

Cooperative Relative Localization in MAV Swarms with Ultra-wideband Ranging

Master of Science Thesis

Changrui Liu

Cooperative Relative Localization in MAV Swarms with Ultra-wideband Ranging

by

Changrui Liu

to obtain the degree of Master of Science,
at the Delft University of Technology,
to be defended publicly on Wednesday November 23rd, 2022 at 13:00

Affiliation:	C&S, Faculty of Aerospace Engineering	
Thesis Duration:	February 2022 - November 2022	
Thesis Committee:	Prof. dr. G.C.G.E. de Croon	TU Delft, MAVLab (Chair & Supervisor)
	Assoc. Prof. dr. M.Mazo Espinosa	TU Delft, DCSC
	Dr. ir. C. de Wagter	TU Delft, MAVLab
	S.U.Pfeiffer	TU Delft, MAVLab (Supervisor)

Cover Image: A swarm of flying birds from <https://pixabay.com/images/id-3406667/>

Preface

It is a pleasure to conduct thesis research during my study at the faculty of aerospace engineering, TU Delft. The topic is on the cooperative relative localization problem in MAV swarms with ultra-wideband ranging, which essentially is to solve the underlying state estimation problem. The work that has been conducted only ends with computer simulations due to limited project time, and further experiments on real MAVs will continue to substantiate this robotic research.

This thesis report consists of two separate parts. The first part is a self-contained scientific paper that summarizes the main findings of the thesis work, while the second part is a literature study that provides more background details.

This is the first time I do a research project in the area of state estimation, which helped me learn a lot about the general control theory given that my previous research experiences only touched on designing control laws and decision-making. Therefore, I would like to first thank my supervisor, Prof. de Croon, for giving me this wonderful topic that inevitably will have a great impact on my future academic journey. During the past months of conducting this thesis research, I am also grateful for the patience, encouragement, and kind support from my daily supervisors, Prof. de Croon and Sven, indeed, I could never finish this work without their suggestions, inspiration, and also judgments. Besides, I would also like to thank Prof. J.H. van Schuppen who taught me stochastic systems theory, from which I also benefited a lot for doing this thesis research. In addition, I also would like to thank Prof. Q. Shen from Shanghai Jiao Tong University, Shanghai, Prof. S.Z. Yong from Northeastern University, Boston, Prof. M.P. Chapman, and Prof. R.H. Kwong, both from University of Toronto, Toronto, since they all provided me with helpful discussions during the thesis work. Lastly, I would like to thank my parents, and my friends both in China and in the Netherlands for bringing happiness and joy into my daily life.

*Changrui Liu
Delft, November 2022*

Introduction

Why swarming?

Swarm micro aerial vehicles (MAVs), inspired by animals (e.g., bees and birds), have undergone rapid development and demonstrated potential to benefit humanity at large [9, 22, 71] owing to their superiority in performing complex tasks such as cooperative transportation, surveillance, and detection.

Having a larger group with multiple agents at hand has many advantages. First, the sensing and actuating capabilities are enhanced. Second, the increased redundancy provides system-wide fault tolerance. Finally, efficiency can be increased by distributing and assigning tasks to different agents.

Why direct relative localization?

Given that localization is the foremost prerequisite in robotic operations from the navigational perspective, achieving accurate localization is always desirable regardless of the type of robot.

Swarming is powerful, but it also brings challenges to the built robotic system, among which relative localization is one of the fundamental problems that need to be well solved. Relative localization solutions can be broadly categorized into indirect ones and direct ones. Indirect relative localization means to first determine the *absolute* positions of the robots in a common reference frame and then compute the relative positions thereafter, whereas direct relative localization obtains the relative positions directly using onboard sensors and inter-agent sensing without relying on external devices (e.g., GPS, ground stations, fixed beacons, and optical motion tracking systems). Direct solutions are more favorable than indirect ones since they are more economical and easier to set up. Therefore, it is valuable to develop GPS-denied and infrastructure-free direct relative localization solutions for MAV swarms.

