \mathbf{R}), \tag{2.14}$$

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx}). \tag{2.15}$$

To approximately compute (2.13) various approaches can be taken, such as the use of sigma points or linearization [22]. To approximately compute (2.13) via linearization, first note that the linealization of $\mathbf{g}(\cdot)$ about the mean $\boldsymbol{\mu}_x$ is computed as,

$$\mathbf{g}(\mathbf{x}) \approx \mathbf{g}(\boldsymbol{\mu}_x) + \mathbf{G}(\mathbf{x} - \boldsymbol{\mu}_x), \tag{2.16}$$

$$\mathbf{G} = \left.\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\boldsymbol{\mu}_x} \triangleq [\mathbf{g}_1, \ldots, \mathbf{g}_n], \tag{2.17}$$

where $\mathbf{g}_i$ is the i-th column vector of $\mathbf{G}$. Let, $\mathbf{g}_i = \left[\frac{\partial g_1}{\partial x_i} \ldots \frac{\partial g_n}{\partial x_i}\right]^\mathsf{T}$ be the i-th column vector of

$\mathbf{G}$. This column vector is given as,

$$\mathbf{g}_i = \left.\frac{\partial \mathbf{g}(\mathbf{x})}{\partial x_i}\right|_{\mathbf{x}=\boldsymbol{\mu}_x} \triangleq \lim_{h\to 0} \frac{\mathbf{g}(\boldsymbol{\mu}_x + h\mathbf{1}_i) - \mathbf{g}(\boldsymbol{\mu}_x)}{h}, \tag{2.18}$$

where $\mathbf{1}_i$ is the i-th vector of the natural basis of $\mathbb{R}^n$. The Gaussian approximation of the output $\mathbf{y}$ is then given as,

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_{yy}) = \mathcal{N}(\mathbf{g}(\boldsymbol{\mu}_x), \mathbf{G}\boldsymbol{\Sigma}_{xx}\mathbf{G}^\mathsf{T} + \mathbf{R}), \tag{2.19}$$

The detailed derivation of the above equations can be found in [22].

2.5 Matrix Lie Groups

Matrix Lie groups are used to represent the state robotic systems in the literature. As such, in this thesis, the state of the robot will also be represented using matrix Lie groups. The state of a robot is represented as an element of a matrix Lie group because the state of a robot is often constrained to a manifold, and matrix Lie groups are a natural way to represent such a constrained state [23]. Additionally, there are kinematic singularities and ambiguities that result from Euler angle parametrization of attitude, which makes matrix Lie groups a more suitable representation for attitude [24]. In Matrix Lie groups, the elements are represented as square, invertible matrices, and the group operation is matrix multiplication.

2.5.1 Definitions and Identities

Consider the elements $\mathbf{X}, \mathbf{Y} \in G$, where $G \subset \mathbb{R}^{n\times n}$ is a matrix Lie group. Any matrix Lie group is closed under matrix multiplication, which means, $\mathbf{XY} \in G$. Furthermore, the inverse of an element $\mathbf{X} \in G$ is also in G, such that $\mathbf{X}^{-1} \in G$. The identity element of dimension n, $\mathbf{1} \in G$ satisfies $\mathbf{X1} = \mathbf{1X} = \mathbf{X}$ for all $\mathbf{X} \in G$.

The *matrix Lie algebra* $\mathfrak{g}$, associated with the matrix Lie group G, is the tangent space of G at the identity element $\mathbf{1}$. The elements of the matrix Lie algebra are denoted as $\boldsymbol{\Xi} \in \mathfrak{g}$, and are represented as skew-symmetric matrices, such that $\boldsymbol{\Xi} = -\boldsymbol{\Xi}^\mathsf{T}$. The linear map, $(\cdot)^\wedge : \mathbb{R}^m \to \mathfrak{g}$, maps the elements of the vector space to the matrix Lie algebra, such that,

$$\boldsymbol{\Xi} = \boldsymbol{\xi}^\wedge, \quad \boldsymbol{\xi} \in \mathbb{R}^m. \tag{2.20}$$

Similarly, the inverse map $(\cdot)^\vee : \mathfrak{g} \to \mathbb{R}^m$ maps the elements of the matrix Lie algebra to the vector space, such that,

$$\boldsymbol{\xi} = \boldsymbol{\Xi}^\vee, \quad \boldsymbol{\Xi} \in \mathfrak{g}. \tag{2.21}$$

The exponential map $\exp : \mathfrak{g} \to G$ maps the elements of the matrix Lie algebra to the matrix

Lie group, such that,

$$\mathbf{X} = \exp(\boldsymbol{\xi}^\wedge) \triangleq \operatorname{Exp}(\boldsymbol{\xi}), \quad \boldsymbol{\xi} \in \mathbb{R}^m. \tag{2.22}$$

The logarithmic map $\log : G \to \mathfrak{g}$ maps the elements of the matrix Lie group to the matrix Lie algebra, such that,

$$\boldsymbol{\xi} = \log(\mathbf{X})^\vee \triangleq \operatorname{Log}(\mathbf{X}), \quad \mathbf{X} \in G. \tag{2.23}$$

Both, $\operatorname{Exp}(\cdot) : \mathbb{R}^m \to G$ and $\operatorname{Log}(\cdot) : G \to \mathbb{R}^m$ are defined for conciseness. The exponential map and logarithmic map are the same as the matrix exponential and matrix logarithm, respectively, for matrix Lie groups.

An important definition that primarily appears in the derivation of Jacobians of a function involving matrix Lie group elements is the adjoint operator, $\operatorname{Ad}_{\mathbf{X}} : \mathfrak{g} \to \mathfrak{g}$, such that

$$\operatorname{Ad}_{\mathbf{X}}(\boldsymbol{\xi}^\wedge) = \mathbf{X}\boldsymbol{\xi}^\wedge\mathbf{X}^{-1}. \tag{2.24}$$

Since the adjoint operator is a linear map, there exists a corresponding matrix representation of the adjoint operator, denoted $\mathbf{Ad} : G \to \mathbb{R}^{m\times m}$, such that

$$\mathbf{Ad}(\mathbf{X})\boldsymbol{\xi} = (\mathbf{X}\boldsymbol{\xi}^\wedge\mathbf{X}^{-1})^\vee. \tag{2.25}$$

Finally, another identity that is used often used in the derivation of the Jacobians is

$$\mathbf{p}^{\odot}\boldsymbol{\xi} \triangleq \boldsymbol{\xi}^\wedge\mathbf{p}, \quad \mathbf{p} \in \mathbb{R}^n. \tag{2.26}$$

More details on the definitions and identities can be found in [25].

2.5.2 Perturbation of Matrix Lie Group Elements

Matrix Lie group elements can be perturbed in two ways due to the non-commutativity of matrix multiplication. A generalized “addition” operator $\oplus : G \times \mathbb{R}^m \to G$ and “subtraction” operator $\oplus : G \times G \to \mathbb{R}^m$ are defined for matrix Lie groups, to perturb the group elements. From the left,

$$\bar{\mathbf{X}} \oplus \delta\boldsymbol{\xi} = \exp(\delta\boldsymbol{\xi}^\wedge)\, \bar{\mathbf{X}} \qquad \text{(addition)}, \tag{2.27}$$

$$\bar{\mathbf{X}} \ominus \mathbf{Y} = \log(\bar{\mathbf{X}}\mathbf{Y}^{-1})^\vee \qquad \text{(subtraction)}, \tag{2.28}$$

and from the right,

$$\bar{\mathbf{X}} \oplus \delta\boldsymbol{\xi} = \bar{\mathbf{X}}\, \exp(\delta\boldsymbol{\xi}^\wedge) \qquad \text{(addition)}, \tag{2.29}$$

$$\bar{\mathbf{X}} \ominus \mathbf{Y} = \log(\mathbf{Y}^{-1}\bar{\mathbf{X}})^\vee \qquad \text{(subtraction)}, \tag{2.30}$$

where $\bar{\mathbf{X}}$ is the nominal matrix Lie group elements, and $\delta\boldsymbol{\xi}$ is the perturbation vector.

The previous addition and subtraction definitions can be used to model the perturbation of the matrix Lie group elements. For example, the perturbation of the matrix Lie group element $\bar{\mathbf{X}} \in G$ can be modelled as $\bar{\mathbf{X}} \oplus \delta\boldsymbol{\xi}$, where $\delta\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is a zero-mean Gaussian random variable with covariance $\boldsymbol{\Sigma}$.

2.5.3 Linearization

For an element of a matrix Lie group, $\mathbf{X} = \exp(\delta\boldsymbol{\xi}^\wedge) \in G$, the matrix exponential can be written using the Taylor series expansion as

$$\exp(\delta\boldsymbol{\xi}^\wedge) = \sum_{k=0}^{\infty} \frac{1}{k!}(\delta\boldsymbol{\xi}^\wedge)^k. \tag{2.31}$$

For a small perturbation $\delta\boldsymbol{\xi}$, the second and higher order terms can be neglected, and the first order approximation of the matrix exponential is

$$\mathbf{X} = \exp(\delta\boldsymbol{\xi}^\wedge) \approx \mathbf{1} + \delta\boldsymbol{\xi}^\wedge, \tag{2.32}$$

which is commonly used to linearize nonlinear models.

2.5.4 Derivatives of Matrix Lie Group Elements

As stated in [25], the Jacobian of a function $f : G \to G$, with respect to the matrix Lie group element $\mathbf{X}$ is defined as

$$\left.\frac{D\, f(\mathbf{X})}{D\mathbf{X}}\right|_{\bar{\mathbf{X}}} \triangleq \left.\frac{\partial f(\bar{\mathbf{X}} \oplus \delta\mathbf{x}) \ominus f(\bar{\mathbf{X}})}{\partial \delta\mathbf{x}}\right|_{\delta\mathbf{x}=\mathbf{0}}, \tag{2.33}$$

where the function $\partial f(\bar{\mathbf{X}} \oplus \delta\mathbf{x}) \ominus f(\bar{\mathbf{X}})$ for $\delta\mathbf{x}$ has $\mathbb{R}^m$ as both its domain and codomain, and therefore can be differentiated using any standard technique.

Using the generalized definition above, the *Group Jacobian of G* is simply,

$$\mathbf{J}(\mathbf{x}) \triangleq \frac{D\,\mathrm{Exp}(\mathbf{x})}{D\,\mathbf{x}}, \tag{2.34}$$

where left and right Jacobians are obtained using the left or right definition of the $\oplus$ and $\ominus$ operators, respectively. In this thesis, $\mathbf{J}^l$ and $\mathbf{J}^r$ are used to denote the left and right Jacobians, respectively. Further details on the Jacobians of matrix Lie group elements can be found in [25].

2.5.5 Matrix Lie Groups Useful for Robotics

Now that the basic definitions and identities of matrix Lie groups have been introduced, the following are the matrix Lie groups that are commonly used in robotics, when the state of a robot is an attitude and a position in either a two-dimensional (2D) or three-dimensional space (3D). Only the formal definitions are provided here, and the exact expression of the operators on the different groups can be found in [22].

2.5.5.1 Special Orthogonal Group

Attitude of a robot is often represented by a *direction cosine matrix*, which is an element of the special orthogonal group $SO(2)$, in 2D and $SO(3)$ in 3D. These groups in 2D and 3D are defined as

$$SO(n) \triangleq \{\mathbf{C} \in \mathbb{R}^{n\times n} \mid \mathbf{C}^{\mathsf{T}}\mathbf{C} = \mathbf{1}, \det(\mathbf{C}) = +1\}, \quad n = 2, 3, \tag{2.35}$$

respectively.

2.5.5.2 Special Euclidean Group

The *pose*, which collectively represents the position and attitude of a robot is often represented by a *pose transformation matrix*, which is an element of the special Euclidean group $SE(2)$ in 2D and $SE(3)$ in 3D. Considering that the relative attitude and position between one robot and another is $(\mathbf{C}, \mathbf{r})$, the special Euclidean groups in 2D and 3D are defined as

$$SE(2) \triangleq \left\{ \mathbf{T} \in \mathbb{R}^{3\times 3} \quad \middle| \quad \mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0} & 1 \end{bmatrix}, \quad \mathbf{C} \in SO(2), \quad \mathbf{r} \in \mathbb{R}^2 \right\}, \tag{2.36}$$

$$SE(3) \triangleq \left\{ \mathbf{T} \in \mathbb{R}^{4\times 4} \quad \middle| \quad \mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0} & 1 \end{bmatrix}, \quad \mathbf{C} \in SO(3), \quad \mathbf{r} \in \mathbb{R}^3 \right\}, \tag{2.37}$$

respectively.

2.5.6 Composite Groups

In this thesis, a *composite group* is a tuple of N matrix Lie groups $G_1, \ldots, G_N$ [26], with elements of the form

$$\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N) \in G_1 \times \ldots \times G_N. \tag{2.38}$$

These elements form a matrix Lie group where the group operation, inverse, identity, exponential map, and logarithmic map are defined elementwise. For instance, the $\oplus$ operator for this group is defined as

$$\mathbf{X} \oplus \delta\mathbf{x} = (\mathbf{X}_1 \oplus \delta\mathbf{x}_1, \ldots, \mathbf{X}_N \oplus \delta\mathbf{x}_N), \tag{2.39}$$

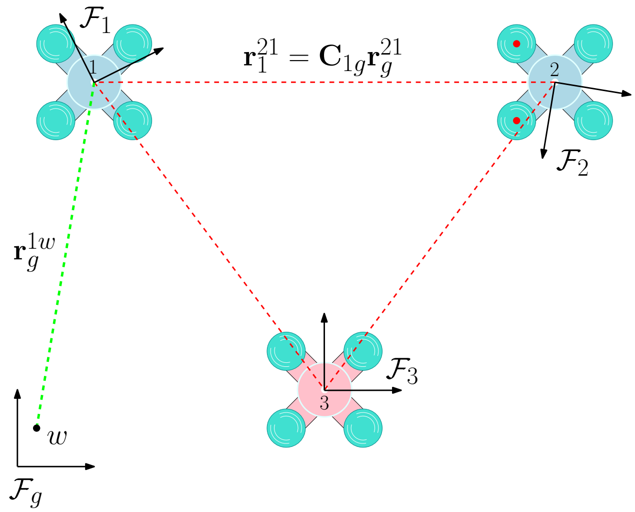


Figure 2.1: Problem setup for formation control.

where $\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{x}_1^\mathsf{T} & \dots & \delta\mathbf{x}_N^\mathsf{T} \end{bmatrix}^\mathsf{T}$.

2.6 Formation Control

Formation control is a subfield of multi-robot systems, where the objective is to control the relative positions and orientations of a group of robots to achieve a desired formation. The formation control is divided into two major parts: the formation shape control and the formation motion control. The formation shape control is responsible for maintaining the

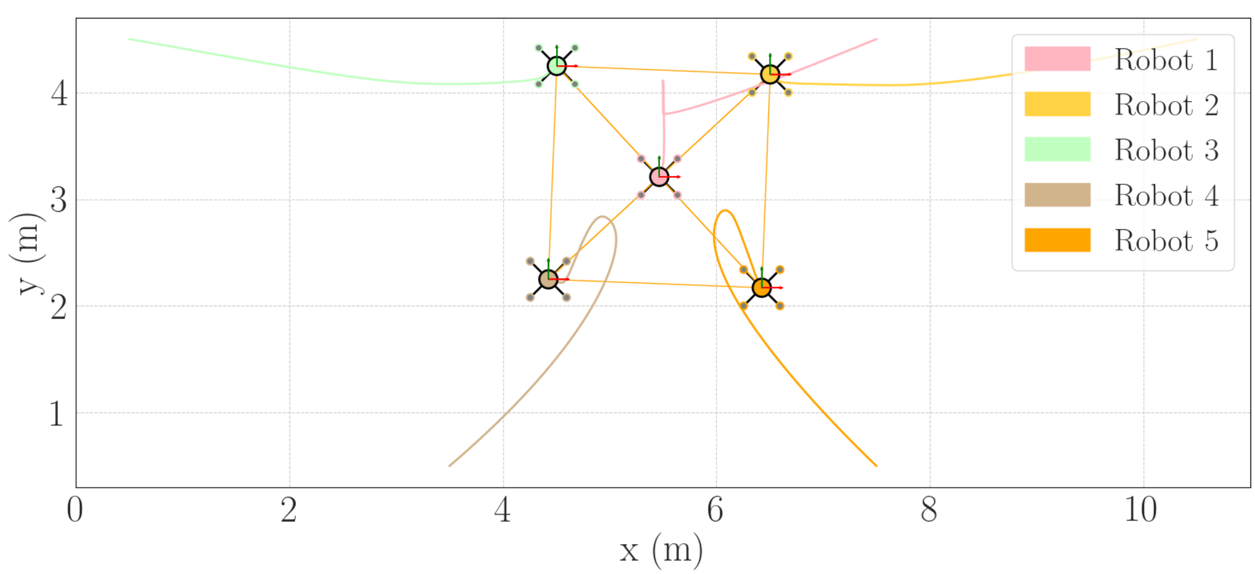


Figure 2.2: Formation control for a square formation with one robot in the middle.

desired relative positions and orientations of the robots, while the formation motion control is responsible for controlling the motion of the robots to reach a desired target location. The formation control problem is challenging due to the nonlinear dynamics of the robots, and the constraints imposed by the formation shape.

Consider N robots, an example of which is shown in Fig. 2.1. The set $\mathcal{P} \triangleq \{1, \dots, N\}$ denotes the Robot IDs. The position of point i affixed to Robot i relative to a static point w, resolved in the global reference frame $\mathcal{F}_g$, is denoted as $\mathbf{r}_g^{iw}$. The attitude of $\mathcal{F}_i$, a frame affixed to Robot i, relative to the global reference frame $\mathcal{F}_g$, is denoted as $\mathbf{C}_{gi}$. The position of point i affixed to Robot i, relative to point j affixed to Robot j, resolved in $\mathcal{F}_i$, is denoted as $\mathbf{r}_i^{ij}$, and the attitude of $\mathcal{F}_i$ relative to $\mathcal{F}_j$ is denoted as $\mathbf{C}_{ij}$. These relative positions and attitudes are related as

$$\mathbf{r}_i^{ij} = \mathbf{C}_{gi}^{\mathsf{T}}(\mathbf{r}_g^{iw} - \mathbf{r}_g^{jw}), \tag{2.40}$$

$$\mathbf{C}_{ij} = \mathbf{C}_{ig}\mathbf{C}_{jg}^{\mathsf{T}}. \tag{2.41}$$

Control inputs to a robot can consist of acceleration or velocity inputs. In this thesis, velocity control inputs are used for the formation control. The velocity control responsible for maintaining the desired formation is defined as

$$\mathbf{u}_i^{\text{formation}/g}(t) = \sum_{\substack{j \in \mathcal{P}, \\ j \neq i}} -k_u ||\mathbf{r}_i^{ij} - \mathbf{r}_i^{ij^*}||\mathbf{r}_i^{ij}(t), \tag{2.42}$$

where $k_u > 0$, and $\mathbf{r}_i^{ij^*}$ is the desired relative position of point i affixed to Robot i with respect to point j affixed to Robot j. For instance, for 5 robots, if the desired formation is a square, with one robot in the middle, the formation controller produces the path shown in Figure 2.2.

Reaching a desired target location is another important aspect of formation control. In this task, the formation is assigned an arbitrary leader and the rest of the robots are followers. The leader is required to reach a desired target location, while the followers maintain the desired formation. The velocity control law for the formation to reach a desired target location is different for the leader and the followers. The velocity control law for the leader is defined as

$$\mathbf{u}_1^{\text{reach target}/g}(t) = \mathbf{u}_1^{\text{formation}/g}(t) + k_v \mathbf{C}_{g1}^{\mathsf{T}}(\mathbf{r}_g^{1w} - \mathbf{r}_g^{1w^*}), \tag{2.43}$$

where $k_v > 0$, and $\mathbf{r}_g^{1w^*}$ is the desired position of the leader, with respect to the global reference frame $\mathcal{F}_g$. The velocity control law for the followers is defined as

$$\mathbf{u}_i^{\text{reach target}/g}(t) = \mathbf{u}_i^{\text{formation}/g}(t) - \mathbf{C}_{i1}\mathbf{u}_1^{\text{formation}/g}(t) + k_v \mathbf{C}_{gi}^{\mathsf{T}}(\mathbf{r}_g^{1w} - \mathbf{r}_g^{1w^*}), \tag{2.44}$$

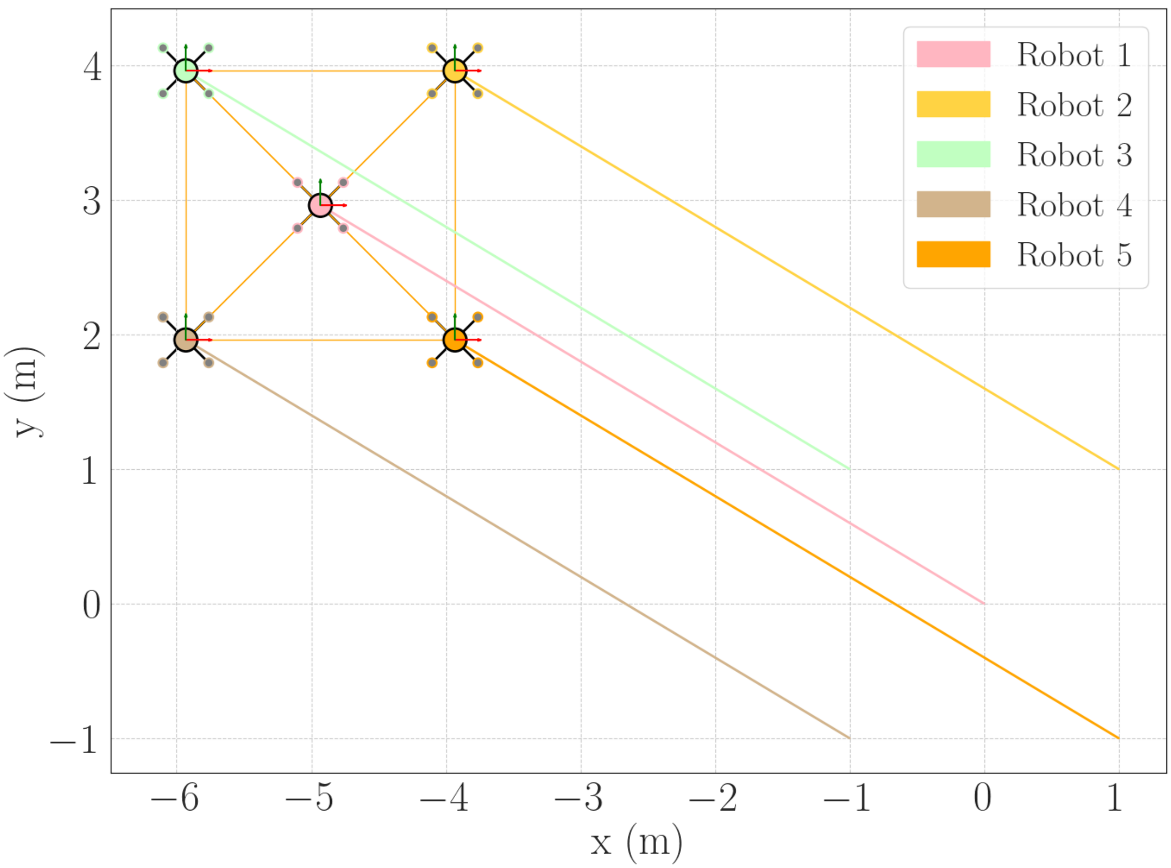


Figure 2.3: Formation control for a square formation with one robot in the middle, and the leader reaching a desired target location.

where $i \in \mathcal{P}$, and $i \neq 1$. Now, if a square formation similar to the one shown in Figure 2.2 is required to reach a desired target location, the path produced by the formation controller is shown in Figure 2.3. Here, the leader, Robot 1, is required to reach the target location $\mathbf{r}_g^{1w^*} = [5\ \ -5]^\mathsf{T}$ starting from its initial position $\mathbf{r}_g^{1w} = [0\ 0]^\mathsf{T}$, while the followers maintain the desired formation. Attitude control between the robots is not presented because it is not applicable to the thesis. More information on formation control can be found in [27].