Why Ultra-wideband (UWB)? & Additional hardware settings

In the framework of direct relative localization, onboard sensors play more dominant roles. Considering that MAVs have limited power supply, light and computationally efficient onboard sensors are preferred.

In terms of inter-agent sensing, ultra-wideband (UWB) telecommunication is a perfect option in that it can also provide accurate inter-agent distance measurements if each of the MAVs is equipped with a UWB tag. By combining communication and relative sensing capabilities, UWB is superior to the microphone and infrared sensors. Besides, the centimeter-level distance measurement accuracy of UWB is higher than that of Bluetooth and Wi-Fi. More importantly, UWB provides omnidirectional sensing, unlike vision-based sensors which suffer from the limited field-of-view issue.

Though placing multiple UWB tags on one MAV can provide bearing measurements or attitude-related information between a pair of agents, we restrict ourselves to the lighter and cheaper setting where only one UWB tag is equipped on each of the MAVs, which leads to the distance-based relative localization problem. Considering that the relative localization for MAVs requires estimating the relative position in 3-D, distance-based schemes are more challenging, and representative previous works either only consider 2-D operation [49, 50] or rely on altitude measurements from altimeters [94]. In this regard, it is also of great interest to find a distance-based 3-D relative localization solution.

Relative localization for swarms

For MAV swarms, the goal is, for each agent in the swarm (referring to as the host agent), to determine the relative position of all its neighboring agents in the body frame of this host agent. When dealing with body frames, it is common to use Euler angles, quaternions, or the general rotation matrix. To simplify the problem, we focus on the relative localization in the body-centered horizontal frame where the heading/yaw angle is the only variable that determines the corresponding body-centered horizontal frame. To extract the relative position, the most effective and straightforward method is to exchange velocity information between agents. At this point, we finally can synthesize the problem as follows.

Problem: Relative Localization for Swarms

For a given agent in the swarm, estimate the relative position of all its neighboring agents in its body-centered horizontal frame using only the velocity measurements obtained by the IMUs of each of the involved agents and the inter-agent distance measurements obtained by UWB units.

From peer-to-peer relative localization to cooperative relative localization

Swarming poses challenges to the relative localization problem especially when we additionally set many hardware constraints. However, swarming also brings benefits in that relative localization can be viewed as a task per se. Thus, performing relative localization cooperatively to fully exploit the potential of having a larger group is a promising direction to continue. Though the peer-to-peer distance-based relative localization as done by Li et al. [94] suffices to solve the desired problem, the localization accuracy is not satisfactory enough, which also necessitates research on cooperative relative localization. In the context of cooperative relative localization, the host agent and its neighbors are combined as an integrated system where couplings between different host-neighbor pairs and additional distances between any pair of binding neighbors are present, which leads to the following questions.

Q1-Q3: Cooperative Relative Localization

- Q1: How to cooperatively perform relative localization?
- Q2: How to account for the coupling between different host-neighbor pairs and does accounting for this coupling help improve localization accuracy?
- Q3: How to properly incorporate the indirect distances between neighbors and what are the side effects of doing this?

Still, we should also be asking ourselves about any proposed cooperative scheme the following more basic questions.

Q4-Q6: More on the Cooperative Relative Localization

- Q4: Can the cooperative scheme improve the estimation accuracy after all?
- Q5: What are possible theoretical explanations of using the cooperative scheme if it provides improved estimation accuracy?
- Q6: What are possible theoretical explanations of using the cooperative scheme if it does *not* provide improved estimation accuracy?

State estimation in relative localization

When comes to the algorithmic part of the relative localization problem where a state estimation algorithm or a stochastic realization needs to be developed to reconstruct the state of interest, existing solutions can be categorized as filtering-based (e.g. extended Kalman filter (EKF) [25, 94], unscented Kalman filter (UKF), and particle filter [48]) or optimization-based (e.g. moving horizon estimation [102], semidefinite programming [96], and quadratic programming [109]).