Chapter 3

Bayesian Filtering

3.1 Overview

Bayesian inference helps make optimal decisions in the presence of uncertainty. For state estimation of robots, Bayesian filters are used to find the optimal state given noisy, uncertain sensor data. A probabilistic state-space model, composed of the process model and the measurement model, is used to estimate the state of the robot.

- Process model: The process model encodes the prior beliefs of how the state evolves over time. Using the Markov assumption [22, 28], the state at time k is dependent only on the state at time $k-1$, which is denoted as

$$\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}), \tag{3.1}$$

  where $\mathbf{x}_k$ is the state at time k, $\mathbf{u}_{k-1}$ is the process model input at time $k-1$. The input is typically an interoceptive sensor reading, such as accelerometer or gyroscope data.

- Measurement model: The measurement model encodes distribution of the measurement given the state, and is denoted as

$$\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}_k), \tag{3.2}$$

  where $\mathbf{y}_k$ is the measurement at time k. The measurement is typically an exteroceptive sensor reading, such as camera or UWB range data.

The goal of the Bayesian estimation is to estimate the posterior distribution of the current state given the entire history of inputs and measurements, which is denoted as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \check{\mathbf{x}}_0). \tag{3.3}$$

The Bayesian filter consists of a prediction step and correction step. The filter is initialized with a prior guess $p(\check{\mathbf{x}}_0)$. Then the filter is run iteratively for each time-step k to estimate the posterior distribution of the state. The filter is run in two steps,

- Prediction step: For interoceptive sensor data, the state's distribution is predicted using the process model, which is denoted as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0)\, \mathrm{d}\mathbf{x}_{k-1}. \tag{3.4}$$

- Correction step: For exteroceptive sensor data, the state's distribution is corrected using the measurement model, which is denoted as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0) = \frac{1}{\eta}\, p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0), \tag{3.5}$$

  where η is a normalization constant, given as,

$$\eta = \int p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0)\, \mathrm{d}\mathbf{x}_k. \tag{3.6}$$

3.2 Discrete-time Process and Measurement Models

The process and measurement models of a filter are typically posed as discrete-time nonlinear state-space models instead of the probabilistic state space models given in (3.1) and (3.2). They are given as,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}, \qquad \mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}), \tag{3.7}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \tag{3.8}$$

where $\mathbf{w}_k$ and $\mathbf{v}_k$ are zero-mean Gaussian noise, and $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$ are the process and measurement noise covariance matrices, respectively.

3.3 Extended Kalman Filter

A Gaussian filter assumes that the state distribution is Gaussian,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \check{\mathbf{x}}_0) \approx \mathcal{N}(\mathbf{x}_k|\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k), \tag{3.9}$$

since that whole distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \check{\mathbf{x}}_0)$ is difficult to compute. The filter only computes the mean $\hat{\mathbf{x}}_k$ and covariance $\hat{\mathbf{P}}_k$ of the Gaussian distribution and uses it to approximate the distribution. The extended Kalman filter (EKF) is a Gaussian filter that linearizes the process and measurement models about the current state estimate. The linearized models are,

$$\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \approx \check{\mathbf{x}}_k + \mathbf{A}_{k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) \tag{3.10}$$

$$\mathbf{g}(\mathbf{x}_k) \approx \check{\mathbf{y}}_k + \mathbf{H}_k(\mathbf{x}_k - \check{\mathbf{x}}_k), \tag{3.11}$$

where

$$\check{\mathbf{x}}_k = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}), \tag{3.12}$$

$$\mathbf{A}_{k-1} = \left.\frac{\partial \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})}{\partial \mathbf{x}_{k-1}}\right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}}, \tag{3.13}$$

and

$$\check{\mathbf{y}}_k = \mathbf{g}(\check{\mathbf{x}}_k), \tag{3.14}$$

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{g}(\mathbf{x}_k)}{\partial \mathbf{x}_k}\right|_{\check{\mathbf{x}}_k}. \tag{3.15}$$

Using basic statistical properties, and the formula for passing a Gaussian through a stochastic nonlinearity in (2.19), as derived in [22, Chap. 4, Pg. 116], it can be shown that,

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) = \mathcal{N}(\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}), \mathbf{Q}_{k-1}) \tag{3.16}$$

$$\approx \mathcal{N}(\check{\mathbf{x}}_k + \mathbf{A}_{k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}), \mathbf{Q}_{k-1}), \tag{3.17}$$

$$p(\mathbf{y}_k | \mathbf{x}_k) = \mathcal{N}(\mathbf{g}(\mathbf{x}_k), \mathbf{R}_k) \tag{3.18}$$

$$\approx \mathcal{N}(\check{\mathbf{y}}_k + \mathbf{H}_k(\mathbf{x}_k - \check{\mathbf{x}}_k), \mathbf{R}_k), \tag{3.19}$$

where $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are additive noise terms to the process and measurement models, respectively. Now, the above expressions lead to the prior and posterior distributions,

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0) = \int \mathcal{N}(\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}), \mathbf{Q}_{k-1}) \mathcal{N}(\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{P}}_{k-1})\, \mathrm{d}\mathbf{x}_{k-1} \tag{3.20}$$

$$\Rightarrow \mathcal{N}(\check{\mathbf{x}}_k, \check{\mathbf{P}}_k) = \mathcal{N}(\check{\mathbf{x}}_k, \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^{\mathsf{T}} + \mathbf{Q}_{k-1}). \tag{3.21}$$

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0) = \frac{1}{\eta}\, \mathcal{N}(\mathbf{g}(\mathbf{x}_k), \mathbf{R}_k) \mathcal{N}(\check{\mathbf{x}}_k, \check{\mathbf{P}}_k) \tag{3.22}$$

$$\Rightarrow \mathcal{N}(\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k) = \frac{1}{\eta} \mathcal{N}(\mathbf{g}(\check{\mathbf{x}}_k), \mathbf{V}_k) \mathcal{N}(\check{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \check{\mathbf{y}}_k), (\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)\check{\mathbf{P}}_k), \tag{3.23}$$

where $\mathbf{K}_k$ is the Kalman gain, and $\mathbf{V}_k = \mathbf{H}_k\check{\mathbf{P}}_k\mathbf{H}_k^{\mathsf{T}} + \mathbf{R}_k$ is the covariance of the predicted measurement from the measurement model. Note that, (3.21) is derived using the steps for passing a Gaussian through a stochastic nonlinearity given in Section 2.19. The detailed derivation of (3.23) is given in [22, Chap. 4, Pg. 116]. Following (3.23), the normalization constant, η, is given as,

$$\eta = \int \mathcal{N}(\mathbf{g}(\mathbf{x}_k), \mathbf{R}_k) \mathcal{N}(\check{\mathbf{x}}_k, \check{\mathbf{P}}_k)\, \mathrm{d}\mathbf{x}_k \tag{3.24}$$

$$= \int \mathcal{N}(\mathbf{g}(\check{\mathbf{x}}_k), \mathbf{V}_k) \mathcal{N}(\check{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \check{\mathbf{y}}_k), (\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)\check{\mathbf{P}}_k)\, \mathrm{d}\mathbf{x}_k. \tag{3.25}$$

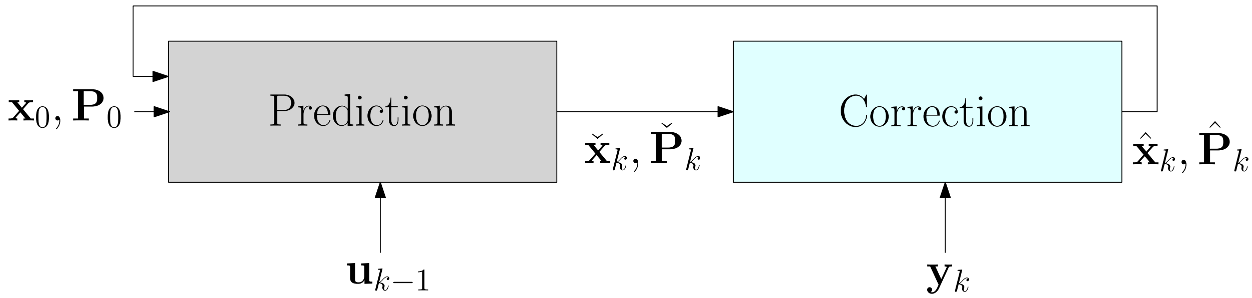


Figure 3.1: Structure of the Extended Kalman Filter

Inside the integral, the first factor is independent of $\mathbf{x}_k$, and the second factor is a Gaussian distribution that sums to 1. Therefore,

$$\eta = \mathcal{N}(\mathbf{g}(\check{\mathbf{x}}_k), \mathbf{V}_k), \tag{3.26}$$

and the posterior distribution is simply,

$$\mathcal{N}(\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k) = \mathcal{N}(\check{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \check{\mathbf{y}}_k), (\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)\check{\mathbf{P}}_k). \tag{3.27}$$

The EKF algorithm is given as,

- Prediction step:

$$\check{\mathbf{x}}_k = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}), \tag{3.28}$$

$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^{\mathsf{T}} + \mathbf{Q}_{k-1}. \tag{3.29}$$

- Correction step:

$$\mathbf{V}_k = \mathbf{H}_k\check{\mathbf{P}}_k\mathbf{H}_k^{\mathsf{T}} + \mathbf{R}_k, \tag{3.30}$$

$$\mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{H}_k^{\mathsf{T}}\mathbf{V}_k^{-1}, \tag{3.31}$$

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \check{\mathbf{y}}_k), \tag{3.32}$$

$$\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)\check{\mathbf{P}}_k. \tag{3.33}$$

The structure of the EKF is shown in Fig. 3.1. A detailed derivation of the EKF can be found in [22].

3.4 Gaussian-Sum Filter Derivation

The Gaussian sum filter (GSF) is a Gaussian filter that approximates the posterior distribution as a sum of Gaussians. As such, it is assumed that, the PDF of the state is a mixture of Gaussians, and is given as,

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1},\mathbf{u}_{1:k-1},\check{\mathbf{x}}_0) \approx \sum_{i=1}^{N} w_{k-1}^{(i)} \mathcal{N}(\hat{\mathbf{x}}_{k-1}^{(i)}, \hat{\mathbf{P}}_{k-1}^{(i)}). \tag{3.34}$$

The GSF algorithm can be derived by inserting the Gaussian sum approximation into the prediction and correction steps of the EKF. It follows that,

- Prediction step: In the density $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1},\mathbf{u}_{1:k-1},\check{\mathbf{x}}_0)$, a dependence on $\mathbf{x}_{k-1}$ can be inserted through marginalization that gives,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1},\mathbf{u}_{1:k-1},\check{\mathbf{x}}_0) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{u}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1},\mathbf{u}_{1:k-1},\check{\mathbf{x}}_0)\,\mathrm{d}\mathbf{x}_{k-1}, \tag{3.35}$$

where it is known that,

$$p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{u}_{k-1}) = \mathcal{N}\big(\mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1}), \mathbf{Q}_{k-1}\big), \tag{3.36}$$

which results in,

$$\begin{aligned} &p(\mathbf{x}_k|\mathbf{y}_{1:k-1},\mathbf{u}_{1:k-1},\check{\mathbf{x}}_0) \\ &\quad= \sum_{i=1}^{N} w_{k-1}^{(i)} \underbrace{\int \mathcal{N}\big(\mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1}), \mathbf{Q}_{k-1}\big) \mathcal{N}(\hat{\mathbf{x}}_{k-1}^{(i)}, \hat{\mathbf{P}}_{k-1}^{(i)})\,\mathrm{d}\mathbf{x}_{k-1}}_{\text{EKF prediction step}} && (3.37) \\ &\quad= \sum_{i=1}^{N} w_{k-1}^{(i)} \mathcal{N}(\check{\mathbf{x}}_k^{(i)}, \check{\mathbf{P}}_k^{(i)}), && (3.38) \end{aligned}$$

where,

$$\begin{aligned} \check{\mathbf{x}}_k^{(i)} &= \mathbf{f}(\hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{k-1}), && (3.39) \\ \check{\mathbf{P}}_k^{(i)} &= \mathbf{A}_{k-1}^{(i)} \hat{\mathbf{P}}_{k-1}^{(i)} \mathbf{A}_{k-1}^{(i)} + \mathbf{Q}_{k-1}. && (3.40) \end{aligned}$$

- Correction step: Using Bayes' rule, the posterior distribution is given as,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k},\mathbf{u}_{1:k-1},\check{\mathbf{x}}_0) = \frac{p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k-1},\mathbf{u}_{1:k-1},\check{\mathbf{x}}_0)}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1},\mathbf{u}_{1:k-1},\check{\mathbf{x}}_0)}, \tag{3.41}$$

where, it is known that,

$$p(\mathbf{y}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{g}(\mathbf{x}_k), \mathbf{R}_k). \tag{3.42}$$

Now, the posterior distribution is

$$p(\mathbf{x}_k|\mathbf{y}_{1:k},\mathbf{u}_{1:k-1},\check{\mathbf{x}}_0) = \frac{1}{\eta}\sum_{i=1}^{N} w_{k-1}^{(i)}\mathcal{N}(\mathbf{g}(\mathbf{x}_k),\mathbf{R}_k)\mathcal{N}(\check{\mathbf{x}}_k^{(i)},\check{\mathbf{P}}_k^{(i)}) \tag{3.43}$$

$$= \frac{1}{\eta}\sum_{i=1}^{N} w_{k-1}^{(i)}\underbrace{\mathcal{N}(\mathbf{g}(\check{\mathbf{x}}_k^{(i)}),\mathbf{V}_k^{(i)})\mathcal{N}(\hat{\mathbf{x}}_k^{(i)},\hat{\mathbf{P}}_k^{(i)})}_{\text{Refer to Eq. (3.23)}}, \tag{3.44}$$

where the normalization constant η is

$$\eta = \sum_{i=1}^{N} w_{k-1}^{(i)}\mathcal{N}(\mathbf{g}(\check{\mathbf{x}}_k^{(i)}),\mathbf{V}_k^{(i)}), \tag{3.45}$$

as derived in (3.26). Therefore, the weights of the Gaussian sum filter at timestep k are given as

$$w_k^{(i)} = \frac{w_{k-1}^{(i)}\mathcal{N}(\mathbf{g}(\check{\mathbf{x}}_k^{(i)}),\mathbf{V}_k^{(i)})}{\sum_{i=1}^{N} w_{k-1}^{(i)}\mathcal{N}(\mathbf{g}(\check{\mathbf{x}}_k^{(i)}),\mathbf{V}_k^{(i)})}. \tag{3.46}$$

In summary, the GSF algorithm is given as,

- Prediction step:

$$\check{\mathbf{x}}_k^{(i)} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^{(i)},\mathbf{u}_{k-1}), \tag{3.47}$$

$$\check{\mathbf{P}}_k^{(i)} = \mathbf{A}_{k-1}^{(i)}\hat{\mathbf{P}}_{k-1}^{(i)}\mathbf{A}_{k-1}^{(i)} + \mathbf{Q}_{k-1}. \tag{3.48}$$

- Correction step:

$$\mathbf{V}_k^{(i)} = \mathbf{H}_k^{(i)}\check{\mathbf{P}}_k^{(i)}\mathbf{H}_k^{(i)\mathsf{T}} + \mathbf{R}_k, \tag{3.49}$$

$$\mathbf{K}_k^{(i)} = \check{\mathbf{P}}_k^{(i)}\mathbf{H}_k^{(i)\mathsf{T}}\mathbf{V}_k^{(i)-1}, \tag{3.50}$$

$$\hat{\mathbf{x}}_k^{(i)} = \check{\mathbf{x}}_k^{(i)} + \mathbf{K}_k^{(i)}(\mathbf{y}_k - \check{\mathbf{y}}_k^{(i)}), \tag{3.51}$$

$$\hat{\mathbf{P}}_k^{(i)} = (\mathbf{1} - \mathbf{K}_k^{(i)}\mathbf{H}_k^{(i)})\check{\mathbf{P}}_k^{(i)}, \tag{3.52}$$

$$w_k^{(i)} = \frac{w_{k-1}^{(i)}\mathcal{N}(\mathbf{g}(\check{\mathbf{x}}_k^{(i)}),\mathbf{V}_k^{(i)})}{\sum_{i=1}^{N} w_{k-1}^{(i)}\mathcal{N}(\mathbf{g}(\check{\mathbf{x}}_k^{(i)}),\mathbf{V}_k^{(i)})}. \tag{3.53}$$

- State estimate:

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{N} w_k^{(i)}\hat{\mathbf{x}}_k^{(i)}, \tag{3.54}$$

$$\hat{\mathbf{P}}_k = \sum_{i=1}^{N} w_k^{(i)}\Big(\check{\mathbf{P}}_k^{(i)} + (\hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k)(\hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k)^{\mathsf{T}}\Big). \tag{3.55}$$

The structure of the GSF is shown in Fig. 3.2.

At timestep k :

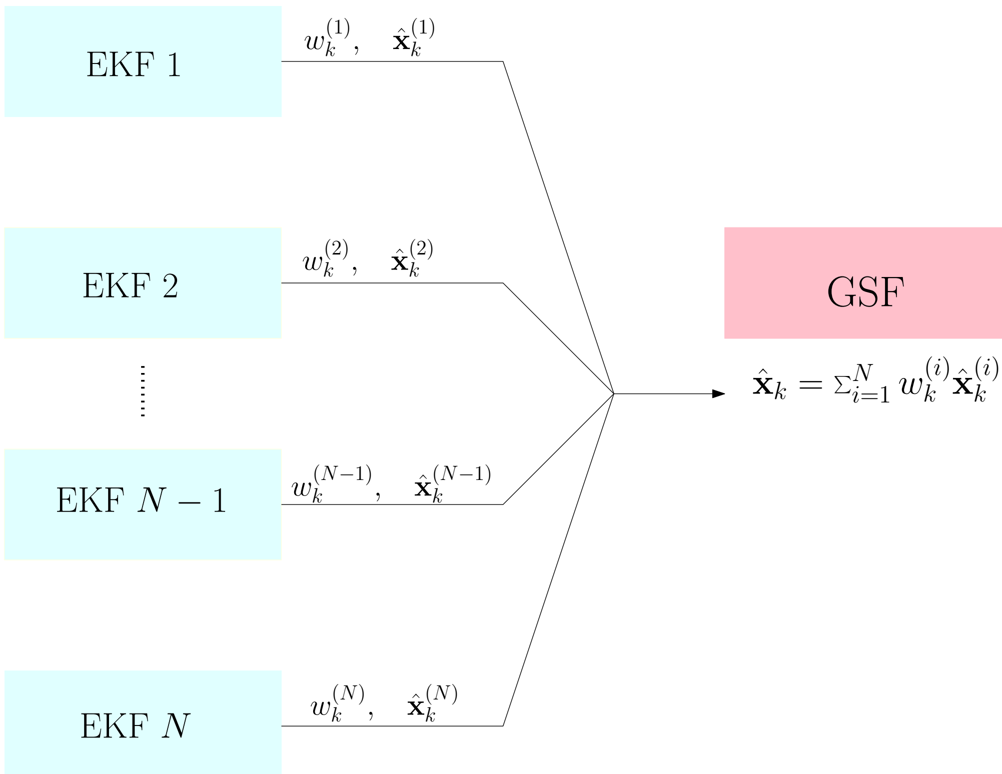


Figure 3.2: Structure of the Gaussian-sum filter (GSF) with N modes. The EKFs are run in parallel and the posterior density at time-step k is represented as a Gaussian sum of M modes, with the i^{th} mode weighted using $w_k^{(i)}$.

3.5 Particle Filter

The particle filter (PF) is a nonparametric filter that approximates the posterior distribution as a set of weighted particles. Assume that there are $\mathbf{x}_{k-1}^{(i)}, i = 1, \ldots, N$ samples at timestep $k-1$, with weights $w_{k-1}^{(i)}$, which together represent $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-2}, \check{\mathbf{x}}_0)$. The particle filter algorithm is given as,

- Prediction step:
  1. Draw N samples $\mathbf{w}_k^{(i)}$ from the noise distribution $p(\mathbf{w}_k)$.
  2. Compute the predicted particles with

$$\mathbf{x}_k^{(i)} = \mathbf{f}(\mathbf{x}_{k-1}^{(i)}, \mathbf{u}_{k-1}) + \mathbf{w}_k^{(i)}, \tag{3.56}$$

  which now approximates $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0)$.
- Correction step:

1. Compute the un-normalized importance weights as,

$$w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{y}_k|\mathbf{x}_k^{(i)}) = w_{k-1}^{(i)} \mathcal{N}(\mathbf{g}(\mathbf{x}_k^{(i)}), \mathbf{R}_k) \tag{3.57}$$

   and normalize them to sum to 1.
2. Resampling step: Resample N particles with replacement from the set $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}$, where the probability of selecting a particle is proportional to its weight. More information on resampling can be found in [29].

The particle filter is a powerful tool for state estimation in nonlinear, non-Gaussian systems. However, it suffers from the curse of dimensionality, where the number of particles required to represent the posterior distribution grows exponentially with the state dimension. This makes the particle filter computationally expensive for high-dimensional state spaces. A detailed derivation of the particle filter can be found in [28].

3.6 Consistency

The *consistency* of a filter is used to evaluate whether the statistics reported by the filter matches the true statistics of the state $\mathbf{x}$. Let the filter's estimated mean and covariance be $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{P}}_k$, respectively, at timestep k. Then, according to [30], the filter is consistent if

$$E\left[\mathbf{x}_k - \hat{\mathbf{x}}_k\right] = 0, \tag{3.58}$$

$$E\left[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^\mathsf{T}\right] = \hat{\mathbf{P}}_k, \tag{3.59}$$

where $E[\cdot]$ is the expectation operator. The first equation ensures that the filter is unbiased, and the second equation ensures that the filter's covariance is equal to the true covariance of the state. The consistency of the filter can be evaluated using the *normalized estimation error squared* (NEES) and *normalized innovation squared* (NIS) metrics. The NEES is given as

$$\epsilon_k = (\mathbf{x}_k - \hat{\mathbf{x}}_k)^\mathsf{T} \hat{\mathbf{P}}_k^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k). \tag{3.60}$$

If the filter is consistent, then $\epsilon_k \sim \chi_n^2$ where χ_n^2 is the chi-squared distribution with n degrees of freedom. For N Monte Carlo trials, the N-run average NEES is given as

$$\bar{\epsilon}_k = \frac{1}{N} \sum_{i=1}^{N} \epsilon_k^{(i)}, \tag{3.61}$$

where i is the Monte Carlo trial index. If the filter is consistent, then $N\bar{\epsilon}_k \sim N\chi_n^2$. The NEES test requires ground truth data, which is not available in real-time when a filter is used. The NIS test, however, requires that the innovation $\mathbf{z}_k = \mathbf{y}_k - \check{\mathbf{y}}_k$ should satisfy the measurement estimate covariance $\mathbf{V}_k$. The NIS is given as,

$$\nu_k = \mathbf{z}_k^{\mathsf{T}} \mathbf{V}_k^{-1} \mathbf{z}_k, \tag{3.62}$$

and if the measurement is not an outlier, then $\nu_k \sim \chi_m^2$ where m is the dimension of the measurement. The NIS test is used as an outlier-rejection test, where if ν_k is greater than a threshold, then the measurement is considered an outlier and is rejected from the estimator correction step in real-time. Refer to [30] for more information on the NIS test.

Chapter 4

Gaussian-Sum Filter for Range-based 3D Relative Pose Estimation in the Presence of Ambiguities

4.1 Summary

Three-dimensional relative pose estimation using range measurements oftentimes suffers from a finite number of non-unique solutions, or *ambiguities*. This chapter: 1) identifies and accurately estimates all possible ambiguities in 2D; 2) treats them as components of a Gaussian mixture model; and 3) presents a computationally-efficient estimator, in the form of a Gaussian-sum filter (GSF), to realize range-based relative pose estimation in an infrastructure-free, 3D, setup. This estimator is evaluated in simulation and experiment and is shown to avoid divergence to local minima induced by the ambiguous poses. Furthermore, the proposed GSF outperforms an extended Kalman filter, demonstrates similar performance to the computationally-demanding particle filter, and is shown to be consistent.

4.2 Introduction

The relative pose needs to be accurately estimated to realize autonomous multi-robot tasks. Sensors such as cameras with object-detection ability [31] or LiDAR [32] can satisfy the relative pose estimation requirement, but they are computationally expensive. For infrastructure-free localization, it in shown [11] that placing two UWB tags per robot ensures "local observability". This setup combined with an interoceptive IMU or velocity readings allow for infrastructure-free relative pose estimation in 3D. However, even with two tags per robot, the range measurements yield multiple solutions for relative robot poses, referred to as "discrete" *ambiguities*, which are not addressed in [11].

These ambiguities form a multi-modal distribution of relative poses that the estimator must account for. Adding more UWB tags per robot reduces the number of ambiguities at the cost of the tags not communicating at their highest data rate. In fact, even with three strategically-positioned tags, only relative robot positions can be disambiguated, while relative attitude still remains ambiguous. Therefore, for a range-based approach, designing estimators that can handle these ambiguities is of great importance.

In the face of ambiguities, Gaussian-based filters, such as an EKF, can perform poorly since they assume that the distribution is unimodal [33]. A particle filter (PF) can handle a multi-modal distribution [34–36], but it is computationally expensive due to the need for many particles to describe the multi-modality [37]. Range-based localization of the ambiguous position of one robot with the help of three static anchors with known positions in 2D has been addressed using a Gaussian-sum filter (GSF) in [38]. Additionally, signal map measurements often exhibit multi-modality while tracking multiple targets [39], and a Gaussian mixture model (GMM) helps isolate the "true" measurement for a particular target. In this chapter, the ideas presented in [38, 39] are extended to design a localization solution involving a Gaussian-sum filter (GSF) where the "true" relative pose between multiple robots is identified among the ambiguous poses in 3D. Note that, [38] solves a single-robot localization problem, where one robot has one range sensor affixed to it, which provides distance measurements to three static anchors. Unlike [38], this chapter provides a complete anchor-free 3D pose estimation solution for multi-robot systems. The solution only uses two UWB tags per robot to ensure that the system is locally observable [11], and no static anchors are required.

As such, the key contributions of this chapter are as follows.

- Identification of all the possible ambiguous relative poses between N robots using a geometric approach is presented. The geometric estimates are fed into a least-squares estimator to form a GMM of ambiguous relative poses in 3D. These estimates are used to initialize a GSF to identify the "true" relative pose. Since this GSF is only initialized at the ambiguous poses, it contains the minimum number of Gaussian components required to model the multi-modal state.
- To the best of the Author's knowledge, this is the first work where a GSF is used for anchor-free, range-based 3D relative pose estimation between robots in the presence of ambiguities. Approaching this problem in 3D is non-trivial due to the increased complexity of the state space and the number of ambiguities.
- In simulations and experiments, the proposed estimator involving the GSF is shown to have a similar performance to the PF, while, as expected, being orders of magnitude faster.

The remainder of this chapter is organized as follows. The problem formulation is in Section 4.4 and the GSF is discussed in Section 4.5. The ambiguous pose estimation procedure for initializing the GSF is presented in Section 4.6. The estimator is validated in simulation and experiment in Sections 4.7 and 4.8, respectively.

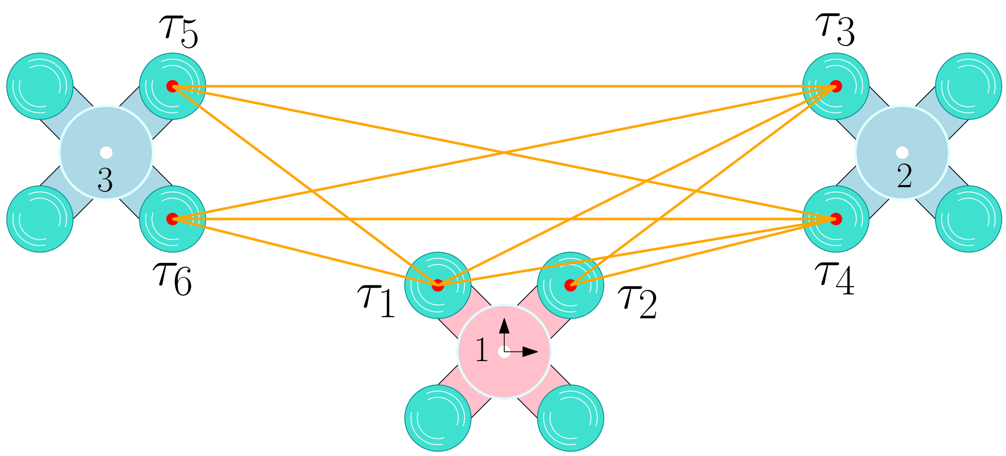


Figure 4.1: Problem setup for a two-tag multi-robot system. Without loss of generality, the pink robot, defined as Robot 1, is considered to be the reference robot.

4.3 Notation and Preliminaries

Consider N robots with IDs, $\mathcal{P} = \{1, \ldots, N\}$. Each robot is equipped with two ranging tags, resulting in a total of $2N$ tags collectively, as shown in Fig. 4.1. The physical points $\tau_1, \ldots, \tau_{2N}$ denote the location of the tags on the robots. The set of tag IDs is denoted as $\mathcal{V} = \{1, \ldots, 2N\}$. A measurement graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes the inter-tag range measurements. The nodes $\mathcal{V} = \{1, \ldots, 2N\}$ are the set of tag IDs and the edges $\mathcal{E}$ denote the set of inter-tag range measurements.

A 2-dimensional orthonormal reference frame $\mathcal{F}_p$ is attached to Robot p. A common global reference frame and a static point are denoted by $\mathcal{F}_g$ and w, respectively. The position of a chosen reference point in Robot p relative to point w, resolved in $\mathcal{F}_p$ is denoted $\mathbf{r}_p^{pw} \in \mathbb{R}^n$, and the robot's translational velocity with respect to another arbitrary reference frame $\mathcal{F}_c$ is denoted $\mathbf{v}_p^{pw/c} \in \mathbb{R}^n$. Vectors resolved in different frames are related by the transformation, $\mathbf{r}_p^{pw} = \mathbf{C}_{pq}\mathbf{r}_q^{pw}$, where $\mathbf{C}_{pq} \in SO(n)$. The angular velocity of $\mathcal{F}_p$ relative to $\mathcal{F}_q$ resolved in $\mathcal{F}_c$ is denoted $\boldsymbol{\omega}_c^{pq}$. For conciseness, Robot p is referred to as R_p in plot legends. The relative pose between Robots p and q is

$$\mathbf{T}_{pq} = \begin{bmatrix} \mathbf{C}_{pq} & \mathbf{r}_p^{qp} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(n), \tag{4.1}$$

where $SE(n)$ is the special Euclidean group in n dimensions.

4.4 Problem Formulation

The poses of all the robots are expressed relative to Robot 1, which is arbitrarily chosen to be the reference robot. As such, the state of the system is

$$\mathbf{x} = (\mathbf{T}_{12}, \ldots, \mathbf{T}_{1N}) \in SE(3)^{N-1}. \tag{4.2}$$

The position of Robot p relative to Robot q, resolved in $\mathcal{F}_1$, is

$$\mathbf{r}_1^{pq} = \mathbf{D}\mathbf{T}_{1p}\mathbf{b} - \mathbf{D}\mathbf{T}_{1q}\mathbf{b}, \tag{4.3}$$

where $\mathbf{D} = [\mathbf{1}_{2\times 2}\ \mathbf{0}_{2\times 1}]$, $\mathbf{b} = [\mathbf{0}_{1\times 2}\ 1]^\mathsf{T}$.

The range measurement of Tag i relative to Tag j in Robots p and q, respectively, is modelled as

$$y_{ij}(\mathbf{x}) = \left\| \mathbf{D}\mathbf{T}_{1p}\tilde{\mathbf{r}}_p^{\tau_i p} - \mathbf{D}\mathbf{T}_{1q}\tilde{\mathbf{r}}_q^{\tau_j q} \right\| + \eta_{ij}, \tag{4.4}$$

where $\tilde{\mathbf{r}} = [\mathbf{r}^\mathsf{T}\ 1]^\mathsf{T}$, and $\eta_{ij} \sim \mathcal{N}(0, \sigma_{ij}^2)$. Therefore, the augmented measurement vector of all the range measurements is

$$\begin{aligned} \mathbf{y} = \mathbf{g}(\mathbf{x}) + \boldsymbol{\eta} = \begin{bmatrix} \cdots & y_{ij}(\mathbf{x}) & \cdots \end{bmatrix}^\mathsf{T} + \boldsymbol{\eta} \in \mathbb{R}^{|\mathcal{E}|}, \\ \forall (i,j) \in \mathcal{E}, \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}),\ \mathbf{R} = \operatorname{diag}(\ldots, \sigma_{ij}^2, \ldots). \end{aligned} \tag{4.5}$$

The objective is to accurately estimate the state $\mathbf{x}$. For this, the interoceptive measurements are each robot's angular and translational velocities as resolved in its body frame, denoted as

$$\mathbf{u}_p = [\boldsymbol{\omega}_p^{pg\mathsf{T}}\ \ \mathbf{v}_p^{pw/g\mathsf{T}}]^\mathsf{T} + \mathbf{w}_p \in \mathbb{R}^m, \quad \mathbf{w}_p \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_p),$$

where $\mathbf{w}_p$ is zero-mean Gaussian noise with covariance $\mathbf{Q}_p$. The relative pose between Robots 1 and p at time-step k is

$$\begin{aligned} \mathbf{T}_{1p_k} &= \mathbf{T}_{g1_k}^{-1}\mathbf{T}_{gp_k} \\ &= \left(\exp(-\Delta t\, \mathbf{u}_{1_{k-1}}^\wedge)\mathbf{T}_{g1_{k-1}}^{-1}\right)\left(\mathbf{T}_{gp_{k-1}} \exp(\Delta t\, \mathbf{u}_{p_{k-1}}^\wedge)\right) \\ &= \exp(-\Delta t\, \mathbf{u}_{1_{k-1}}^\wedge)\mathbf{T}_{1p_{k-1}} \exp(\Delta t\, \mathbf{u}_{p_{k-1}}^\wedge), \\ &\triangleq \mathbf{f}(\mathbf{T}_{1p_{k-1}}, \mathbf{u}_{1_{k-1}}, \mathbf{u}_{p_{k-1}}), \end{aligned} \tag{4.6}$$

where $\Delta t = t_k - t_{k-1}$ is the time interval. The relative poses $\mathbf{T}_{1p_k}$, $p = 2, \ldots, N$, collectively form the state $\mathbf{x}_k$. At timestep k, the range measurement between the UWB tags in Robots p and q, and the measurement model are given in (4.4) and (4.5), respectively.

Estimating the state $\mathbf{x}$ of a multi-robot system with two tags per robot is non-trivial. As shown in Fig. 4.2, in this setup, there is a finite set of discrete relative poses or ambiguities that correspond to the same range measurements. These ambiguities will be referred to as *modes* in the chapter. In 2D, the two obvious ambiguities are modes 1 and 2 since the range measurements are equal in both the modes. Given noisy range measurements, when $y_{1i} \approx y_{1j}$ and $y_{2i} \approx y_{2j}$, there is a likelihood of "flip" ambiguities occurring, where tags τ_i and τ_j swap their positions, yielding modes 3 and 4. These modes present an issue for estimator initialization when robots are static, as there is no motion to disambiguate the multiple modes.

The multi-modal state representing this system can be estimated using a GSF. The GSF is typically initialized by sampling from either a uniform distribution of all possible states or a Gaussian distribution based on the prior knowledge. With limited prior knowledge, both methods can require many Gaussian components in the GSF. In this chapter, a Gaussian component is assigned per ambiguous pose in 2D, which are denoted as modes in Fig. 4.2, to form a GMM that captures the state's multi-modality effectively. This GMM is used to initialize a GSF that isolates the "true" mode when the robots are in motion and avoids divergence to new ambiguities in-flight, which allows accurate and efficient state estimation. This novel initialization method minimizes the number of Gaussian components required in the GSF, thus improving computational efficiency.

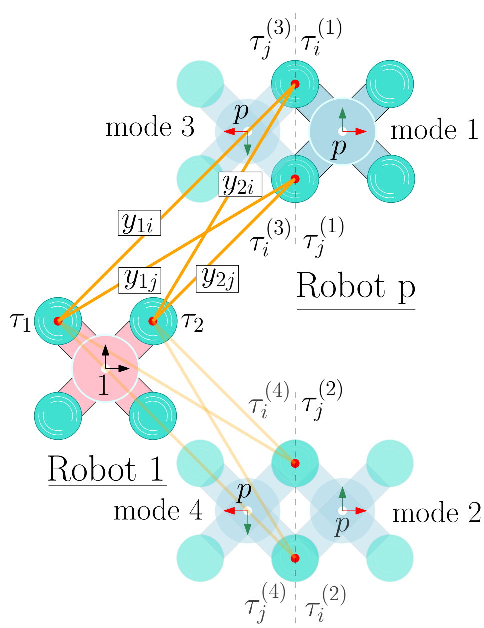


Figure 4.2: Visualization of all the possible ambiguous relative poses between robots 1 and p. The relative pose in mode 1 is the "true" pose and modes 2, 3, and 4 are ambiguities. The range measurements are y_{1i}, y_{1j}, y_{2i}, and y_{2j}.

To initialize the GSF using the proposed GMM, given the challenge posed by measurement noise, a two-step solution is undertaken. Firstly, analytical geometric derivations are used to evaluate all ambiguous poses in 2D as a preliminary guess. Secondly, this guess is refined through a nonlinear least-squares algorithm to get a more accurate estimate. These methodologies are discussed in Section 4.6.

4.5 Gaussian-Sum Filter

The GSF as introduced in Chapter 3 consists of M EKFs, each initialized with an equal weightage at a different initial state, $\check{\mathbf{x}}_0^{(i)}$ and covariance, $\check{\mathbf{P}}_0^{(i)}$, $i = 1, \dots, M$, such that $\sum_{i=1}^{M} w_0^{(i)} = 1$. Each of these EKFs is referred to as a *mode* of the GSF in this chapter.

The process and measurement models require for the GSF prediction and correction step are given in (4.6) and (4.5), respectively. The process model Jacobian, $\mathbf{A}(\mathbf{x})$, and the measurement model Jacobian, $\mathbf{H}(\mathbf{x})$, are given in Section 4.5.1 and Section 4.5.3, respectively.

The primary feature of the GSF is that when a new measurement $\mathbf{y}_k$ is received, the weights are updated by comparing the measurement with the predicted measurement of each mode, which is given by

$$\check{\mathbf{y}}_k^{(i)} = \mathbf{g}(\check{\mathbf{x}}_k^{(i)}), \tag{4.7}$$

where, $\check{\mathbf{x}}_k^{(i)}$ is the predicted states of the i^{th} mode. If a mode's predicted value $\check{\mathbf{y}}_k^{(i)}$ closely matches $\mathbf{y}_k$, it is more likely to be responsible for the observation and thus receives a higher weight and vice versa. The weights quantify the probability of a measurement being associated with each mode, and are updated using (3.53).

The mean estimate of the GSF is a weighted average of the estimates in all the modes, while the covariance is assumed as the covariance of the max-weighted mode since the objective is to detect the true Gaussian mode. According to [40], for matrix Lie groups, the mean estimate is given by

$$\hat{\boldsymbol{\xi}}_k = \sum_{i=1}^{M} w_k^{(i)} \hat{\boldsymbol{\xi}}_k^{(i)}, \text{ where, } \hat{\boldsymbol{\xi}}_k^{(i)} = \mathbf{x}_k^{(i)} \ominus \hat{\mathbf{x}}_{k-1}, \tag{4.8}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} \oplus \hat{\boldsymbol{\xi}}_k, \tag{4.9}$$

$$\hat{\mathbf{P}}_k = \hat{\mathbf{P}}_k^{(i)}, \; i = \arg\max_i w_k^{(i)}, \tag{4.10}$$

Note that, conventionally, the covariance of the GSF is computed as a weighted sum of the covariance of all the modes as shown in (3.55). However, in this work, the covariance of the max-weighted mode is used since it is the most likely mode to be the true mode. As will be shown in Section 4.7, the GSF estimates almost instantaneously coverage to the true mode, and so does the covariance, which supports this assumption. A detailed derivation of the GSF is given in Chapter 3.

4.5.1 Process Model Jacobian

Let $\mathbf{T}_{1p} = \bar{\mathbf{T}}_{1p} \exp(\delta\boldsymbol{\xi}^\wedge)$, where $\bar{\mathbf{T}}_{1p} \in SE(n)$, and $\delta\boldsymbol{\xi} \in \mathbb{R}^m$ is small. Replacing $\mathbf{T}_{1p_k}$ and $\mathbf{T}_{1p_{k-1}}$ by this approximation into (4.6) and left-multiplying both sides by $\bar{\mathbf{T}}_{1p_k}^{-1}$ yields

$$\bar{\mathbf{T}}_{1p_k} \exp(\delta\boldsymbol{\xi}_k^\wedge) = \exp(-\Delta t\, \mathbf{u}_{1_{k-1}}^\wedge)\bar{\mathbf{T}}_{1p_{k-1}} \exp(\delta\boldsymbol{\xi}_{k-1}^\wedge) \exp(\Delta t\, \mathbf{u}_{p_{k-1}}^\wedge) \tag{4.11}$$
$$\Rightarrow \exp(\delta\boldsymbol{\xi}_k^\wedge) = \bar{\mathbf{T}}_{1p_k}^{-1} \exp(-\Delta t\, \mathbf{u}_{1_{k-1}}^\wedge)\bar{\mathbf{T}}_{1p_{k-1}} \exp(\delta\boldsymbol{\xi}_{k-1}^\wedge) \exp(\Delta t\, \mathbf{u}_{p_{k-1}}^\wedge) \tag{4.12}$$

Now, replacing $\bar{\mathbf{T}}_{1p_k} = \exp(-\Delta t\, \mathbf{u}_{1_{k-1}}^\wedge)\bar{\mathbf{T}}_{1p_{k-1}} \exp(\Delta t\, \mathbf{u}_{p_{k-1}}^\wedge)$ into (4.12) it follows that

$$\Rightarrow \exp(\delta\boldsymbol{\xi}_k^\wedge) = \exp(-\Delta t\, \mathbf{u}_{p_{k-1}}^\wedge) \exp(\delta\boldsymbol{\xi}_{k-1}^\wedge) \exp(\Delta t\, \mathbf{u}_{p_{k-1}}^\wedge).$$

Using the adjoint operator given in (2.24), it follows that

$$\delta\boldsymbol{\xi}_k = \mathrm{Ad}(\exp(-\Delta t\, \mathbf{u}_{p_{k-1}}^\wedge))\delta\boldsymbol{\xi}_{k-1}. \tag{4.13}$$

Based on [25],

$$\frac{D\,\mathbf{f}(\mathbf{T}_{1p_{k-1}}, \mathbf{u}_{1_{k-1}}, \mathbf{u}_{p_{k-1}})}{D\,\mathbf{T}_{1p_{k-1}}} = \mathrm{Ad}(\exp(-\Delta t\, \mathbf{u}_{p_{k-1}}^\wedge)). \tag{4.14}$$

Thus, $\mathbf{A}_{k-1}(\mathbf{x})$ a block-diagonal matrix in $\mathbb{R}^{(m\times m)(N-1)}$, where the $(p-1)^{\text{th}}$ block is given by (4.14), for $p = 2, \ldots, N$.

4.5.2 Process Model Noise Covariance

The process noise covariance is a block-diagonal matrix in $\mathbb{R}^{(m\times m)(N-1)}$, where the $(p-1)^{\text{th}}$ block is given by

$$\mathbf{Q}_{1p_{k-1}} = \mathbf{L}_1\mathbf{Q}_1\mathbf{L}_1^\mathsf{T} + \mathbf{L}_p\mathbf{Q}_p\mathbf{L}_p^\mathsf{T}, \text{ where,} \tag{4.15}$$
$$\mathbf{L}_1 = \Delta t\, \mathbf{Ad}(\mathbf{T}_{1p_{k-1}} \exp(\Delta t\, \mathbf{u}_{p_{k-1}}^\wedge))\mathbf{J}^l(\Delta t\, \mathbf{u}_{1_{k-1}}),$$
$$\mathbf{L}_p = \Delta t\, \mathbf{J}^l(-\Delta t\, \mathbf{u}_{p_{k-1}}), \tag{4.16}$$

and are formulated based on [41]. The $\mathbf{J}^l(\cdot)$ operator is the left Jacobian of the exponential map, and $\mathbf{Q}_1$ and $\mathbf{Q}_p$ are the covariance matrices for the velocity inputs $\mathbf{u}_{1_{k-1}}$ and $\mathbf{u}_{p_{k-1}}$, respectively.