Considering the limited computational resource of MAVs during online operation, it is favorable to implement algorithms of low computational complexity, moreover, it is also desired to have fewer auxiliary modules to assist the main state estimation algorithm. Optimization-based approaches are computationally expensive and fail to address measurement outliers properly without an additional outlier rejection algorithm. For filtering-based approaches, the UKF, the particle filter, and many other variations that also rely on the general *parallel multi-filtering principle* (e.g., cubature Kalman filter [165], multi-hypothesis Kalman filter [19]) all suffer from the high computational complexity issue. In this regard, EKF becomes the ideal option due to its simple recursive structure which enables its easy implementation in real microchips.

However, one of the challenges in the relative localization using UWB is dealing with the heavy-

tailed noise of the UWB distance measurements. Traditional EKF, being a suboptimal filter due to linearization, provides an even more inaccurate estimation if the noise is non-Gaussian and/or the covariance information of the noise distribution is unknown (i.e., poor initialization of the covariance matrices). In this regard, we also aim to answer the following questions.

Q7-Q9: EKF based State Estimation

Q7: How to address the heavy-tailed UWB noise in the framework of EKF?

Q8: How to make the EKF robust against incorrect initialization of the covariance matrices?

Q9: Given that a proposed advanced EKF addresses Q7 and Q8, whether its computational complexity is acceptable? If not, how to reduce it or possibly make a compromise?

Organization of the thesis report

The thesis report has two main parts. The first part is a self-contained scientific article that answers the research questions listed above, and it has its own tailored introduction and references. The second part is a literature study that was carried out at the very early stage of this thesis research to fulfill the required courses AE4010 (Research Methodology) and AE4020 (Literature Review), and it has already been graded by the corresponding supervisors. The literature study is also a self-contained document that has its own introduction and references, and the nomenclature and list of tables/figures in the second part are only responsible for the literature study itself. Besides, the cited articles in this introduction are all listed in the references of the literature review.

Contents

Preface	i
Introduction	ii
Part I: Scientific Article	1
Part II: Literature Study	2
Nomenclature	3
List of Tables	6
1 Introduction	7
1.1 Background	7
1.2 Related Work	8
1.3 Outline of the Review	10
2 Ultra-wideband Ranging for Localization	11
2.1 Ranging with Ultra-wideband	11
2.1.1 Two-step Positioning	11
2.1.2 Error Sources of UWB Ranging	12
2.2 Communication Protocols for Ultra-wideband Ranging	12
2.3 Distance Measurement Model using Ultra-wideband	13
2.3.1 Bias identification and Mitigation	13
2.3.2 Noise Modeling	13
3 Overview of Cooperative Localization	14
3.1 Early Days of Cooperative Localization	14
3.2 Centralized Cooperative Localization	15
3.3 Decentralized Cooperative Localization	16
3.3.1 Tightly Coupled Algorithms	17
3.3.2 Loosely Coupled Algorithms	17
3.4 Cooperative Relative Localization	18
3.4.1 Representative Solutions	18
3.4.2 Cooperation for Relative Localization of Dynamic Networks	20
4 Filtering Theory Based Cooperative Relative Localization	22
4.1 Kalman Filter Based Cooperative Relative Localization	22
4.1.1 Kalman Filter Introduction	22
4.1.2 Observability Analysis	25
4.1.3 Covariance Intersection	25
4.1.4 Initialization & Extensions	26
4.2 Particle Filter Based Cooperative Localization	28
4.2.1 Introduction to Particle Filter	28
4.2.2 Relative Localization Using Particle Filter	32
4.3 Moving Horizon Estimation Based Cooperative Relative Localization	33
4.3.1 Introduction to Moving Horizon Estimation	33
4.3.2 Relative Localization Using Moving Horizon Estimation	35
4.4 Comparison of Different Filters	35