4.5.3 Measurement Model Jacobian

The measurement model Jacobian is given by

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} \vdots \\ \mathbf{H}^{ij}(\mathbf{x})^{\mathsf{T}} \\ \vdots \end{bmatrix}, \tag{4.17}$$

where,

$$\mathbf{H}^{ij}(\mathbf{x}) = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{H}_p^{ij}(\mathbf{x}) & \cdots & \mathbf{H}_q^{ij}(\mathbf{x}) & \cdots & \mathbf{0} \end{bmatrix} \qquad \in \mathbb{R}^{1\times m(N-1)},$$

$$\mathbf{H}_p^{ij}(\mathbf{x}) = \boldsymbol{\rho}_{ij}\mathbf{D}\bar{\mathbf{T}}_{1p}\tilde{\mathbf{r}}_p^{\tau_i p \odot} \qquad \in \mathbb{R}^{1\times m}, \tag{4.18}$$

$$\mathbf{H}_q^{ij}(\mathbf{x}) = -\boldsymbol{\rho}_{ij}\mathbf{D}\bar{\mathbf{T}}_{1q}\tilde{\mathbf{r}}_q^{\tau_j q \odot} \qquad \in \mathbb{R}^{1\times m}, \tag{4.19}$$

$$\boldsymbol{\rho}_{ij} = \frac{\mathbf{D}\mathbf{T}_{1p}\tilde{\mathbf{r}}_p^{\tau_i p} - \mathbf{D}\mathbf{T}_{1q}\tilde{\mathbf{r}}_q^{\tau_j q}}{||\mathbf{D}\mathbf{T}_{1p}\tilde{\mathbf{r}}_p^{\tau_i p} - \mathbf{D}\mathbf{T}_{1q}\tilde{\mathbf{r}}_q^{\tau_j q}||}. \tag{4.20}$$

The p^{th} and q^{th} block columns of $\mathbf{H}^{ij}(\mathbf{x})$ are populated by (4.18) and (4.19), respectively. A detailed derivation of the measurement model Jacobian is given in [21].

4.6 GSF Initialization Process

4.6.1 Pose Evaluation using Geometry

The estimation of the four possible solutions for relative poses between Robot 1 and Robot p in 2D, as shown in Fig. 4.2, is a challenging problem. They are first computed using a geometric method. These solutions form a combination of all ambiguous relative poses between Robots 1 to N. Since the robots only have two ranging tags each, to ensure a finite number of solutions, the problem is addressed in 2D, assuming zero relative roll, and pitch between the robots. Assuming, that the robots can be at different heights, to project the range measurements into a 2D plane, the relative height between the robots is taken from the laser-range finders mounted on the robots, and then a 2D projection is performed on the range measurements. This method assumes that the robots are static or hovering over a common flat surface, which is a reasonable assumption for indoor environments.

The notational preliminaries are as follows. The Tags 1 and 2 are in Robot 1 and Tags i and j are in Robot p. The range measurements between Robots 1 and p are y_{1i}, y_{1j}, y_{2i}, and y_{2j}. The unit vector between tags τ_1 and τ_2 is,

$$\mathbf{n}_1 = \frac{1}{d}\mathbf{r}_1^{\tau_2\tau_1},\ d = ||\mathbf{r}_1^{\tau_2\tau_1}||, \quad \text{and} \quad \mathbf{n}_{1\perp} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\mathbf{n}_1, \tag{4.21}$$

is its dextral orthonormal counterpart. Additionally, note that, any attitude $\mathbf{C}_{pq} \in SO(2)$

between the frames $\mathcal{F}_p$ and $\mathcal{F}_q$ is a function of the heading ϕ_{qp} between the frames, and is denoted as $\mathbf{C}_{pq} \triangleq \mathbf{C}_{pq}(\phi_{qp})$ [22].

In Fig. 4.3a, the two possible position vectors between Tags τ_1 and τ_μ, $\mu \in \{i, j\}$, and subsequently, the possible position vectors between Tags τ_i and τ_j are,

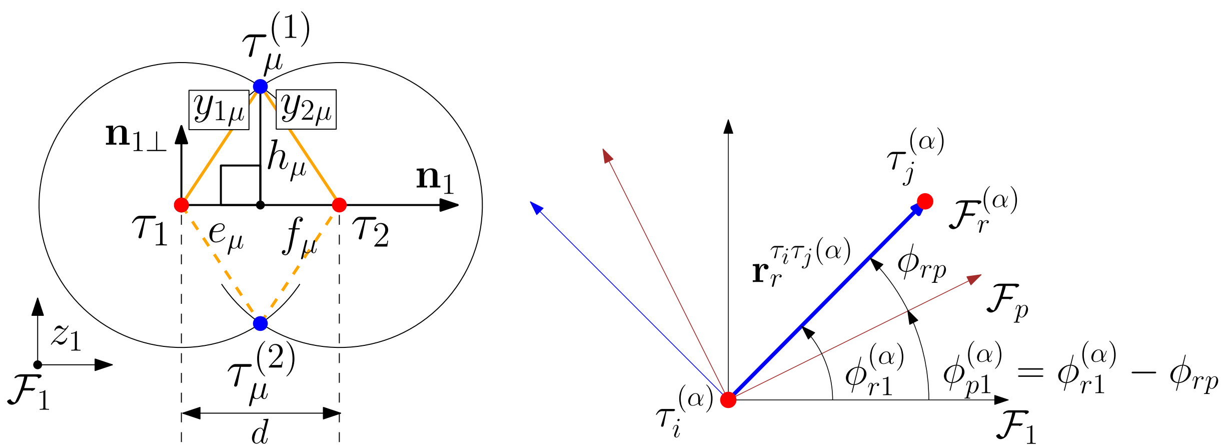
$$e_\mu = \frac{1}{2d}(y_{1\mu}^2 - y_{2\mu}^2 + d^2),\ h_\mu = \sqrt{(y_{1\mu}^2 - e_\mu^2)},\ \mu \in \{i, j\},$$

$$\mathbf{r}_1^{\tau_\mu \tau_1 (1)} = e_\mu \mathbf{n}_1 + h_\mu \mathbf{n}_{1\perp}, \quad \mu \in \{i, j\}, \tag{4.22}$$

$$\mathbf{r}_1^{\tau_\mu \tau_1 (2)} = e_\mu \mathbf{n}_1 - h_\mu \mathbf{n}_{1\perp}, \quad \mu \in \{i, j\}, \tag{4.23}$$

$$\mathbf{r}_1^{\tau_i \tau_j (\alpha)} = \mathbf{r}_1^{\tau_i \tau_1 (\alpha)} - \mathbf{r}_1^{\tau_j \tau_1 (\alpha)}, \quad \alpha = 1,\, 2, \tag{4.24}$$

where α is the mode number of the ambiguity.



(a) Geometry between tags

(b) Geometry between frames

Figure 4.3:(a) Visualization of the geometric relation between tags τ_1, τ_2 of Robot 1 and τ_μ, $\mu \in \{i, j\}$ of Robot p resolved in $\mathcal{F}_1$. The range measurements consist of $y_{1\mu}$ and $y_{2\mu}$, $\mu \in \{i, j\}$. The reference point in Robot 1, 1, and the frame $\mathcal{F}_1$ are arbitrarily defined. (b) Visualization of the relation between frames $\mathcal{F}_1$, $\mathcal{F}_p$, and $\mathcal{F}_r$. Tags τ_i and τ_j are mounted on Robot p. In both figures, the superscript $(\cdot)$ represents the mode number.

A right-handed frame denoted as $\mathcal{F}_r^{(\alpha)}$ whose x-axis is aligned with the physical vector $\underrightarrow{r}^{\tau_i \tau_j (\alpha)}$ is shown in Fig. 4.3b in blue. The heading of $\mathcal{F}_r$ relative to $\mathcal{F}_p$ and $\mathcal{F}_1$, and subsequently the attitude, $\mathbf{C}_{1p}^{(\alpha)}$, in modes 1 and 2 are,

$$\phi_{rp} = \tan^{-1}(y_p/x_p), \qquad \text{s.t. } \mathbf{r}_p^{\tau_i \tau_j} = [x_p \;\; y_p]^\mathsf{T},$$

$$\phi_{r1}^{(\alpha)} = \tan^{-1}(y_1^{(\alpha)}/x_1^{(\alpha)}), \qquad \text{s.t. } \mathbf{r}_1^{\tau_i \tau_j (\alpha)} = [x_1^{(\alpha)} \;\; y_1^{(\alpha)}]^\mathsf{T},$$

$$\mathbf{C}_{1p}^{(\alpha)} = \mathbf{C}_{1r}^{(\alpha)} \mathbf{C}_{pr}^\mathsf{T}, \quad \alpha = 1,\, 2. \tag{4.25}$$

Thus, the relative robot positions in modes 1 and 2 are,

$$\mathbf{r}_1^{p1(\alpha)} = \mathbf{C}_{1p}^{(\alpha)} \mathbf{r}_p^{p\tau_i} + \mathbf{r}_1^{\tau_i \tau_1(\alpha)} + \mathbf{r}_1^{\tau_1 1}, \quad \alpha = 1,\, 2. \tag{4.26}$$

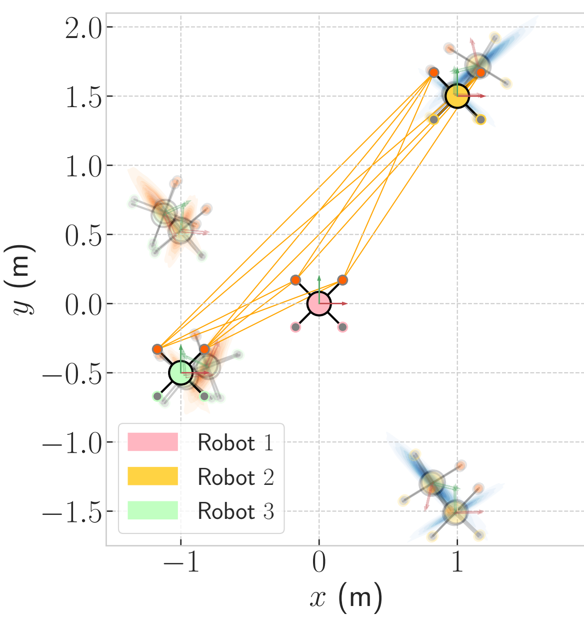
The flip ambiguities in modes 3 and 4 are reflections of the modes 1 and 2 about the axis joining the Tags τ_i and τ_j relative to $\mathcal{F}_1$, given by [42, Eq. (8)],

$$\mathbf{r}_1^{p1(\alpha+2)} = \frac{\begin{bmatrix} d_\alpha & -2a_\alpha b_\alpha \\ -2a_\alpha b_\alpha & -d_\alpha \end{bmatrix} \mathbf{r}_1^{p1(\alpha)} - 2c_\alpha \begin{bmatrix} a_\alpha \\ b_\alpha \end{bmatrix}}{a_\alpha^2 + b_\alpha^2},$$

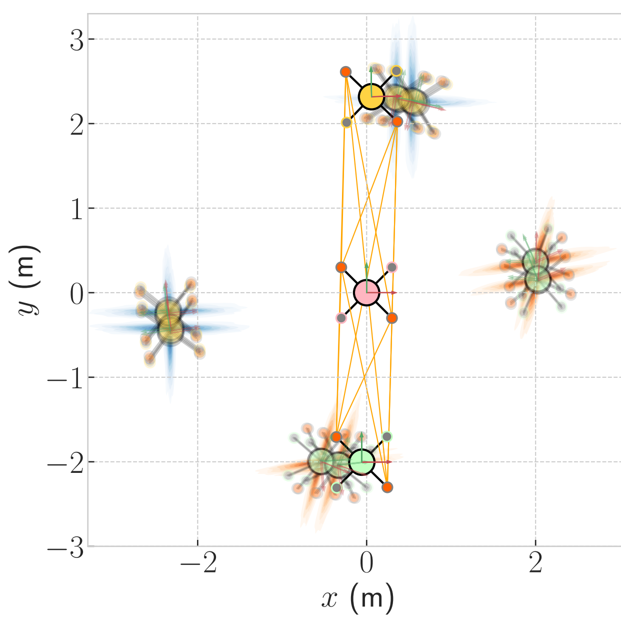
where, $d_\alpha = b_\alpha^2 - a_\alpha^2$, $c_\alpha = \operatorname{diag}(-a_\alpha,\, b_\alpha)\, \mathbf{r}_1^{\tau_i 1(\alpha)}$, and $\mathbf{r}_1^{\tau_j \tau_i(\alpha)} = [b_\alpha \; a_\alpha]^\mathsf{T}$, $\alpha = 1,\, 2$. As shown in Fig. 4.2, the respective attitudes in these modes have a heading of π relative to the attitudes in modes 1 and 2, given by,

$$\mathbf{C}_{1p}^{(\alpha+2)} = \mathbf{C}(\pi) \mathbf{C}_{1p}^{(\alpha)}, \quad \alpha = 1,\, 2. \tag{4.27}$$

By repeating this process, there will be 4 modes of $N-1$ relative poses between Robots 1 and p, for $p = 2, \ldots, N$. Therefore, the total number of combinations of modes are $M = (4)^{N-1}$, collectively denoted as $\mathbf{x}_{\text{geom}}^{(i)}$, $i = 1, \ldots, M$.



(a) Simulated UWB range data



(b) Experimental UWB range data

Figure 4.4: Comparison between the true pose and the ambiguous GI-LS pose estimates in a system of three robots, each having two tags. The opaque drones denote the true poses. The lighter shaded drones with their respective covariance plots are the pose estimates and their corresponding uncertainties.

4.6.2 Nonlinear Least-Squares Optimization

The geometric estimates $\mathbf{x}_{\text{geom}}^{(i)}, i = 1, \ldots, M$ are used to initialize a nonlinear least-squares algorithm [30] by solving

$$\hat{\mathbf{x}}_0 = \frac{1}{2} \arg\min_{\mathbf{x}} \|\mathbf{e}(\mathbf{x})\|^2 , \quad \text{where} \quad \mathbf{e}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) - \bar{\mathbf{y}}.$$

Here, instead of a single set of inter-tag range measurements, an average of $\gamma \geq 100$ range measurements, $\bar{\mathbf{y}}$, are used, which are collected when the robots are static. The averaging enhances the signal-to-noise ratio and improves estimation accuracy. For $i = 1, \ldots, M$, $\mathbf{x} \in SE(2)^{N-1}$ is iteratively updated using the $\oplus$ operator as,

$$\hat{\mathbf{x}}_t^{(i)} = \hat{\mathbf{x}}_{t-1}^{(i)} \oplus \left(\lambda \, \delta\mathbf{x}_{t-1}^{(i)}\right), \tag{4.28}$$

where λ is the step size, t is the iteration number, and $\hat{\mathbf{x}}_0^{(i)} = \mathbf{x}_{\text{geom}}^{(i)}$. The optimal step $\delta\mathbf{x}_{t-1}^{(i)}$ is given by

$$\delta\mathbf{x}_{t-1}^{(i)} = -\left(\mathbf{H}(\mathbf{x})^{\mathsf{T}}\mathbf{H}(\mathbf{x})\right)^{-1} \mathbf{H}(\mathbf{x})^{\mathsf{T}}\mathbf{e}(\mathbf{x})\Big|_{\hat{\mathbf{x}}_{t-1}^{(i)}}, \tag{4.29}$$

where $\mathbf{H}(\mathbf{x})$ is the measurement model Jacobian. The iterations are repeated until $||\delta\mathbf{x}_{t-1}^{(i)}||$ is small. In the measurement Jacobian, by taking the measurements between all the tags into account, the least-squares method produces a far more accurate estimate of the ambiguous relative poses than the geometric method. It even reduces the number of ambiguities since it is fed more inter-robot measurement information compared to the geometric method. However, the geometric method confines the initial guesses to a small and informed state space, essential for efficient convergence of the least-squares estimator.

The covariance of the least-squares estimate represents the uncertainties associated with estimating the state using the range measurements. Assuming that the average range measurements, $\bar{\mathbf{y}}$, are unbiased, using this matrix as the covariance of state estimates is a good starting point for any filter initialization. This covariance is given by [43]

$$\mathbf{P}_{\tau}^{(i)} = \Sigma^{(i)}(\mathbf{H}(\hat{\mathbf{x}}_{\tau}^{(i)})^{\mathsf{T}}\mathbf{H}(\hat{\mathbf{x}}_{\tau}^{(i)}))^{-1}, \ \Sigma^{(i)} = \frac{1}{L}\mathbf{e}(\hat{\mathbf{x}}_{\tau}^{(i)})^{\mathsf{T}}\mathbf{e}(\hat{\mathbf{x}}_{\tau}^{(i)}),$$

where $L = |\mathcal{E}| - (N-2)$, $\mathbf{e}(\hat{\mathbf{x}}_{\tau}^{(i)}) = \bar{\mathbf{y}} - \mathbf{g}(\hat{\mathbf{x}}_{\tau}^{(i)})$, $i = 1, \ldots, M$, and τ is the last iteration number. Finally, the relative poses between Robots 1 and p in the state $\mathbf{x}_{\tau}^{(i)}$ are transformed from $SE(2)$ to $SE(3)$ by augmenting them with zero quantities such that the relative roll, pitch and height are zero, which is a reasonable assumption for the start-up phase, where the robots are at ground level. These estimates and their covariances, denoted as $\{\hat{\mathbf{x}}_{\tau}^{(i)}, \hat{\mathbf{P}}_{\tau}^{(i)}\}_{i=1}^{M}$, are referred to as the *geometrically-initialized least-squares* (GI-LS) estimates.

This approach is validated in simulation and experiment, as shown in Fig. 4.4, using

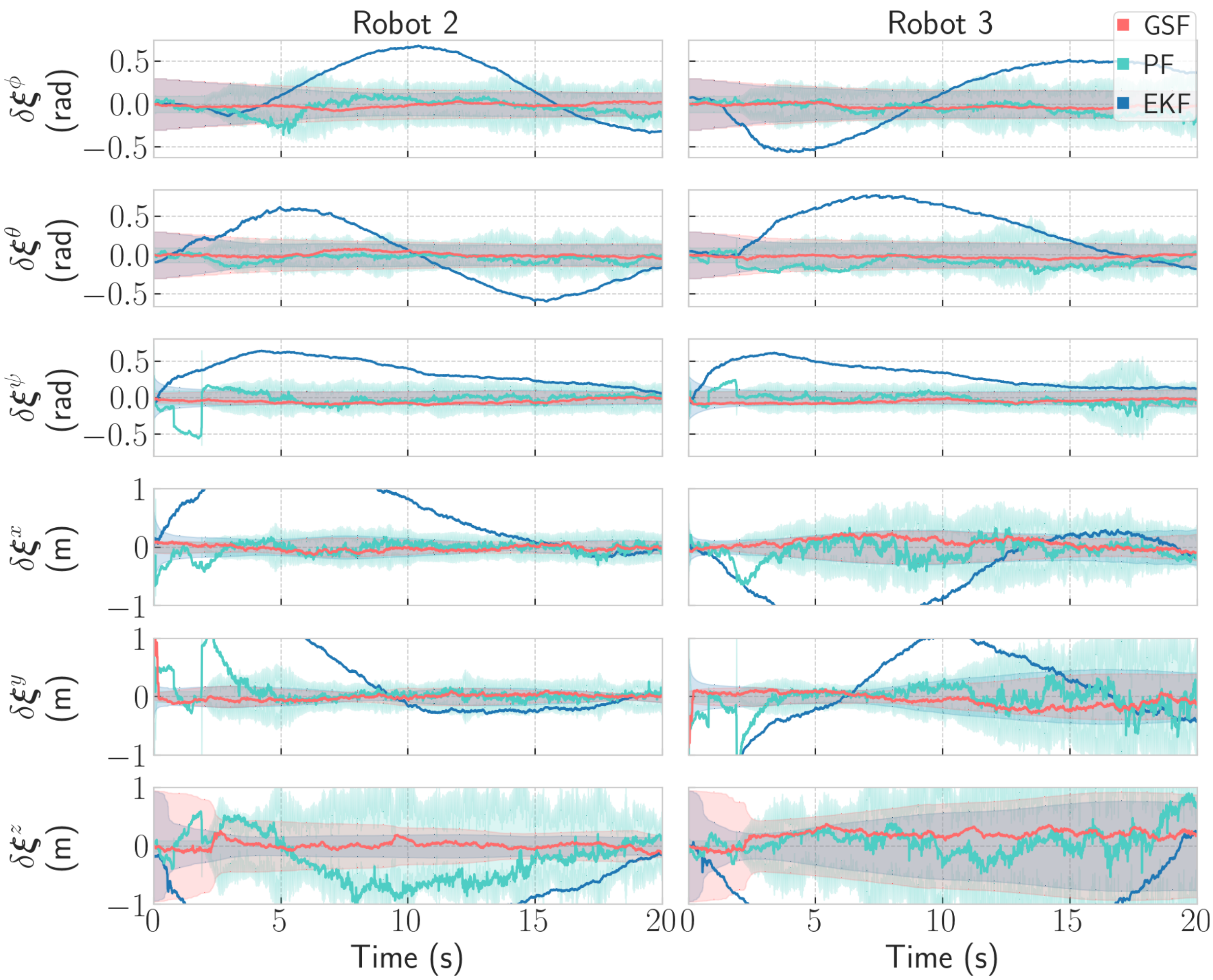


Figure 4.5: The performance of the EKF, GSF and PF on simulated data for two-tag Robots 2 and 3, with Robot 1 as reference robot. The GSF and PF are initialized with 8 GI-LS estimates and 1500 particles, respectively. The EKF is initialized in a wrong mode among the 8 GI-LS estimates. The shaded regions represent the $\pm 3\sigma$ bounds.

the problem setup in Fig. 4.1. Here, the lighter shaded drones depict the estimates with their covariances. Using 4 s of noisy range measurements at 50 Hz in simulation and 5 s at 90 Hz in experiment, both with a covariance $\mathbf{R} = 0.1^2\mathbf{1}\,\mathrm{m}^2$, the proposed method identifies all four ambiguities in $SE(2)$ for each robot. The estimates are accurate despite noise and disturbances. Furthermore, the estimates with lower covariances are more likely to be the "true" mode, given that the covariance indicates confidence. Note that, since the least squares method has more measurement information, in Fig. 4.4, it is able to reduce the number of ambiguities from 16 geometric estimates to 8 final estimates for the three-robot scenario.

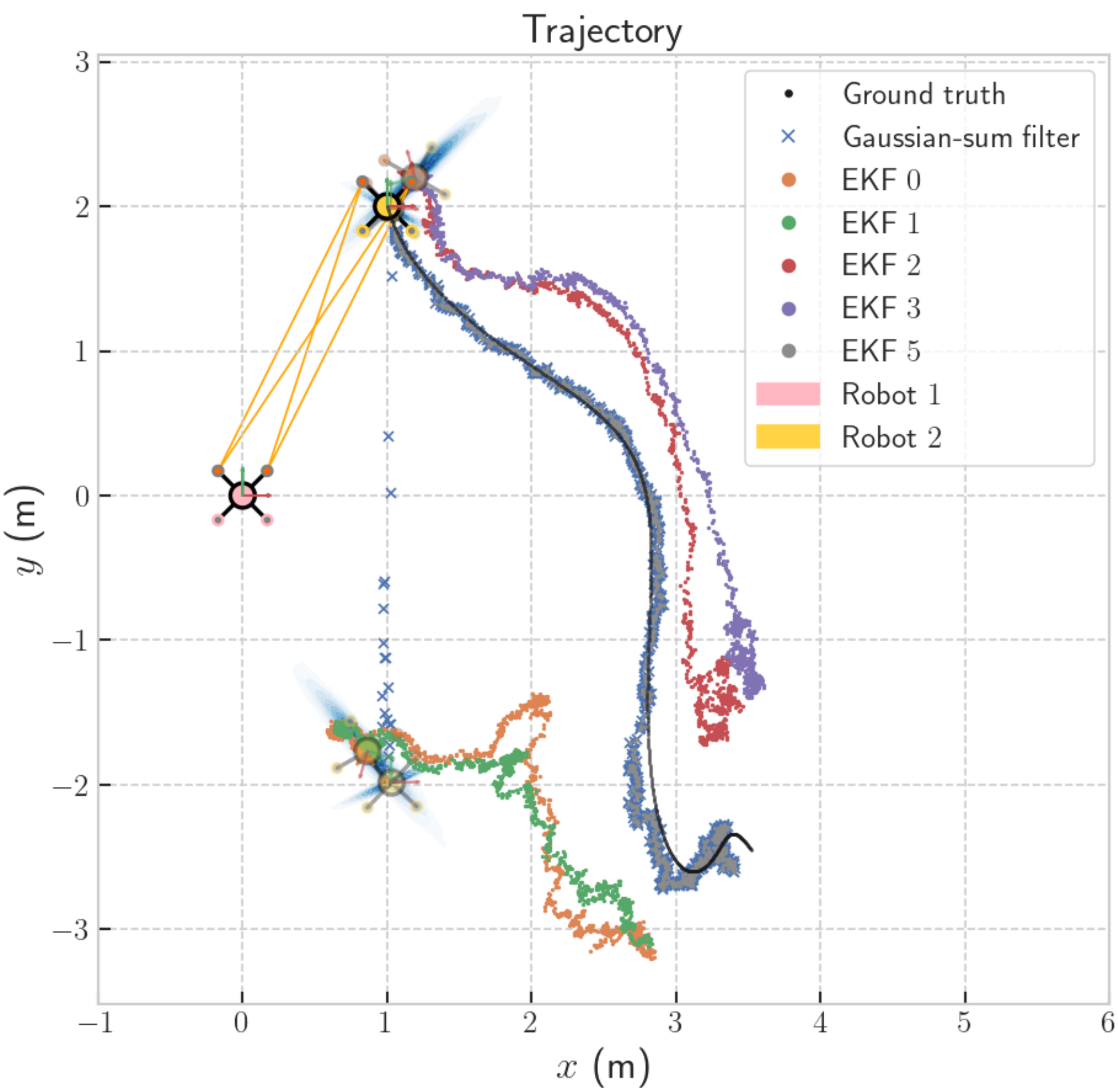


Figure 4.6: GSF trajectory estimation plot for a single run in simulation, shown in 2D. Only some modes of the GSF and only the relative position between Robot 1 and Robot 2 are shown for clarity. The ground truth starts at the location the quadcopters are plotted, and Robot 1 is the reference robot.

4.7 Simulations

The GSF with its proposed initialization features is compared with the PF and EKF in simulation. The setup is shown in Fig. 4.1, where the three robots have two tags each, and Robot 1 is the reference robot. The two tags are located at

$$\mathbf{r}_p^{\tau_i p} = \begin{bmatrix} 0.17 \\ 0.17 \\ 0 \end{bmatrix}, \mathbf{r}_p^{\tau_j p} = \begin{bmatrix} 0.17 \\ -0.17 \\ 0 \end{bmatrix},$$

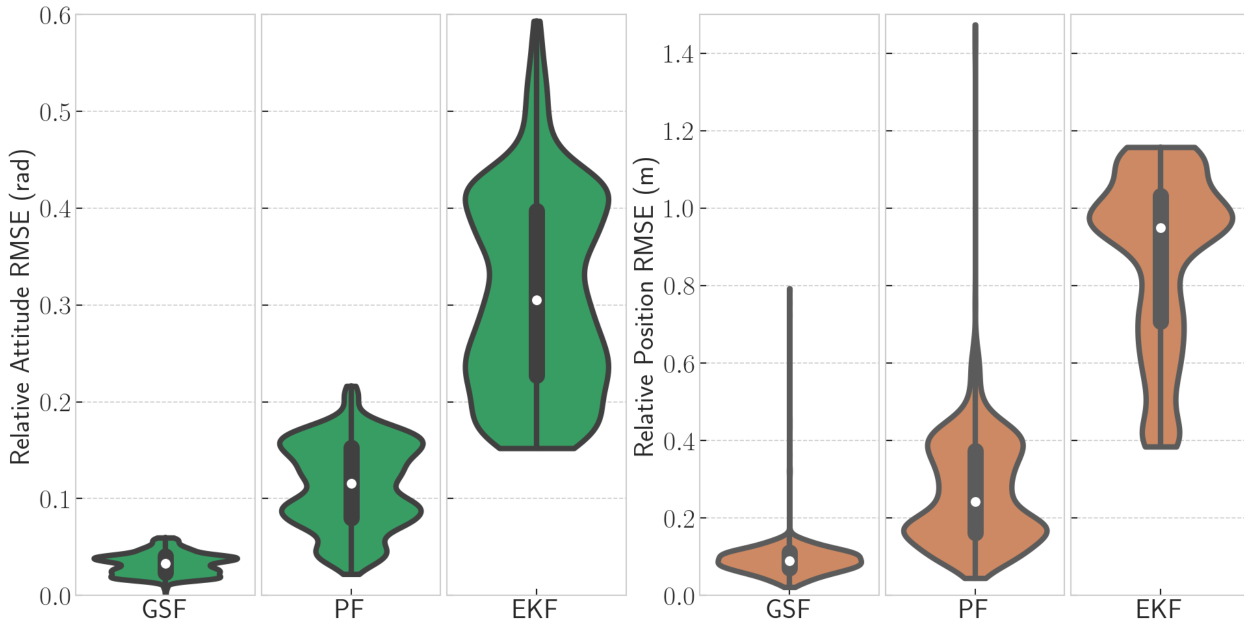


Figure 4.7: Violin and box plots showing the distribution of the 100-trial attitude and position RMSEs for simulation in $SE(3)$. The envelope shows the relative frequency of RMSE values. The white dot is the median, and the lower and upper bound of the black bar represent the first and third quartile of the data, respectively.

where i and j are the tag IDs, p is the robot ID, and the units are in meters. The robot velocities are inputs to the process model, and inter-tag range data at 50 Hz, with a covariance of $\mathbf{R} = 0.1^2\mathbf{1}\,\mathrm{m}^2$ are the measurements.

In Fig. 4.5, the pose-error plots for a single run of the GSF, PF, and EKF in simulation are shown. The GSF is initialized with 8 equally-weighted GI-LS estimates, $\{\hat{\mathbf{x}}_\tau^{(i)}, \hat{\mathbf{P}}_\tau^{(i)}\}_{i=1}^{8}$, the PF with 1500 particles around the ambiguities, and the EKF is initialized in a wrong mode among the 8 GI-LS estimates. Note that, the particles in the PF lie in the vicinity of the ambiguous poses evaluated using the nonlinear least-squares method. From the error plots alone, for this single run, despite the GSF having far fewer Gaussian components than the PF's particles, it is visibly more stable and accurate. The EKF diverges since it is initialized in a wrong mode. An EKF initialized in the correct mode is not shown, as it is not practical to know the correct mode in real-world scenarios. Additionally, Fig. 4.6 shows the GSF trajectory estimation plot in 2D for the same run. For clarity of reading the plot, only the relative position estimates between Robot 1 and Robot 2 and only some modes of the EKFs running inside the GSF are shown. The plot clearly shows that the GSF almost instantaneously converges to the "true" EKF as its highest-weighted mode, which is EKF 5.

The proposed GSF's performance is assessed over 100 Monte-Carlo trials with varied initial conditions and noise realizations on random trajectories. Its *root-mean-squared error*

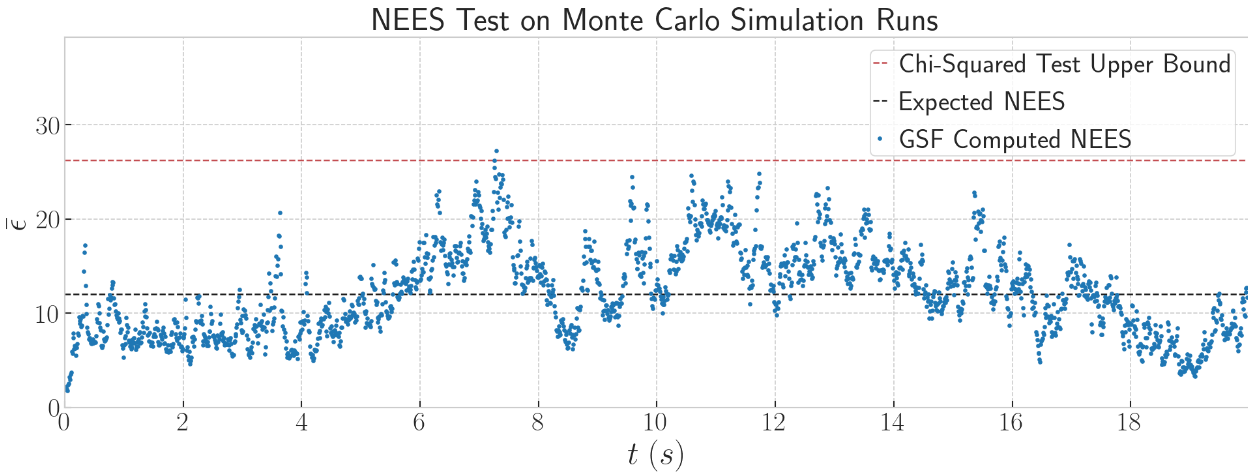


Figure 4.8: 100-trial NEES plot for the proposed GSF estimator in simulation.

(RMSE) is compared to 100 EKF and PF trials. The GSF is initialized with 8 GI-LS estimates, and the PF with 1500 particles, and the EKF is randomly initialized in one of the 8 modes. In Fig. 4.7, the GSF has a median attitude RMSE of 0.034 rad, which is 70.6% lower than the PF's 0.116 rad, and EKF's 0.305 rad. Similarly, the median position RMSE is 0.090 m for GSF, 0.242 m for PF and 0.949 m for EKF. Due to the proposed initialization method, the Gaussian components are highly informative while being far fewer than the particles in PF. This allows the GSF to converge to the true mode faster than the PF, making it more accurate, and computationally more efficient. The normalized estimation error squared (NEES) test in Fig. 4.8 confirms GSF's consistency within a 99% confidence interval.

4.8 Experimental Results

The filters are tested on three Uvify IFO-S quadcopters to validate their performances in experiment. The setup of three robots is depicted in Fig. 4.1, with each robot having two tags, and Robot 1 is the reference robot. The two tags in all the robots are located at

$$\mathbf{r}_p^{\tau_i p} = \begin{bmatrix} 0.16 \\ -0.17 \\ -0.05 \end{bmatrix}, \mathbf{r}_p^{\tau_j p} = \begin{bmatrix} -0.17 \\ 0.16 \\ -0.05 \end{bmatrix},$$

where i and j are the tag IDs, p is the robot ID, and the units are in meters. Each robot has an onboard IMU and an Intel RealSense D435i stereo camera set. These sensors provide the translational velocity estimates through VIO using the ROS package Vins-Fusion [44] at 30 Hz, and the angular velocity readings are taken from the gyroscope at 200 Hz. The velocity estimates from VIO only serve as interoceptive measurements to validate the proposed

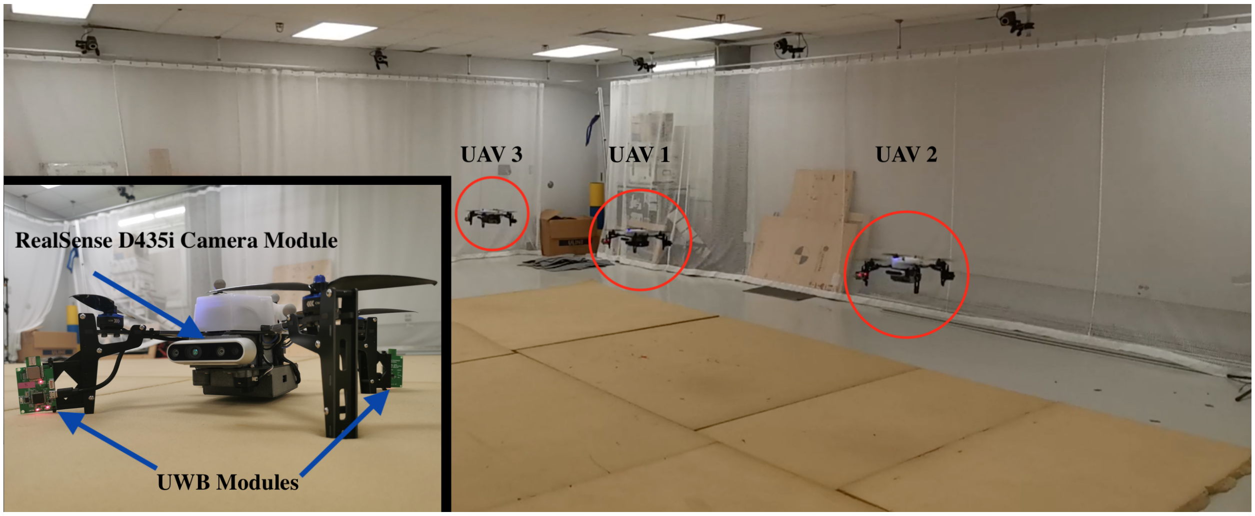


Figure 4.9: Experimental setup showing the three robots. Two UWB modules or tags and an Intel RealSense D435i camera are mounted on each robot.

estimation approach. Any other interoceptive measurements can be used in place of VIO as well. Pose data from the Vicon motion-capture system serve as ground truth. The robots follow a random 3D trajectory in a $6 \times 6 \times 3\,\mathrm{m}^3$ space as shown in Fig. 4.9.

The UWB range measurements are provided to all the estimators at 90 Hz, which are corrected for uncertainties and biases using the works of [45]. The GSF is initialized with 8 Gaussian GI-LS estimates, the PF with 1500 particles, and the EKF is initialized in a wrong mode among the 8 GI-LS estimates. In this instance, the particles in the PF lie in the vicinity of the ambiguous poses evaluated using the nonlinear least-squares method, similar to the simulation setup. In the filters, any measurement that does not pass the *normalized innovation squared* NIS test is rejected. Fig. 4.10 displays the pose-error plots of the filters in experiment. The GSF and PF perform similarly, but the EKF diverges as expected. Initially, the GSF has large error spikes, but it soon stabilizes once it isolates the "true" mode. In Python 3.8, the GSF estimates the states at an average rate of 40 Hz, and the PF does the same at 3.5 Hz, making the GSF many folds faster and strongly eligible for online implementation.

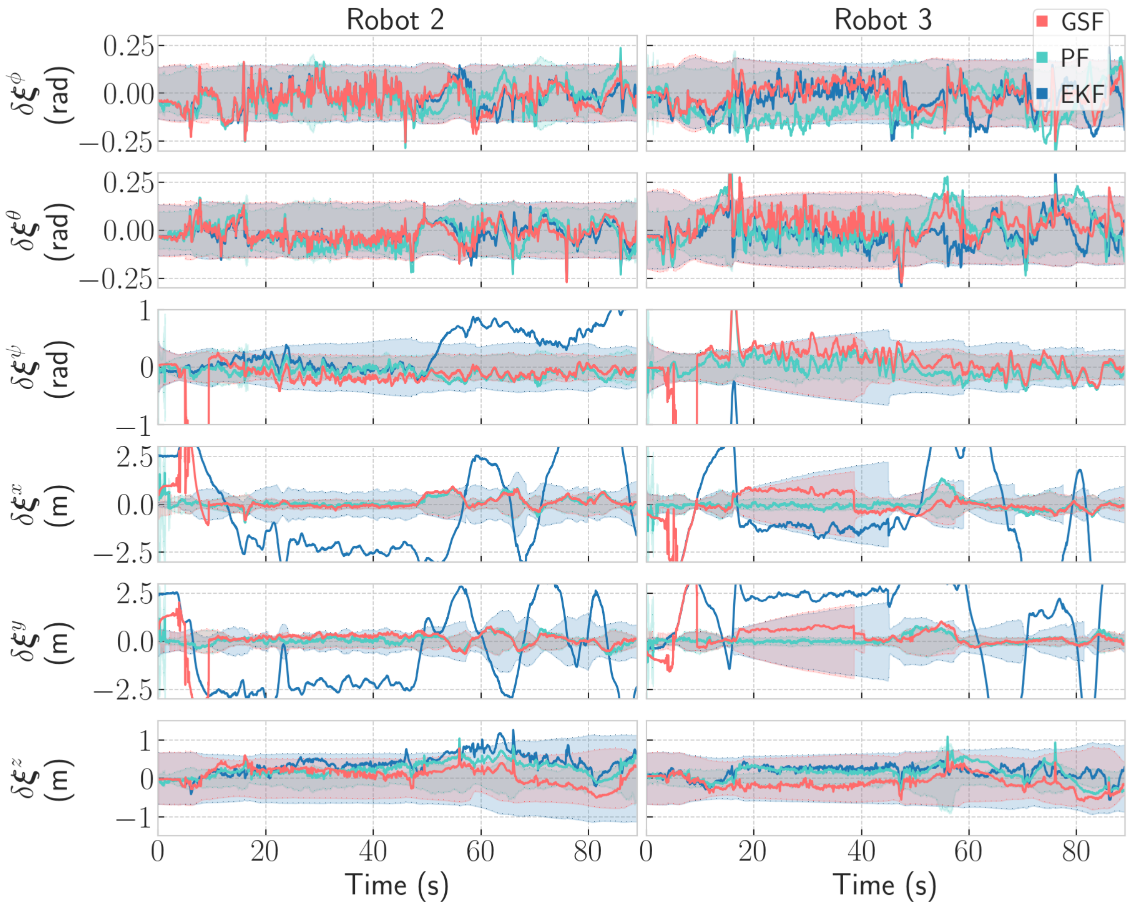


Figure 4.10: The performance of the EKF, GSF and PF on experimental data for two-tag Robots 2 and 3, with Robot 1 as reference robot. The GSF and PF are initialized with 8 GI-LS estimates and 1500 particles, respectively. The EKF is initialized in a wrong mode among the 8 GI-LS estimates. The shaded regions represent the $\pm 3\sigma$ bounds.

4.9 Conclusion

Multi-robot systems with non-stationary range sensors suffer from ambiguous poses due to observability issues. This chapter provides a complete and efficient 3D relative pose estimation solution for these systems where UWB ranging tags are the only exteroceptive sensors. In simulations and experiments, the proposed estimator in the form of a Gaussian-sum filter is shown to be accurate and computationally efficient. The GSF is consistent within a 99% confidence interval, and it performs comparatively faster than the particle filter. The results establish that a well-modelled GSF should be the default tool for range-based 3D relative pose estimation in multi-robot systems. Looking ahead, for larger systems, decentralizing with multiple reference robots can optimize the number of Gaussian components in the GSF, thus reducing computational strain while retaining accuracy.

Chapter 5

Optimal Robot Formations: Balancing Range-Based Observability and User-Defined Configurations

5.1 Summary

This chapter introduces a set of customizable and novel cost functions that enable the user to easily specify desirable robot formations, such as a "high-coverage" infrastructure-inspection formation, while maintaining high relative pose estimation accuracy. The overall cost function balances the need for the robots to be close together for good ranging-based relative localization accuracy and the need for the robots to achieve specific tasks, such as minimizing the time taken to inspect a given area. The formations found by minimizing the aggregated cost function are evaluated in a coverage path planning task in simulation and experiment, where the robots localize themselves and unknown landmarks using a simultaneous localization and mapping algorithm based on the extended Kalman filter. Compared to an optimal formation that maximizes ranging-based relative localization accuracy, these formations significantly reduce the time to cover a given area with minimal impact on relative pose estimation accuracy.

5.2 Introduction

With the goal of adopting anchor-free localization, the two-tag multi-robot setup, proven to be locally observable [11, 46], is the chosen setup in this chapter. This is because this setup has the smallest number of tags required to reliably estimate the relative pose between two robots by fusing range measurements from the two tags in each robot with inertial measurement

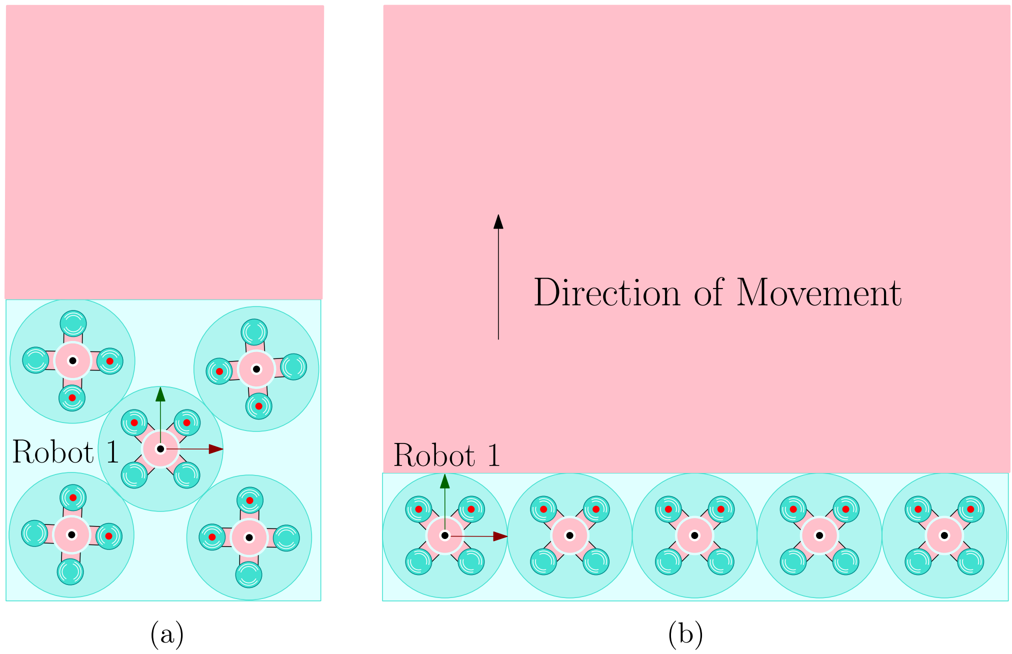


Figure 5.1: Comparing the coverage span of two formations. The circles represent the camera's field-of-view of each robot, and the red dots denote the location of the ranging tags. (a) The robots are clustered together to ensure high relative pose estimation accuracy, as shown in [21]. (b) The robots are spread apart in a horizontal line to cover a larger area, which minimizes coverage time.

unit (IMU) data using an extended Kalman filter (EKF) [11, 46]. However, with any range-measurement based setup, relative pose estimation accuracy is highly dependent on the robots' formation. In some formations, there are ambiguities, which can cause the estimator to diverge [21, 47]. Adding more than two tags may reduce the number of ambiguities in certain formations, but still does not eliminate them all. The presence of ambiguities causes the estimator to diverge in certain formations, such as when all the robots are in a straight line, as shown in Fig. 5.1b [21, 47].

To address this issue, [21] suggests keeping the team of robots in formations where they are close and clustered together, as shown in Fig. 5.1a, which theoretically maximizes the relative pose estimation accuracy for two-tagged robots. However, these clustered formations are not ideal for applications such as infrastructure inspection or surveillance, where maximizing coverage is beneficial. An example of robot clustering resulting in reduced coverage is shown in Fig. 5.1.