5	Graph Rigidity Theory Based Cooperative Relative Localization	37
5.1	Introduction to Graph Rigidity Theory	37
5.2	Rigidity Theory for Cooperative Relative Localization	40
5.3	Optimization for Rigidity Theoretic Approaches	41
5.3.1	Introduction to Semidefinite Programming	41
5.3.2	Semidefinite Programming for Cooperative Relative Localization	42
5.3.3	Relaxation in Semidefinite Programming	43
5.4	Advances in Rigidity Theory	43
5.4.1	Rigidity Theory Meets Ultra-wideband.	43
5.4.2	Rigidity Theory Meets Filtering Theory	44
5.4.3	Rigidity in Barycentric Coordinates	44
6	Conclusion	46
6.1	Concluding Remarks of Different Techniques.	46
6.2	Research Questions & Research Objectives	46
6.2.1	Research Questions	46
6.2.2	Research Objective	47
6.3	Inspiring Future Research Instruction	47
	References	56

Part I: Scientific Article

Cooperative Relative Localization in MAV Swarms with Ultra-wideband Ranging

Changrui Liu, Sven Pfeiffer and Guido C.H.E. de Croon

Abstract—Relative localization (RL) is essential for the successful operation of micro air vehicle (MAV) swarms. Achieving accurate 3-D RL in infrastructure-free and GPS-denied environments with only distance information is a challenging problem that has not been satisfactorily solved. In this work, based on the range-based peer-to-peer RL using the ultra-wideband (UWB) ranging technique, we develop a novel UWB-based cooperative relative localization (CRL) solution which integrates the relative motion dynamics of each host-neighbor pair to build a unified dynamic model and takes the distances between the neighbors as *bonus information*. Observability analysis using differential geometry shows that the proposed CRL scheme can expand the observable subspace compared to other alternatives using only direct distances between the host agent and its neighbors. In addition, we apply the kernel-induced extended Kalman filter (EKF) to the CRL state estimation problem with the novel-designed Logarithmic-Versoria (LV) kernel to tackle heavy-tailed UWB noise. Sufficient conditions for the convergence of the fixed-point iteration involved in the estimation algorithm are also derived. Comparative Monte Carlo simulations demonstrate that the proposed CRL scheme combined with the LV-kernel EKF significantly improves the estimation accuracy owing to its robustness against both the measurement outliers and incorrect measurement covariance matrix initialization. Moreover, with the LV kernel, the estimation is still satisfactory when performing the fixed-point iteration only once for reduced computational complexity.

Index Terms—Cooperative Relative Localization, Ultra-wideband (UWB), Kernel-induced Kalman filtering, Observability Analysis

I. INTRODUCTION

MICRO Air Vehicles (MAVs), with higher agility and a lighter design, have been widely used in many real-world operations such as surveillance [1], exploration [2], and urban construction [3]. Swarm behaviors observed in the natural world (e.g. starlings and bees) motivate the development of MAV swarms to compensate the limited power and mobility of a single MAV agent. Swarming allows robots to perform more complex tasks (e.g., cooperative transportation [4] and gas detection [5]) where efficiency and system redundancy are of major consideration.

Successful operation of an MAV-swarm requires an accurate relative localization (RL) scheme which then provides essential position feedback among agents for collision avoidance [6] and formation control [7], both of which are prerequisites for higher-level objectives. RL schemes can be broadly categorized as indirect and direct ones. Indirect solutions compute

This work is for partial fulfillment of the MSc degree in Aerospace Engineering, Delft University of Technology

The authors are with the Micro Air Vehicle Laboratory, Faculty of Aerospace Engineering, Delft University of Technology, 2628CD Delft, The Netherlands (e-mail: c.liu-31@student.tudelft.nl; s.u.pfeiffer@tudelft.nl; g.c.h.e.decroon@tudelft.nl)

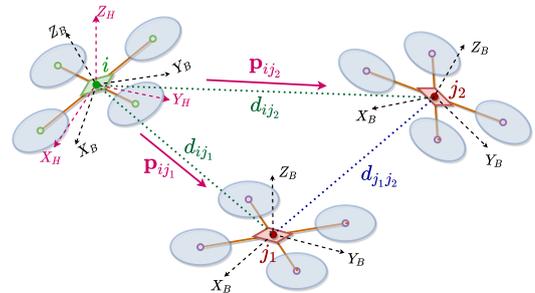


Fig. 1. Cooperative relative localization problem, where agent i aims to localize its neighbors j_1 and j_2 in its body centered horizontal frame, i.e., agent i needs to compute \mathbf{p}_{ij_1} and \mathbf{p}_{ij_2} .