This chapter addresses the contrasting objectives of determining multi-robot formations that both (1) maximize coverage and (2) ensure close proximity between robots for good relative localization accuracy. Other multi-robot path planning mechanisms have focused on distributing the robots into different sectors in a large area, where each robot individually covers its sector to minimize overall coverage time [48–52]. The robots generally localize themselves using the Global Positioning System (GPS). However, with a UWB ranging-based approach, the robots cannot be distributed into sectors since they must be in proximity to each other to achieve high relative pose estimation accuracy, as highlighted in [21].

The key contribution of this chapter is a cost function that brings the robots to any desirable formation, such as a "high-coverage" straight-line formation, while simultaneously maintaining high relative localization accuracy. This cost function has a component that provides the user with the ability to choose the direction and distance between any two adjacent robots. This feature enables the user to realize different formations for various applications, such as bridge inspection, as demonstrated in Section 5.6.4. User-defined formations can be achieved using acceleration inputs [53, 54], but the proposed component within the cost function is easily customizable and integrable with the formulation of [21]. Another component of this cost function allows the user to allocate a certain amount of overlap between adjacent robots' camera views, which is good for image-stitching and in improving mapping accuracy, as mentioned in [55]. Observability and collision avoidance terms are also incorporated into the cost function.

The "high-coverage" formations generated by minimizing the proposed cost function are tested in a planning task in simulation and experiment, where the robots localize themselves and unknown anchors using a simultaneous localization and mapping (SLAM) algorithm based on the EKF. In this chapter, the EKF is chosen rather than the GSF since the position of the robots relative to the global frame is assumed to be known, which allows us to avoid the ambiguity-related issues addressed in Chapter 4 during the initialization of the SLAM algorithm. Compared to the current state-of-the-art, the proposed formations significantly reduce coverage time with minimal impact on localization accuracy.

The remainder of this chapter is organized as follows. The notation and preliminaries are defined in Section 5.3. The problem is motivated in Section 5.5. The proposed cost functions are in Section 5.6. The application of the cost function in simulations and experiments is in Section 5.7.

5.3 Notation and Preliminaries

The problem setup is a multi-robot system, where each robot is equipped with two ranging tags, as shown in Fig. 4.1. Therefore, all the notational preliminaries are same as in 4.3 of Chapter 4. The only difference between the setup in this chapter and the one in Chapter 4

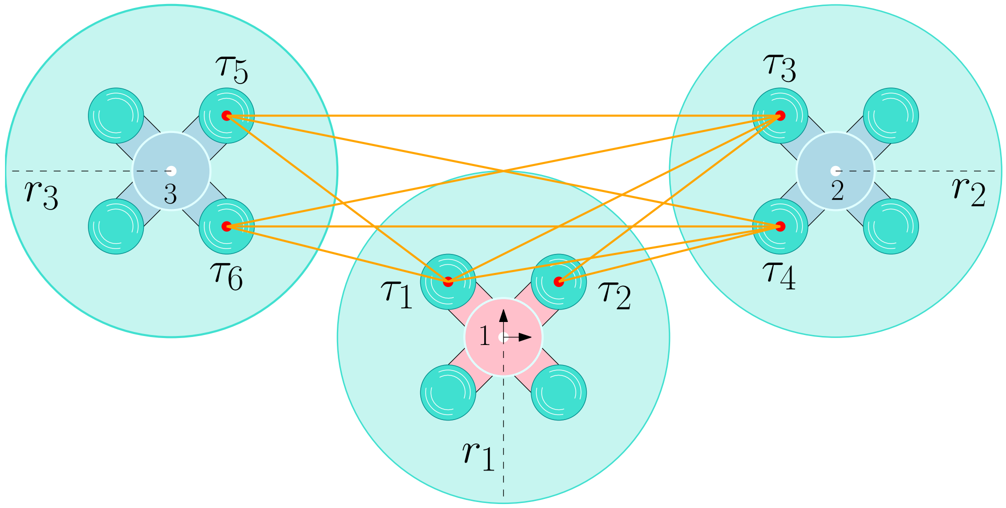


Figure 5.2: Problem setup for a two-tag multi-robot system, where Robot p is equipped with tags τ_i and τ_j, and a camera with a circular view of radius r_p in the up or down direction. Without loss of generality, the pink robot, defined as Robot 1, is considered to be the reference robot.

is that the robots are assumed to be equipped with a downward or upward-facing camera that has a circular field-of-view with a known radius, r_p here. The set of radii is denoted as $\mathcal{R} = \{r_1, \ldots, r_N\}$. The new setup is shown in Fig. 5.2. Furthermore, in this chapter the state of the system is defined in 2D and therefore the relative pose between Robots p and q is $\mathbf{T}_{pq} \in SE(2)$.

5.4 Optimization

In this chapter, locally optimal formations are found by minimizing cost functions of $\mathbf{x} \in SE(2)^{N-1}$, $J(\mathbf{x})$ in 2D. All such cost functions are minimized using a momentum-based gradient descent algorithm. This approach is preferred over a standard gradient descent method as it allows for faster convergence to a global or local minimum [56]. The state is updated from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$ using a left or right perturbation $\delta\mathbf{x}_t \in \mathbb{R}^{3\times(N-1)}$ as

$$\delta\mathbf{x}_t = -\Big(\alpha\nabla J(\mathbf{x}_t) + \beta\delta\mathbf{x}_{t-1}\Big)^\mathsf{T}, \tag{5.1}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t \oplus \delta\mathbf{x}_t, \tag{5.2}$$

where $\nabla J(\mathbf{x}_t)$ is the gradient of the cost function numerically computed using finite difference [57], α is the learning rate, and β is the momentum parameter. Throughout the chapter, the parameters $\alpha = 0.001$ and $\beta = 0.9$ are used. The optimization is terminated when $||\delta\mathbf{x}_t|| < 10^{-4}$.

5.5 Motivation

The goal of this chapter is to find multi-robot formations that minimize the coverage time of a given space, as shown in Fig. 5.1. The challenge is to balance this objective with the necessity for accurate relative pose estimation using range measurements. To find an appropriate multi-robot formation with good ranging-based relative pose estimation accuracy, [21] proposes the minimization of

$$J_{\text{opt}}(\mathbf{x}) = J_{\text{est}}(\mathbf{x}) + J_{\text{col}}(\mathbf{x}), \tag{5.3}$$

where $J_{\text{est}}(\mathbf{x})$ quantifies the relative pose estimation error and uncertainty using the Cramér-Rao lower bound [21, 58, 59], and $J_{\text{col}}(\mathbf{x})$ is the collision avoidance term. Note that,

$$J_{\text{est}}(\mathbf{x}) = -\ln\det\Big(\mathbf{H}(\mathbf{x})^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}(\mathbf{x})\Big), \tag{5.4}$$

where $\mathbf{H}(\mathbf{x})$ is the Jacobian of the measurement model given in (4.17) of Chapter 4, and $\mathbf{R}$ is the measurement covariance. The collision avoidance term is defined as [60]

$$J_{\text{col}}^{mn}(\mathbf{x}) = \left(\min\left\{0, \frac{||\mathbf{r}_1^{mn}||^2 - A^2}{||\mathbf{r}_1^{mn}||^2 - d^2}\right\}\right)^2, \tag{5.5}$$

$$J_{\text{col}}(\mathbf{x}) = \sum_{\substack{m,n\in\mathcal{P},\\ m\neq n}} J_{\text{col}}^{mn}(\mathbf{x}), \tag{5.6}$$

where A is the activation radius and d is the collision avoidance radius, set to $A = 0.9\,\text{m}$, and $d = 0.5\,\text{m}$ throughout this chapter. This collision avoidance term is reused by recent work [21] on multi-robot formation problems and therefore is considered well-suited for this work. The multi-robot formations deduced by minimizing (5.3) generally have the robots clustered together, where the robots have low area coverage as shown in Fig. 5.1a. In fact, [21] shows that a straight-line formation with high coverage, as shown in Fig. 5.1b, unacceptably increases the relative pose estimation error. However, in theory, there are many "high-coverage" formations, possibly near the local minima of $J_{\text{est}}(\mathbf{x})$, where the ranging-based relative pose estimation accuracy is high. These formations are achievable by minimizing a different cost function, as presented in Section 5.6.

5.6 Proposed Cost Functions

Two novel cost functions are proposed in this section, which are added to (5.3). The first one allows any desirable multi-robot formation acquisition suitable for the task, and the second one ensures a certain degree of overlap between adjacent robots' camera views. The final cost function also takes relative localization accuracy and collision avoidance into account. Minimizing the final cost function helps the robots adopt "high coverage" formations, such as a "near" straight-line formation while ensuring consistently high accuracy in relative localization. The problem is approached in 2D since most robots, such as ground vehicles or quadcopters, only have heading as a rotational degree of freedom for planning purposes.

5.6.1 Adjacent Robot Formation Cost Function

Let N robots be initially positioned at random locations. The goal of this section is to allocate the robots into any desired formation, with all formations being relative to Robot 1, the reference robot. The idea is to minimize the error between the actual and desired position vector between any two robots, which results in the cost function

$$J_{\text{adj}}^{mn}(\mathbf{x}) = \left|\left|\mathbf{r}_1^{mn} - \sum_{k=n}^{m-1}(r_{k+1} + r_k)\mathbf{n}_1^{(k)}\right|\right|^2, \tag{5.7}$$

$$J_{\text{adj}}(\mathbf{x}) = \sum_{\substack{n,m\in\mathcal{P},\\ n<m}} J_{\text{adj}}^{mn}(\mathbf{x}), \tag{5.8}$$

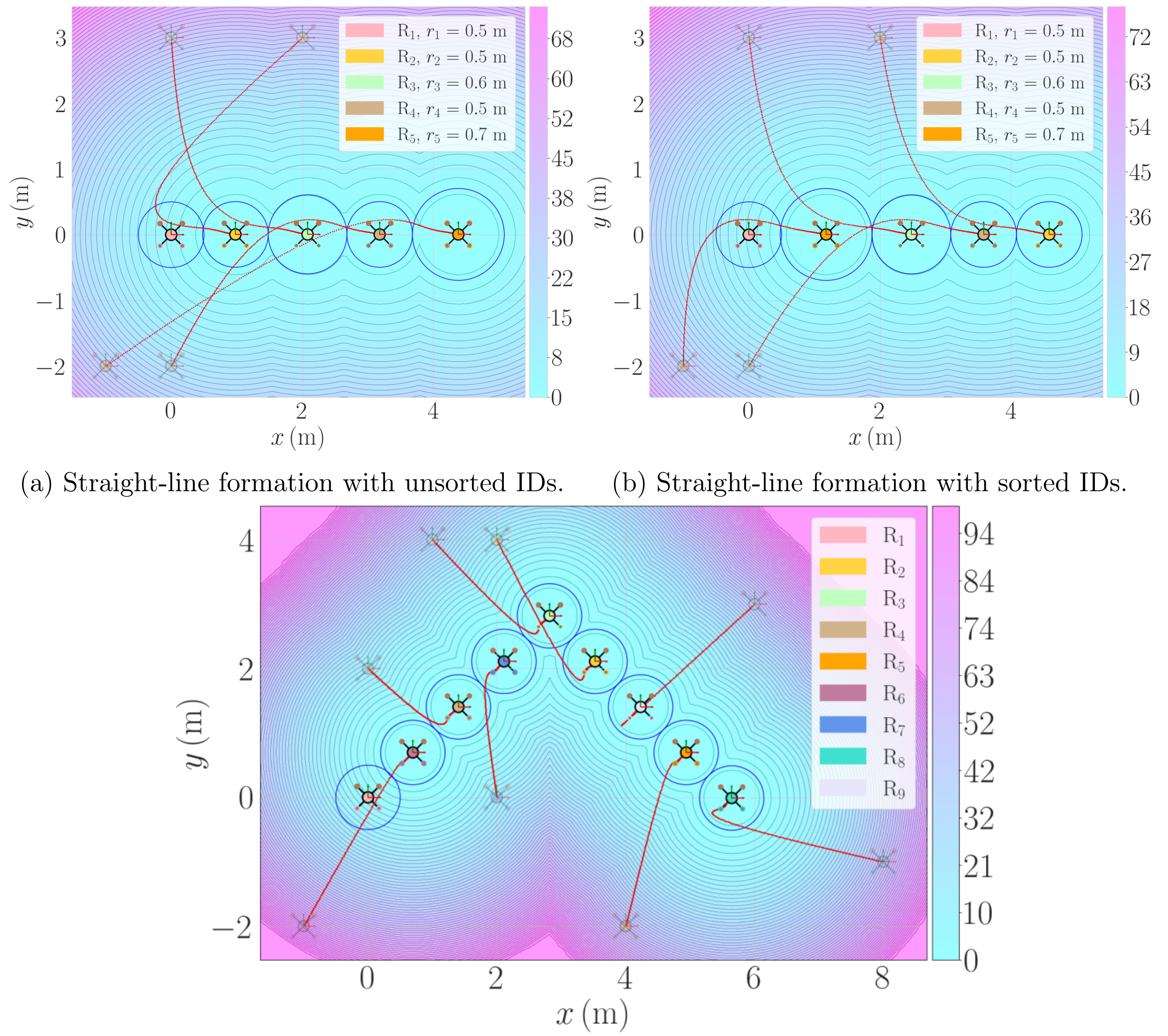
where r_k and $\mathbf{n}_1^{(k)}$ are user-defined parameters that determine the radial distance and direction between adjacent robots, respectively. $\mathbf{n}_1^{(k)}$ is the desired unit vector associated with the position of Robot $k+1$ relative to its adjacent robot, Robot k, resolved in $\mathcal{F}_1$. All the desired unit vectors, starting with the one from the reference robot, Robot 1, can be written compactly as,

$$\mathbf{n}_1 = \begin{bmatrix} \mathbf{n}_1^{(1)\mathsf{T}} & \cdots & \mathbf{n}_1^{(N-1)\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{3\times(N-1)}. \tag{5.9}$$

The desired position vector of Robot m relative to Robot n, resolved in $\mathcal{F}_1$ is found using the summation term in (5.7).

This cost function places the robots adjacent to each other in ascending order of their IDs without determining the shortest path the robots should take to form the desired formation, as shown in Fig. 5.3a. For conciseness, Robot p is referred to as R_p in plot legends of all the figures. However, this is not ideal, and Algorithm 1 sorts the robot IDs so that the robots take the shortest path possible to the user-defined formation. This algorithm finds the permutation of the robot IDs that minimizes the overall distance traveled by the robots to reach the desired formation using the Hungarian matching algorithm [61], and is faster than a brute-force approach.

The sorted set of robot IDs and radii are denoted $\mathcal{P}_s = \{s_1, \dots, s_N\}$ and $\mathcal{R}_s = \{r_{s_1}, \dots, r_{s_N}\}$, respectively. For conciseness, $\mathbf{r}_{s_n}^{s_n s_m}$ is denoted as $\bar{\mathbf{r}}_n^{nm}$, the attitude between robots s_n and s_m is denoted as $\bar{\mathbf{C}}_{nm}$, and the radius of Robot s_n is denoted as $\bar{r}_n$. For this



(a) Straight-line formation with unsorted IDs.

(b) Straight-line formation with sorted IDs.

(c) V-shaped formation with sorted IDs.

Figure 5.3: Formations obtained by minimizing $J_{\text{adj}}(\mathbf{x})$. The contours represent the heatmap of the cost function $J_{\text{adj}}(\mathbf{x})$, by varying $\mathbf{r}_n^{mn}$ between all the robots.

sorted set of robot IDs, (5.7) becomes

$$J_{\text{adj}}^{mn}(\mathbf{x}) = \left|\left|\bar{\mathbf{r}}_1^{mn} - \sum_{k=n}^{m-1}(\bar{r}_{k+1} + \bar{r}_k)\mathbf{n}_1^{(k)}\right|\right|^2. \tag{5.10}$$

Note that, $\mathbf{n}_1$ denotes the desired unit vectors between adjacent robots starting from the reference robot, Robot 1, and therefore is not affected by the sorting of the IDs.

Fig. 5.3b depicts a straight-line formation acquisition by minimizing $J_{\text{adj}}(\mathbf{x})$ with sorted robot IDs. With sorted IDs, the robots reach a straight-line formation by traveling a shorter overall distance compared to the one with unsorted IDs, shown in Fig. 5.3a. In both cases

Algorithm 1 Sort Robot IDs by Distance To Travel

Input: $\mathbf{x}$, $\mathcal{P}$, $\mathcal{R}$, $\mathbf{n}_1$.
Output: $\mathcal{P}_s$, $\mathcal{R}_s$.

1: Let $\mathbf{r}_1 \triangleq \begin{bmatrix} \mathbf{r}_1^{21} & \cdots & \mathbf{r}_1^{N1} \end{bmatrix}^\mathsf{T}$,
2: and $\mathbf{p} = \begin{bmatrix} 2 & \cdots & N \end{bmatrix}^\mathsf{T}$, where $2, \ldots, N \in \mathcal{P} \setminus \{1\}$.
3: $d_{\text{avg}} \leftarrow \frac{2}{N} \sum_{n=1}^{N} r_n$.
4: Compute the approximate target locations in the goal formation,
5: $\mathbf{r}_1^* \leftarrow \begin{bmatrix} \sum_{k=1}^{2} d_{\text{avg}} \mathbf{n}_1^{(k)\mathsf{T}} & \cdots & \sum_{k=1}^{N} d_{\text{avg}} \mathbf{n}_1^{(k)\mathsf{T}} \end{bmatrix}^\mathsf{T}$
6: $\quad \triangleq \begin{bmatrix} \mathbf{r}_1^{d_2 d_1 \mathsf{T}} & \cdots & \mathbf{r}_1^{d_N d_1 \mathsf{T}} \end{bmatrix}^\mathsf{T}$.
7: Create a matrix cost function based on the distance traveled by each robot to the goal formation,
8: $\mathbf{C}(i,j) \leftarrow ||\mathbf{r}_1^*(i) - \mathbf{r}_1(j)||^2$ for $i, j \in \{1, \ldots, N-1\}$.
9: Let $\mathbf{P}$ be a permutation matrix, and $\text{tr}(\cdot)$ is the trace operator. Find the permutation matrix that minimizes the overall distance traveled by the robots using the Hungarian matching algorithm [61], $\quad \mathbf{P}^* \leftarrow \min_{\mathbf{P}} \text{tr}(\mathbf{CP})$.
10: $\mathcal{P}_s \leftarrow \{1\} \cup \{i^{\text{th}}$ element of $\mathbf{P}^*\mathbf{p}\} \triangleq \{s_1, \ldots, s_N\}$.
11: $\mathcal{R}_s \leftarrow \{r_{s_n}\}$.

$\mathbf{n}_1^{(k)} = [1 \quad 0]^\mathsf{T}$, $k = 1, \ldots, N-1$.

Another instance of the implementation of this cost function is shown in Fig. 5.3c, where the robots are in a V-shaped formation. The parameters used for this example are $\mathbf{n}_1^{(k)} = [1 \quad 1]^\mathsf{T}$, $k = 1, \ldots, 4$, $\mathbf{n}_1^{(k)} = [1 \ -1]^\mathsf{T}$, $k = 5, \ldots, 8$, and radii $\bar{r}_k = 0.5\,\text{m}$.

In the rest of this chapter, unless $\mathbf{n}_1$ is stated, the sorted set of IDs is computed using $\mathbf{n}_1^{(k)} = [1 \quad 0]^\mathsf{T}, k = 1, \ldots, N-1$, to maximize coverage span in the x-direction.

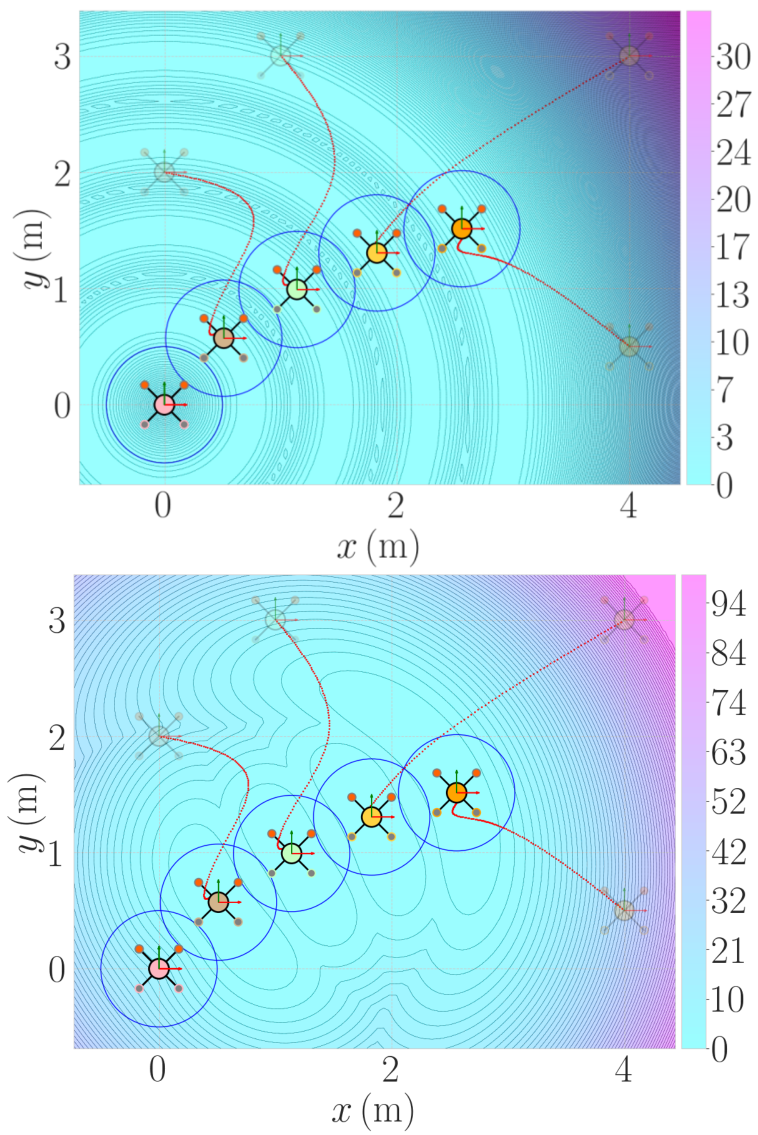


Figure 5.4: The formation with adjacent camera overlap after minimizing J_{overlap}, with $\lambda = 0.25$. The upper plot shows the effects of the heatmap of $J_{\text{overlap}}(\mathbf{x})$ from the perspective of only Robot 1, and the lower plot shows the effects of the heatmap from the perspective of all the robots. Only position $\mathbf{r}_n^{mn}$ is varied between all the robots to generate the heatmaps.

5.6.2 Camera Overlap Cost Function

To simultaneously enable overlap of the camera views of adjacent robots, and to ensure that no more than two adjacent camera views overlap, which in turn helps in maximizing coverage, minimizing the cost function

$$J^{mn}_{\text{overlap}}(\mathbf{x}) = \left|\left|\bar{\mathbf{r}}^{mn}_1 - (1-\lambda)\left(2\sum_{k=n}^{m}\bar{r}_k - \bar{r}_n - \bar{r}_m\right)\bar{\mathbf{n}}^{mn}_1\right|\right|^2, \tag{5.11}$$

$$J_{\text{overlap}}(\mathbf{x}) = \sum_{\substack{s_n, s_m \in \mathcal{P}_s,\\ n<m}} J^{mn}_{\text{overlap}}(\mathbf{x}) \tag{5.12}$$

is proposed, where $\lambda \in [0, 1]$ represents the percentage of the radial distance between the robots that overlap. The direction vector $\bar{\mathbf{n}}^{mn}_1$ is the unit vector pointing from Robot s_n to Robot s_m in the body frame of Robot 1 and is given by

$$\bar{\mathbf{n}}^{mn}_1 = \frac{\bar{\mathbf{r}}^{mn}_1}{||\bar{\mathbf{r}}^{mn}_1||}. \tag{5.13}$$

An example formation with $\lambda = 0.25$ is shown in Fig. 5.4. From the contours in the left plot, note that the cost function is designed to create valleys at a distance equivalent to the summation term in (5.11) scaled by $(1-\lambda)$ around Robot 1, and similar valleys exist around all other robots. The intersection of these valleys causes the robots to overlap their camera views with adjacent robots. The advantage of this cost function is that, regardless of where the robots are initially located, every robot will end up overlapping its camera's field-of-view with adjacent robots. Therefore, this cost function is not limited to any specific formation.

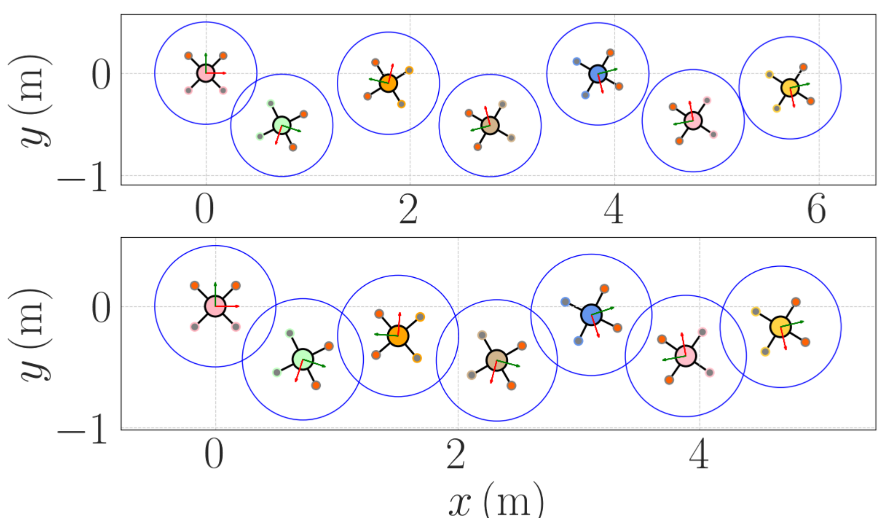


Figure 5.5: Final formation acquisition with coverage in the x-direction without (top) and with (bottom) the camera overlap cost function, $J_{\text{overlap}}(\mathbf{x})$.

5.6.3 Overall Cost Function

By encoding user-defined requirements for certain formations, such as a straight-line formation, and radii overlap mathematically, the proposed cost functions can be added to (5.3) to achieve a comprehensive solution for formations that accommodate a variety of factors. These factors include the need for high coverage, the necessity for accurate relative pose estimation, and the requirement for camera overlap, among others. The overall cost function is given by,

$$J_{\text{cov}}(\mathbf{x}) = J_{\text{adj}}(\mathbf{x}) + J_{\text{overlap}}(\mathbf{x}) + J_{\text{est}}(\mathbf{x}) + J_{\text{col}}(\mathbf{x}). \tag{5.14}$$

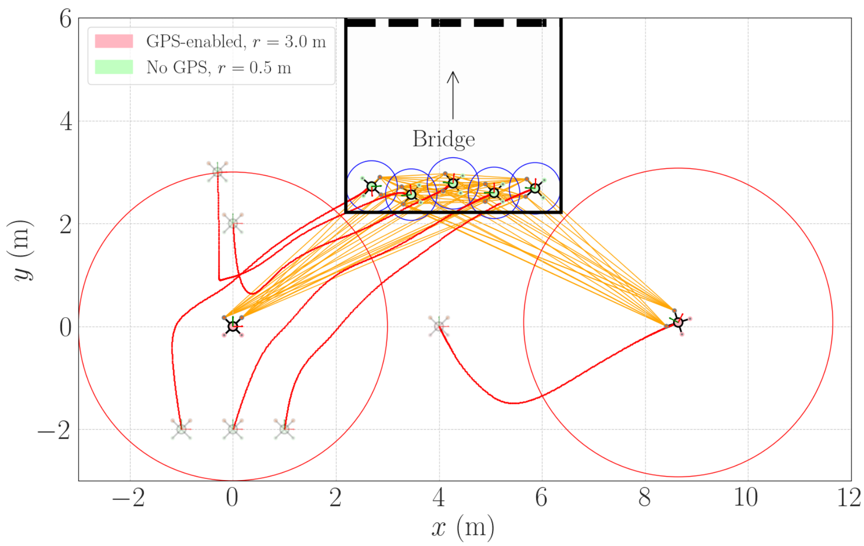
Fig. 5.5 depicts an example formation with coverage in the x-direction by minimizing $J_{\text{cov}}(\mathbf{x})$. The plots highlight the importance of $J_{\text{overlap}}(\mathbf{x})$ in preventing the robots from non-uniformly spreading apart due to the other cost function components, notably $J_{\text{adj}}(\mathbf{x})$. The cost $J_{\text{cov}}(\mathbf{x})$ serves to design suitable formations for planning problems and therefore the optimization is done offline before the start of the mission. These formation results can then be stored in the memory of the robots and used for online planning. Handling online planning initiatives like real-time non-line-of-sight issues between tags or the need for formation changes in the presence of obstacles is beyond the scope of this chapter.

5.6.4 Bridge Inspection Example

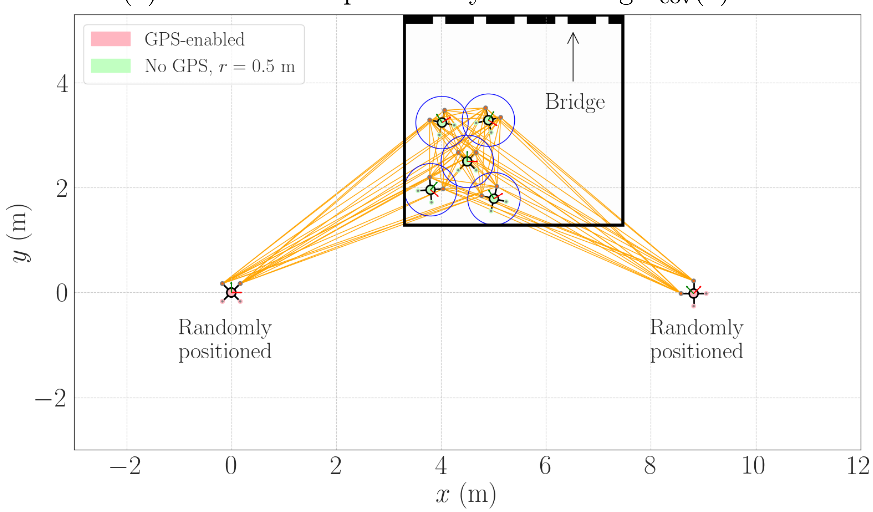
The usefulness of $J_{\text{cov}}(\mathbf{x})$ is shown in the bridge inspection application in Fig. 5.6a. Here, 5 quadcopters with top-facing cameras inspect the underside of a bridge with no access to GPS, and two other GPS-enabled quadcopters are placed at an arbitrary angle to the inspection robots to get good localization accuracy. The desired formation is a straight-line formation of the inspection robots with some camera overlap, while ensuring that the localization accuracy is high. For 7 robots, this is achieved by minimizing $J_{\text{cov}}(\mathbf{x})$ with the parameters,

$$\begin{aligned}
&\mathbf{n}_1^{(1)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{n}_1^{(6)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \mathbf{n}_1^{(k)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad k = 2, \dots, 5, \\
&J_{\text{overlap}}^{mk}(\mathbf{x}) = 0, \forall k \in \mathcal{P}_s \setminus \{m\}, m \in \{1, N\},
\end{aligned} \tag{5.15}$$

and there are no inter-tag range measurements between the two GPS-enabled robots. Notice that, the robots under the bridge have a “near” straight line formation, such that they avoid unobservable ranging-tag configurations, and are additionally aided by the GPS-enabled quadcopters to localize themselves. These planning decisions are possible because of the flexibility in customizing $J_{\text{cov}}(\mathbf{x})$. In contrast, the best formation of 5 robots obtained by minimizing $J_{\text{opt}}(\mathbf{x})$ is shown in Fig. 5.6b. The two GPS-enabled robots are randomly placed



(a) Formation acquisition by minimizing $J_{\text{cov}}(\mathbf{x})$.



(b) Formation acquisition by randomly placing Robots 1 and 2 and minimizing $J_{\text{opt}}(\mathbf{x})$ for the rest of the robots.

Figure 5.6: Comparison of formations obtained by minimizing $J_{\text{opt}}(\mathbf{x})$ and $J_{\text{cov}}(\mathbf{x})$ for a bridge inspection task.

without the help of $J_{\text{opt}}(\mathbf{x})$, since otherwise, they would be very close to the other robots under the bridge, and would not receive GPS measurements. Going by visual observation only, it is clearly evident that the inspection robots are not in a straight line, thus increasing inspection time.

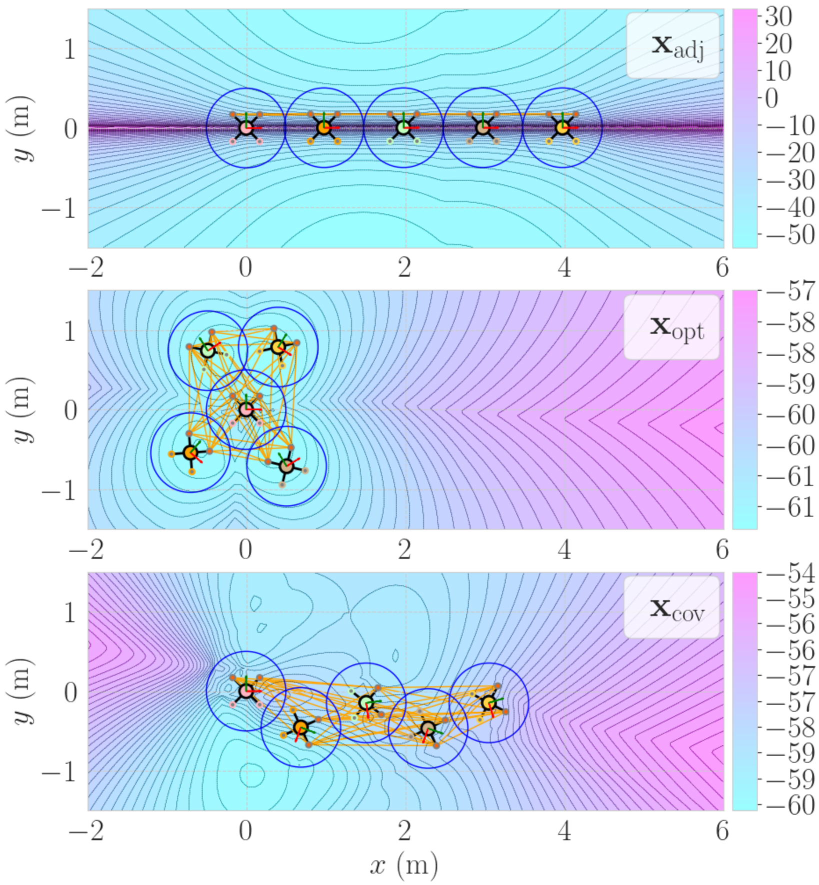


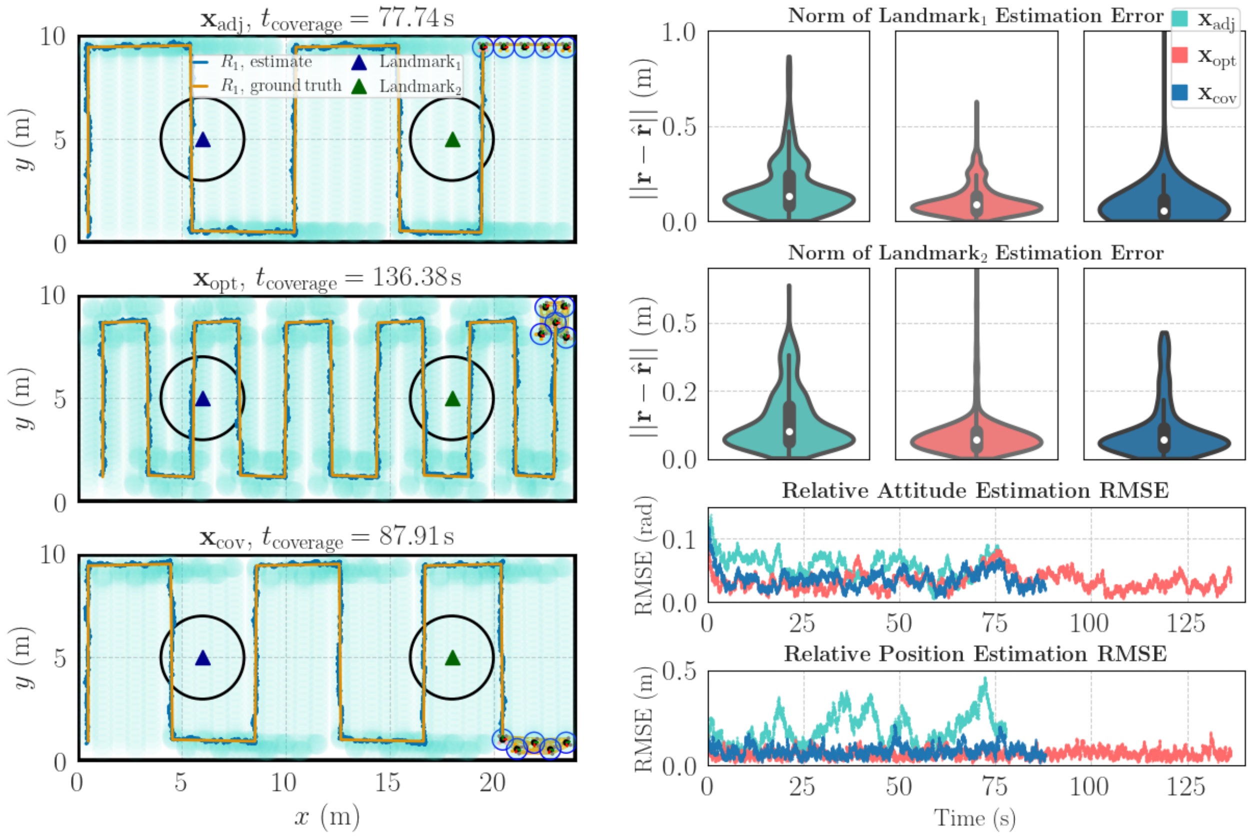
Figure 5.7: Three tested formations. The heatmap of $J_{\text{est}}(\mathbf{x})$ identifies that the straight-line formation has the highest and the cluster formation has the lowest estimation error, as expected.

5.7 Application: Multi-robot Coverage

A multi-robot coverage path planning task is where the usefulness of the proposed cost function is demonstrated with mathematical evaluation. The goal is to inspect a large area in a short amount of time, while ensuring good relative localization accuracy. This is achieved by minimizing $J_{\text{cov}}(\mathbf{x})$ with the parameters, $\mathbf{n}_1^{(k)} = [1 \quad 0]^\mathsf{T}$, $\bar{r}_k = 0.5\,\text{m}$, $k = 1, \dots, N-1$, and $\lambda = 0.25$. The resultant formation is compared with a straight-line formation and a clustered formation in a coverage path planning task. These formations, along with the heatmap of $J_{\text{est}}(\mathbf{x})$, are shown in Fig. 5.8, and denoted as,

$$\mathbf{x}_i \triangleq \arg\min_{\mathbf{x}} J_i(\mathbf{x}), \quad i \in \{\text{adj}, \text{opt}, \text{cov}\}. \tag{5.16}$$

The high-value regions in the heatmap of $\mathbf{x}_{\text{adj}}$ already indicate that this formation has low relative pose estimation accuracy.



(a) Coverage time comparison.

(b) Estimation error comparison.

Figure 5.8: Comparison of the coverage path planning task using the three formations. (a) Comparison of the coverage time for the three formations. The $\mathbf{x}_{\text{cov}}$ formation has a 35.5% time reduction, as compared to the $\mathbf{x}_{\text{opt}}$ formation, while maintaining good relative pose estimation accuracy. (b) Various RMSE plots for the three formations over 100 Monte Carlo trials. The $\mathbf{x}_{\text{cov}}$ formation has comparable inter-robot position and attitude RMSEs to the $\mathbf{x}_{\text{opt}}$ formation.

5.7.1 Simulation

The robots are initially placed near the origin of a $10\,\text{m} \times 24\,\text{m}$ area. They cover the space using a square-wave pattern often used in optimal coverage path planning problems [48, 50, 52]. For simplicity, the map of the environment is assumed to be known except for the position of two static landmarks with ranging tags fitted on them. A list of waypoints is assigned to an arbitrarily chosen leader, which is Robot 1 here, and the other robots follow the leader in a formation using the velocity control,

$$\mathbf{u}_n^{\text{reach target}/g} = \mathbf{u}_n^{\text{formation}/g} + \mathbf{u}_n^{\text{waypoint}/g}, \tag{5.17}$$

where each control term is resolved in the robot's body frame. The components $\mathbf{u}_n^{\text{formation}/g}$ and $\mathbf{u}_n^{\text{waypoint}/g}$ are given in Section 2.6 of Chapter 2. The trajectory generated using this control law is shown in Fig. 5.8a. Note that, each corner of the square-wave pattern is treated

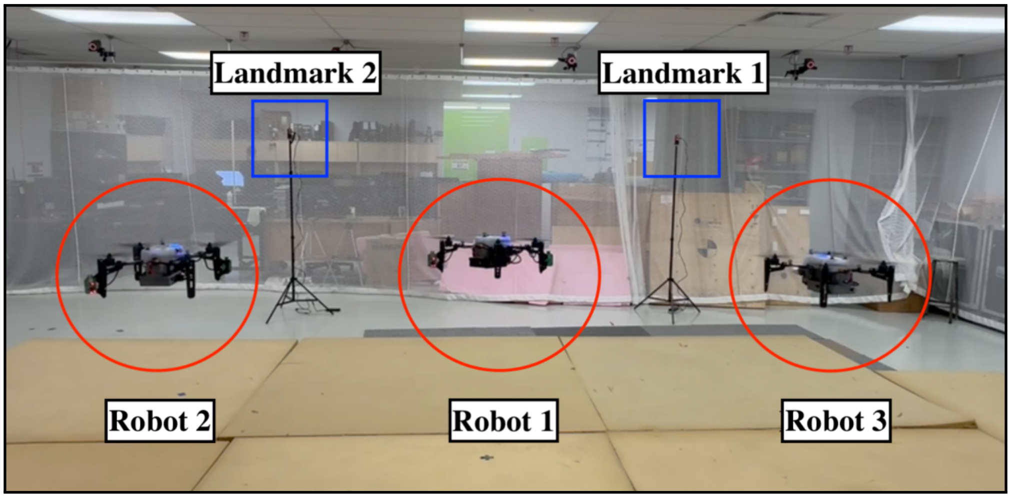
Table 5.1: Percentage reduction in median estimation error with respect to $\mathbf{x}_{\text{adj}}$ over 100 Monte Carlo simulations.

	$\mathbf{x}_{\text{opt}}$ (Eq.(5.3))	$\mathbf{x}_{\text{cov}}$ (proposed)
Landmark$_1$ Est. Error	35.4 %	58.8 %
Landmark$_2$ Est. Error	29.6 %	31.6 %
Inter-robot Att. RMSE	47.0 %	40.0 %
Inter-robot Pos. RMSE	66.2 %	59.4 %

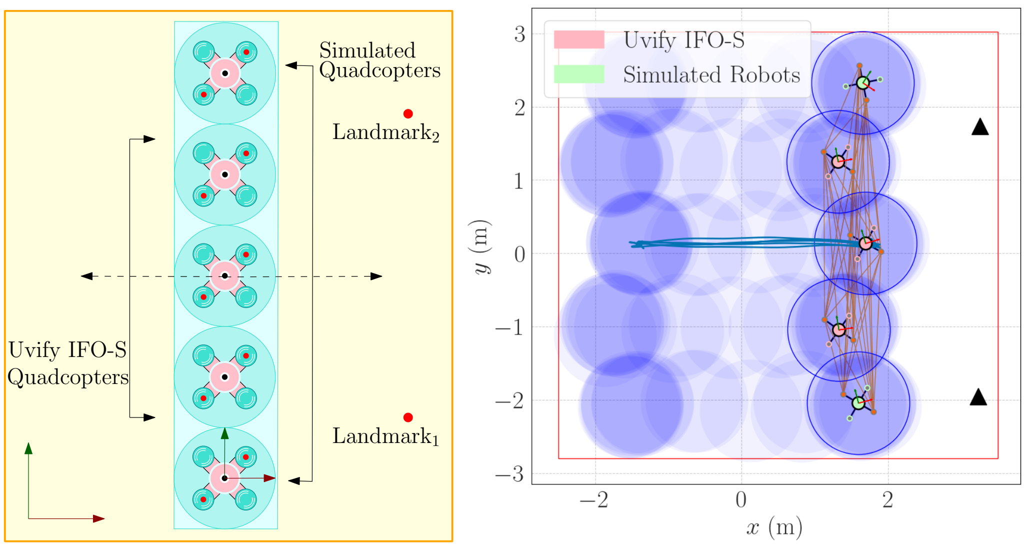
as a static waypoint. Once Robot 1 reaches one corner in formation with the other robots, it moves to the next corner.

The EKF-SLAM algorithm, similar to [24], is used to assess the relative pose estimation accuracy. This estimation directly impacts the precision of localizing the landmarks within the context of an inspection task. EKF-SLAM is used over a batch method since it is computationally less expensive and suitable for online implementation. The interoceptive measurements are the velocity inputs in the body frame of the robots at 100 Hz as shown in [47], and the exteroceptive measurements are either inter-tag or tag-landmark range measurements at 110 Hz with a covariance matrix $\mathbf{R} = 0.1^2\mathbf{1}\,\text{m}^2$. It is assumed that the robots receive range measurements from the static landmarks only when they are within a 2 m radius of the landmark. Additionally, Robot 1 receives GPS measurements at 50 Hz with a standard deviation of 0.1 m in each component to help localize itself in the global reference frame $\mathcal{F}_g$.