relative position using global positions in a shared frame, typical examples use GPS [8], [9], optical motion tracking systems [10] and anchor-dependent positioning systems with fixed beacons or ground stations [11]–[13]. The performance of GPS degrades substantially in indoor and urban environments, and localization relying on external devices requires additional investments (time and money) for setting up the area of operation. Consequently, it is favorable to develop *GPS-denied* and *infrastructure-free* RL solutions that provide position estimation *directly* through inter-agent sensing and communication.

For direct RL schemes where onboard sensors play a more dominant role, the visual-odometry methods [14], [15] which extract relative positions using cameras are the most straightforward. However, the high computational cost and limited field-of-view of the vision-based methods are not ideal for MAVs. Excluding visual information, fusing IMU with other ranging sensors is effective and implementable. Ultra-wideband (UWB) technology has recently drawn great attention in aerial robots localization tasks [7], [16], [17] owing to its superior communication capability and ranging accuracy compared to infrared sensors, microphones, and Bluetooth [18]. Therefore, equipping MAVs with UWB tags is a lightweight and economic solution for the RL purpose in swarming. In terms of UWB-based RL, the underlying problem is 3-D position estimation with inter-agent *distance* measurements. Some previous attempts either consider 2-D scenarios by using altimeters [7], [17] or use multiple UWB tags to provide bearing information [16], [19]. The *purely distance-based* 3-D RL problem with only one UWB tag equipped on each MAV has not been satisfactorily solved, and typical solutions relying on graph rigidity theory [20] or optimization [21], [22] not only fail to address noise issues in the measurements but also rely on a common reference frame which is not available to MAVs flying in unknown environments. In fact, due to multi-pathing and non-line-of-

sight (NLoS) effects, the UWB distance measurements are noisy, thus making filtering-based estimation algorithms more suitable for accurate localization. Considering the limited onboard computational resources and the nonlinear nature of the relative motion dynamics, the estimation algorithm design for the distance-based 3-D RL becomes more challenging.

In this work, we aim to address the RL problem for MAV swarms by determining for each agent in the swarm the relative position of its neighboring agents in its body-centered horizontal frame. Though the basic peer-to-peer RL (i.e., performing position estimation for each of the neighbors individually as done by Li et al. [17]) suffices to solve this problem, the benefits of swarming are not fully exploited. The main contributions of this work are twofold. First, we propose a novel cooperative relative localization (CRL) scheme which integrates the relative motion dynamics of each of the neighbors to build a unified dynamic model and also takes the distances between the neighbors as *bonus information*. The resulting CRL can account for the correlation induced by the same velocity input as well as improve localization accuracy in general by introducing more measurements. Second, inspired by the kernel-based filtering methods [23], we propose a maximum Logarithmic-Versoria criterion-based extended Kalman filtering (MLVC-EKF) for the state estimation in localization tasks to handle the heavy-tailed noise in UWB distance measurements. The proposed filtering strategy substantially improves robustness against measurement outliers, especially for the CRL framework where more measurements are present.

The rest of the paper is organized as follows. In Section II, we present mathematical convention and some theoretical backgrounds. Then, Section III formulates the state estimation problem for the CRL. In the following Section IV, we provide the MLVC-EKF algorithm and analyze the CRL problem in detail. The simulation results and the discussions are given in Section V. Lastly, we conclude the work and delineate future research directions in VI.

II. PRELIMINARIES & PROBLEM FORMULATION

In this section, we will first present some mathematical conventions used throughout the paper. Subsequently, we will present the details of the relative motion dynamics which will be used for further CRL modeling. Finally, we elaborate on the UWB noise modeling where we consider various practical issues (e.g., NLoS, multi-pathing, and transmission delay).