The $\mathbf{x}_{\text{cov}}$ (proposed) formation exhibits a 35.5% reduction in coverage time compared to $\mathbf{x}_{\text{opt}}$ (clustered formation), with only 17% and 11% loss in relative attitude and position estimation accuracy, respectively, as shown in Fig. 5.8a and Fig 5.8b. Table 5.1 displays the percentage reduction in median estimation errors of $\mathbf{x}_{\text{opt}}$ and $\mathbf{x}_{\text{cov}}$ with respect to $\mathbf{x}_{\text{adj}}$ for 100 Monte Carlo simulations. It highlights that there is a trade-off when using $\mathbf{x}_{\text{cov}}$ vs $\mathbf{x}_{\text{opt}}$; $\mathbf{x}_{\text{cov}}$ (proposed) has slightly worse inter-robot attitude and position RMSEs, but either comparable or lower landmark estimation errors than $\mathbf{x}_{\text{opt}}$, indicating $J_{\text{cov}}(\mathbf{x})$'s effectiveness in attaining highly observable, and "high-coverage" formations. The median estimation errors for $\mathbf{x}_{\text{cov}}$ (proposed) are 0.448 m, 0.088 m, 0.032 rad, and 0.062 m for Landmark$_1$, Landmark$_2$, inter-robot attitude, and position, respectively. This affirms that the proposed cost function allows a slight decrease in relative pose estimation accuracy to gain a significant reduction in coverage time, compared to the clustered formation, $\mathbf{x}_{\text{opt}}$.



(a) Experiment in progress.



(b) Visualization (left) and a top graphical view (right) of one of the experiments.

Figure 5.9: Experimental setup.

5.7.2 Experiment

The EKF-SLAM algorithm is tested with the same formations on real quadcopters to experimentally validate that the "high-coverage" formations found by minimizing $J_{\text{cov}}(\mathbf{x})$ (proposed) have good localization accuracy. Due to space limitations, each experiment is conducted with 3 Uvify IFO-S quadcopters moving back and forth in a $4\,\text{m} \times 6\,\text{m}$ space, at a constant height, while in formation for $47\,\text{s}$. Two landmarks with UWB tags are placed at the edge of the room. The remaining two robots, with two tags each, are simulated to be in formation with the other three during the experiment. The Tags i and j in the robots are

Table 5.2: Percentage reduction in median estimation error with respect to $\mathbf{x}_{\text{adj}}$ for experimental data.

	$\mathbf{x}_{\text{opt}}$ (Eq.(5.3))	$\mathbf{x}_{\text{cov}}$ (proposed)
Landmark$_1$ Est. Error	74.1 %	71.1 %
Landmark$_2$ Est. Error	24.2 %	26.9 %
Inter-robot Att. RMSE	32.4 %	32.9 %
Inter-robot Pos. RMSE	64.4 %	62.1 %

placed at

$$\mathbf{r}_p^{\tau_i p} = \begin{bmatrix} 0.17 \\ -0.17 \\ -0.05 \end{bmatrix}, \quad \mathbf{r}_p^{\tau_j p} = \begin{bmatrix} -0.17 \\ 0.17 \\ -0.05 \end{bmatrix}, \tag{5.18}$$

and $r_p = 0.7$, $p \in \mathcal{P}$, with units in meters. Since the simulations establish that the $\mathbf{x}_{\text{cov}}$ (proposed) formation reduces coverage time, the primary goal is to validate that this benefit does not significantly compromise the localization accuracy in real-world experiments. The experimental details are shown in Fig. 5.9.

The process model involves velocity inputs at 10 Hz in the body frame of the robots as shown in [47], the landmarks are static, and the measurement model involves inter-tag and tag-landmark range measurements at 80 Hz. For this experiment, DWM1000 UWB transceivers are used. The ranging protocol and UWB calibration procedure are as in [45]. Any range measurement, which does not pass the NIS test, is discarded. The velocity inputs with added noise are obtained by performing finite difference on ground truth position data, extracted from the Vicon motion-capture system. The added noise has a standard deviation of 0.01 rad and 0.1 m for the angular velocity and translational velocity components, respectively. A covariance of $0.1^2\,\mathrm{m}^2$ is set for the measurements received by the ranging tags in the simulated robots. Robot 1 is also given noisy ground truth position data as GPS measurements at 30 Hz with a standard deviation of 0.1 m in each component.

The results are shown in Fig. 5.10. As expected, the estimator diverges for the straight-line formation due to observability issues. The landmark position and inter-robot relative pose estimation accuracy for the $\mathbf{x}_{\text{cov}}$ (proposed) formation and the clustered one are similar. Furthermore, the $\mathbf{x}_{\text{cov}}$ (proposed) formation maintains landmark position estimation error within the $\pm 3\sigma$ bounds, indicating low estimation error uncertainty. In Table 5.2, this

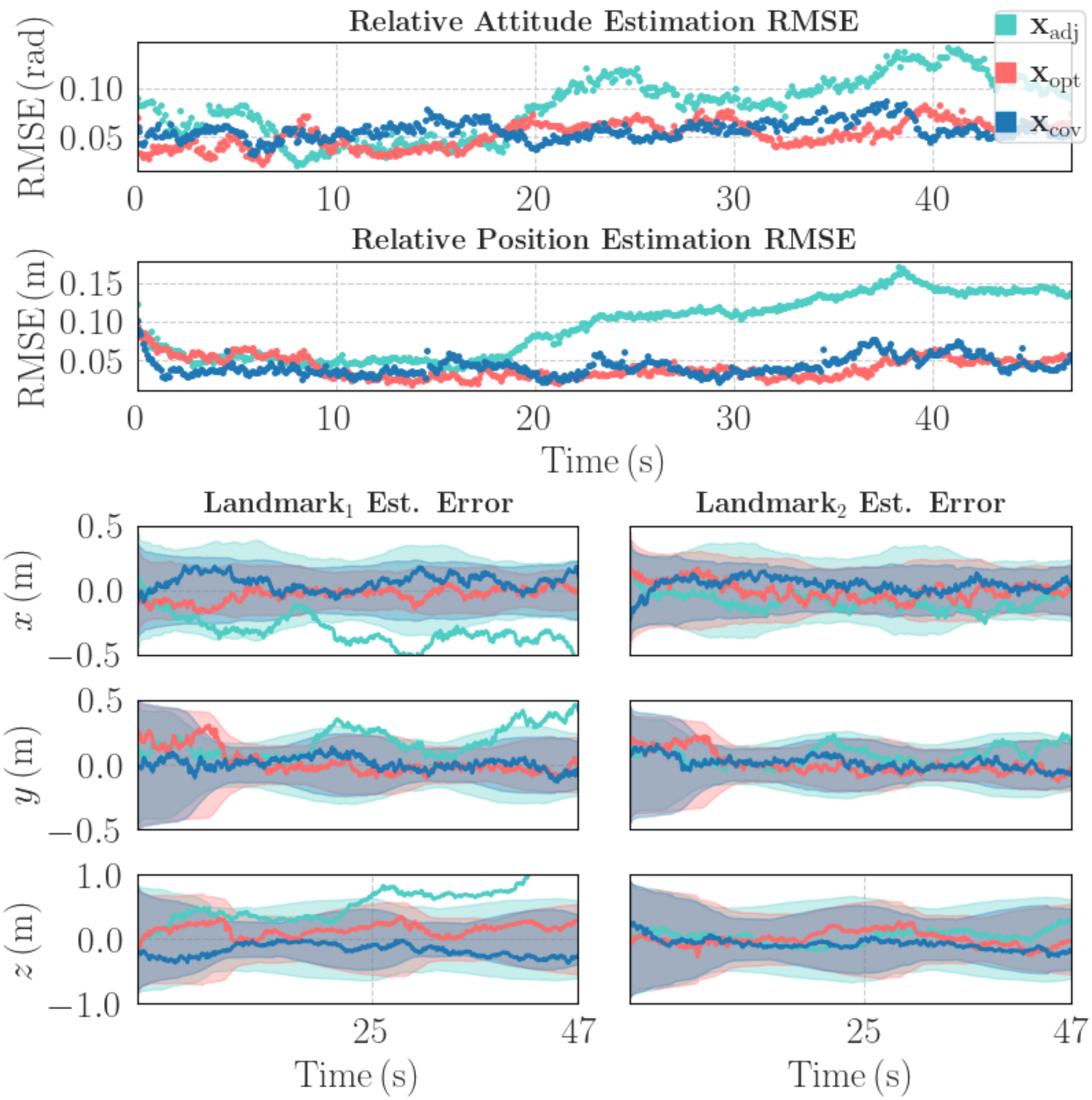


Figure 5.10: Different error metrics for the three formations in the experiment. The proposed formation has comparable RMSEs to the clustered formation while swiping a larger area. The shaded regions in the landmark position estimation error plots represent the $\pm 3\sigma$ bounds of the estimator.

formation also demonstrates a significant reduction in median estimation error compared to the straight-line formation: at least 26.9% for Landmark$_1$ and Landmark$_2$, and 32.9% and 62.1% for inter-robot attitude and position estimates, respectively, approaching levels seen in the clustered formation, $\mathbf{x}_{\text{opt}}$. These error metrics in values are 0.112 m, 0.073 m, 0.056 rad, and 0.041 m for Landmark$_1$, Landmark$_2$, inter-robot attitude, and position, respectively. The experiments again validate the claim of $J_{\text{cov}}(\mathbf{x})$ (proposed) producing "high coverage" formations with insignificant loss in relative pose estimation accuracy.

5.8 Conclusion

This chapter presents, in both simulation and experiment, that with the help of a few geometry-based constraints, "high coverage" formations can be achieved even if they are not optimal for inter-robot range-based relative pose estimation. The decrease in estimation accuracy for these formations is negligible. The easy customizability of the proposed cost function to achieve "high coverage" formations with acceptable relative pose estimation accuracy is one of its strongest points. It can be used for a variety of applications such as multi-robot coverage, multi-robot search and rescue, and multi-robot inspection. Future work includes adopting this cost function for problems in 3D and extending the implementation of this cost function in online planning initiatives where the robots are tasked to cover a large area while avoiding obstacles.

Chapter 6

Concluding Remarks

This thesis presents novel estimators and planning initiatives for multi-robot localization and planning using small, cheap, and low-power UWB range sensors. The work is motivated by the realization that the use of range sensors is multi-robot systems gives rise to observability issues that may result in inconsistent relative pose estimates between the robots using traditional filtering and planning algorithms. The algorithms and methods presented in this thesis are analyzed and validated using both simulation and real-world experiments. However, there are several limitations and future work that can be done to improve the proposed algorithms and methods.

Chapter 4 gives a detailed account of the possible ambiguities that arise in the range-based relative pose estimation problem, due to observability issues. The chapter presents a novel Gaussian-Sum Filter (GSF) that is capable of handling these ambiguities in 3D. The proposed estimation algorithm has a similar performance to the Particle Filter, but with a significantly lower computational cost. More specifically, in this chapter, it is shown that the state of the system, consisting of relative poses between robots in $SE(3)$ has a multimodal distribution. A geometrically initialized least-sqaures estimator helps model the state's multimodal distribution as a Gaussian Mixture Model (GMM). The Gaussian-sum filter is initialized with the GMM and then coverages to an unimodal distribution once the robots move sufficiently.

Further improvements can however be made to the GSF for future work. After the GSF converges and follows the ground truth with good consistency, if the robots come into static motion, or if the robots maintain a constant formation, there may not be sufficient motion to disambiguate the multimodal distribution. In such cases, the GSF may not converge to the ground truth. This is a limitation can be addressed by reinitializing the GSF with a GMM that models the new ambiguities after a certain period of static motion is detected. Finding this GMM is a challenging problem and may require a combination of geometrical methods and batch optimization methods and is left for future work.

Chapter 5 proposes a cost function, the minimization of which results in optimal multi-robot formations that balance observability and user-defined configurations. In this chapter, it is shown that with range-based systems, there are formations, that may be useful for applications such a fast inspection of a warehouse or an agricultural land, but the observability of the system is compromised. The proposed cost function is a weighted sum of the observability of the system and any desirable geometric constraints that the user may have. As such, useful formations can be found that are both observable and satisfy the user-defined constraints and can be used for applications such as fast inspection of a factory or bridge.

For future work, the proposed cost function can be used for online planning initiatives, where the robots can change their formation based on the environment and the task at hand. The cost function can be used in a fashion, where the robots plan their formation for a short period of time, execute the plan, and replan based on the new information. A component can be added to this cost function, which creates a repulsive potential field around obstacles, so that the robots find a formation that is both observable and avoids obstacles. Additionally, the gradient of the proposed cost function is computed using finite difference, which can be replaced with evaluating the gradient analytically. This will result in a faster convergence of the optimization algorithm, and help make real-time planning possible.

Bibliography

[1] Y. Cao, C. Yang, R. Li, A. Knoll, and G. Beltrame, "Accurate position tracking with a single UWB anchor", in *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020.

[2] B. Hepp, T. Naegeli, and O. Hilliges, "Omni-directional person tracking on a flying robot using occlusion-robust ultra-wideband signals", *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea, 2016.

[3] S. Güler, J. Jiang, A. A. Alghamdi, R. I. Masoud, and J. S. Shamma, "Real Time Onboard Ultrawideband Localization Scheme for an Autonomous Two-robot System", in *IEEE Conference on Control Technology and Applications (CCTA)*, Copenhagen, Denmark, 2018.

[4] Z. Sahinoglu, S. Gezici, and I. Güvenc, *Ultra-wideband Positioning Systems*. Cambridge University Press, 2008.

[5] P. T. Morón, J. P. Queralta, and T. Westerlund, "Towards Large-Scale Relative Localization in Multi-Robot Systems with Dynamic UWB Role Allocation", in *International Conference on Robotics and Automation Engineering (ICRAE)*, Singapore, 2022.

[6] W. Shule, C. Almansa, J. P. Queralta, Z. Zou, and T. Westerlund, "UWB-Based Localization for Multi-UAV Systems and Collaborative Heterogeneous Multi-Robot Systems: a Survey", *Procedia Computer Science*, vol. 175, pp. 357–364, 2020.

[7] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation", in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, 2015.

[8] X. Fang, C. Wang, T.-M. Nguyen, and L. Xie, "Graph Optimization Approach to Range-Based Localization", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6830–6841, 2021.

[9] G. Shen, R. Zetik, O. Hirsch, and R. Thomä, "Range-Based Localization for UWB Sensor Networks in Realistic Environments", *EURASIP Journal on Wireless Communications and Networking*, 2010.

[10] K. Guo, D. Han, and L. Xie, "Range-based cooperative localization with single landmark", *IEEE International Conference on Control and Automation (ICCA)*, Ohrid, Macedonia, 2017.

[11] M. Shalaby, C. C. Cossette, J. R. Forbes, and J. Le Ny, "Relative Position Estimation in Multi-Agent Systems Using Attitude-Coupled Range Measurements", *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 3, pp. 4955–4961, 2021.

[12] C. C. Cossette, M. Shalaby, D. Saussié, J. R. Forbes, and J. Le Ny, "Relative Position Estimation Between Two UWB Devices With IMUs", *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 3, pp. 4313–4320, 2021.

[13] H. Xu, Y. Zhang, B. Zhou, L. Wang, X. Yao, G. Meng, and S. Shen, "Omni-Swarm: A Decentralized Omnidirectional Visual–Inertial–UWB State Estimation System for Aerial Swarms", *IEEE Transactions on Robotics*, vol. 38, pp. 3374–3394, 2021.

[14] K. Guo, X. Li, and L. Xie, "Ultra-Wideband and Odometry-Based Cooperative Relative Localization With Application to Multi-UAV Formation Control", *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2590–2603, 2020.

[15] T.-M. Nguyen, Z. Qiu, T. H. Nguyen, M. Cao, and L. Xie, "Distance-Based Cooperative Relative Localization for Leader-Following Control of MAVs", *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3641–3648, 2019.

[16] S. Güler, M. Abdelkader, and J. S. Shamma, "Infrastructure-free Localization of Aerial Robots with Ultrawideband Sensors", *arXiv:1809.08218 [cs.ro]*, 2018.

[17] T.-M. Nguyen, A. Hanif Zaini, C. Wang, K. Guo, and L. Xie, "Robust Target-Relative Localization with Ultra-Wideband Ranging and Communication", in *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, Australia, 2018.

[18] Y. Cao, M. Li, I. ŠVogor, S. Wei, and G. Beltrame, "Dynamic Range-Only Localization for Multi-Robot Systems", *IEEE Access*, vol. 6, pp. 46 527–46 537, 2018.

[19] A. Fishberg and J. P. How, "Multi-Agent Relative Pose Estimation with UWB and Constrained Communications", in *International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, 2022.

[20] Y. Xianjia, L. Qingqing, J. P. Queralta, J. Heikkonen, and T. Westerlund, "Cooperative UWB-Based Localization for Outdoors Positioning and Navigation of UAVs aided by Ground Robots", in *IEEE International Conference on Autonomous Systems (ICAS)*, 2021.

[21] C. C. Cossette, M. A. Shalaby, D. Saussié, J. L. Ny, and J. R. Forbes, "Optimal Multi-robot Formations for Relative Pose Estimation Using Range Measurements",

in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, 2022.

[22] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2023.

[23] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group", *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[24] J. Sola, "Simulataneous localization and mapping with the extended Kalman filter", 2014.

[25] J. Solà, J. Deray, and D. Atchuthan, "A Micro Lie Theory for State Estimation in Robotics", *arXiv: 1812.01537 [cs.RO]*, 2021.

[26] T. D. Barfoot and P. T. Furgale, "Associating Uncertainty With Three-Dimensional Poses for Use in Estimation Problems", *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

[27] M. de Queiroz, X. Cai, and M. Feemster, *Formation Control of Multi-Agent Systems*. John Wiley & Sons, Ltd, 2019.

[28] S. Särkkä, *Bayesian Filtering and Smoothing*, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

[29] J. Elfring, E. Torta, and R. van de Molengraft, "Particle Filters: A Hands-On Tutorial", *Sensors*, vol. 21, no. 2, 2021.

[30] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2002.

[31] S. Li, C. De Wagter, and G. C. H. E. De Croon, "Self-supervised Monocular Multi-robot Relative Localization with Efficient Deep Neural Networks", in *IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, USA, 2022.

[32] Y. Cao and G. Beltrame, "VIR-SLAM: visual, inertial, and Ranging SLAM for single and multi-robot systems", *Autonomous Robots*, vol. 45, pp. 905 –917, 2020.

[33] S. Kamthe, J. Peters, and M. P. Deisenroth, "Multi-modal Filtering for Non-linear Estimation", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014.

[34] S. Maskell and S. Julier, "Optimised Proposals for Improved Propagation of Multi-modal Distributions in Particle Filters", in *International Conference on Information Fusion*, Istanbul, Turkey, 2013.

[35] O. Ö. Bilgin and M. Demirekler, "Multi Mode Projectile Tracking with Marginalized Particle Filter", in *IEEE Radar Conference*, Johannesburg, South Africa, 2015.

[36] P. Nguyen and H. B. Le, "A Multi-modal Particle Filter Based Motorcycle Tracking System", in *Pacific Rim International Conference on Artificial Intelligence*, 2008.

[37] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "A novel Gaussian Sum Filter Method for accurate solution to the nonlinear filtering problem", in *International Conference on Information Fusion*, 2008.

[38] J.-H. Kim and D. Kim, "Computationally Efficient Cooperative Dynamic Range-Only SLAM Based on Sum of Gaussian Filter", *Sensors*, vol. 20, 2020.

[39] M. Raitoharju, Ángel García-Fernández, R. Hostettler, R. Piché, and S. Särkkä, "Gaussian mixture models for signal mapping and positioning", *Signal Processing*, vol. 168, p. 107 330, 2020.

[40] J. Ćesić, I. Marković, and I. Petrović, "Mixture Reduction on Matrix Lie Groups", *IEEE Signal Processing Letters*, vol. 24, no. 11, pp. 1719–1723, 2017.

[41] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill, Inc., 2008.

[42] R. S. Zengin and V. Sezer, "A Novel Point Inclusion Test for Convex Polygons Based on Voronoi Tessellations", *Applied Mathematics and Computation*, vol. 399, p. 126 001, 2020.

[43] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer New York Inc., 2001.

[44] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator", *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[45] M. A. Shalaby, C. C. Cossette, J. R. Forbes, and J. Le Ny, "Calibration and Uncertainty Characterization for Ultra-Wideband Two-Way-Ranging Measurements", in *IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[46] M. A. Shalaby, C. C. Cossette, J. L. Ny, and J. R. Forbes, *Multi-Robot Relative Pose Estimation and IMU Preintegration Using Passive UWB Transceivers*, 2023. arXiv: `2304.03837 [cs.RO]`.

[47] S. S. Ahmed, M. A. Shalaby, C. C. Cossette, J. L. Ny, and J. R. Forbes, "Gaussian-Sum Filter for Range-based 3D Relative Pose Estimation in the Presence of Ambiguities", 2024. arXiv: `2402.08566 [cs.RO]`.

[48] Z. Chen, Z. Peng, L. Jiao, and Y. Gui, "Efficient Multi-Robot Coverage of an Unknown Environment", in *Chinese Control Conference (CCC)*, Shanghai, China, 2021.

[49] J. Tang and H. Ma, "Mixed Integer Programming for Time-Optimal Multi-Robot Coverage Path Planning With Efficient Heuristics", *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6491–6498, 2023.

[50] C. Gao, Y.-X. Kou, Z. Li, A. Xu, Y. Li, and Y. zhe Chang, "Optimal Multirobot Coverage Path Planning: Ideal-Shaped Spanning Tree", *Mathematical Problems in Engineering*, 2018.

[51] S. Lee, "An Efficient Coverage Area Re-Assignment Strategy for Multi-Robot Long-Term Surveillance", *IEEE Access*, vol. 11, pp. 33 757–33 767, 2023.

[52] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-robot Forest Coverage", in *IEEE International Conference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, 2005.

[53] A. D. Dang, H. M. La, T. Nguyen, and J. Horn, "Formation control for autonomous robots with collision and obstacle avoidance using a rotational and repulsive force–based approach", *International Journal of Advanced Robotic Systems*, vol. 16, no. 3, p. 1 729 881 419 847 897, 2019.

[54] X. Yan, D. Jiang, R. Miao, and Y. Li, "Formation Control and Obstacle Avoidance Algorithm of a Multi-USV System Based on Virtual Structure and Artificial Potential Field", *Journal of Marine Science and Engineering*, vol. 9, no. 2, 2021.

[55] H. Li, Y. Liu, L. Fan, and X. Chen, "Towards robust and optimal image stitching for pavement crack inspection and mapping", *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Macau, China, 2017.

[56] N. Qian, "On the Momentum Term in Gradient Descent Learning Algorithms", *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.

[57] C. C. Cossette, A. Walsh, and J. R. Forbes, "The Complex-Step Derivative Approximation on Matrix Lie Groups", *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 2, pp. 906–913, 2020.

[58] J. Le Ny and S. Chauvière, "Localizability-Constrained Deployment of Mobile Robotic Networks with Noisy Range Measurements", in *Annual American Control Conference (ACC)*, Milwaukee, WI, USA, 2018.

[59] J. Cano and J. Le Ny, "Ranging-Based Localizability Optimization for Mobile Robotic Networks", *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2842–2860, 2023.

[60] Y. Xia, X. Na, Z. Sun, and J. Chen, "Formation Control and Collision Avoidance for Multi-agent Systems Based on Position Estimation", *ISA Transactions*, vol. 61, pp. 287–296, 2016.

[61] H. W. Kuhn, "The Hungarian Method for the Assignment Problem", *Naval Research Logistics (NRL)*, vol. 52, 1955.