A. Notation and Definitions

Let $\mathbf{x} \in \mathbb{R}^n$ denote a vector with $\mathbf{x}[k]$ and $\|\mathbf{x}\|_p$ ($p \in \{1, 2, \infty\}$) being its k -th element and vector p -norm. $\mathbf{M} \in \mathbb{R}^{n \times m}$ is a matrix with \mathbf{M}^\top , $\mathbf{M}[k]$ and $\mathbf{M}^\top[k]$ being its transpose, k -th row and k -th column, respectively. Similarly, $\|\mathbf{M}\|_p$ ($p \in \{1, 2, \infty\}$) denotes the induced matrix p -norm. In the sequel, all norms will be referred to as 2-norm if not explicitly specified. The set of positive integers up to n is \mathbb{Z}_n^+ , and \mathbb{Z}_j^i denotes non-negative integers from i to j ($0 \leq i \leq j$).

The minimum (maximum) eigenvalue of a matrix \mathbf{M} is denoted as $\lambda_{\min}(\mathbf{M})$ ($\lambda_{\max}(\mathbf{M})$). Given $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, \mathbf{x}/\mathbf{y} means element-wise division. The symbol \otimes denotes the Kronecker product. $\mathbf{0}_n$, $\mathbf{1}_n$, and \mathbf{I}_n are n -dimensional zero vector, one vector, and identity matrix, respectively. The n -dimensional basis row vector with the i -th entry being 1 while all other

entries being 0 is denoted as \mathbf{b}_i^n . Further, we define operators diag and vec for matrices (vectors) as:

$$\text{diag}\{M_i, M_j\} = \begin{bmatrix} M_i & \mathbf{0} \\ \mathbf{0} & M_j \end{bmatrix}, \quad \text{vec}\{M_i, M_j\} = \begin{bmatrix} M_i \\ M_j \end{bmatrix},$$

$$\text{diag}_{k=i}^j\{M_k\} = \begin{bmatrix} M_i & & \\ & \ddots & \\ & & M_j \end{bmatrix}, \quad \text{vec}_{k=i}^j\{M_k\} = \begin{bmatrix} M_i \\ \vdots \\ M_j \end{bmatrix}.$$

For convenience, the trigonometric functions $\sin(\cdot)$, $\cos(\cdot)$ and $\tan(\cdot)$ are abbreviated as $s(\cdot)$, $c(\cdot)$ and $t(\cdot)$, respectively. The Cholesky decomposition for matrices is denoted as a function $\mathcal{CH}(\cdot)$. Besides, the probability distribution function (PDF) of a random variable X obeying Gaussian distribution is denoted as $G(x; \mu, \sigma)$, with μ being the mean and σ being the standard deviation. For Gamma distribution, we use $\gamma(x; k, \varepsilon) = [\Gamma(k)]^{-1} \varepsilon^k x^{k-1} \exp(-\varepsilon x)$, where $\Gamma(\cdot)$ is the gamma function. Lastly, the uniform distribution on interval $[a, b]$ is denoted as $U(a, b)$.

B. Peer-to-peer Relative Motion Dynamics

We first introduce the relative motion dynamics for a pair of agents, which is the basic building block of the CRL models in later stages. For a given MAV, the 3 axes of the body-fixed body frame (\mathcal{F}_B) originate from the center of gravity of the robot with the X -axis pointing forward, Z -axis being aligned to the direction of the thrust, and Y -axis completing the frame according to the right-handed convention. In addition, to simplify the localization problem, we mainly deal with the body-centered *horizontal frame* (\mathcal{F}_H) with its Z -axis being always *perpendicular* to the ground plane [17]. With \mathcal{F}_B and \mathcal{F}_H defined, the orientation of both frames, as well as the transformation between the frames, can be determined using Euler angles (i.e., ϕ , θ , ψ) and the corresponding rotation matrices $\mathbf{R}_X(\phi)$, $\mathbf{R}_Y(\theta)$, and $\mathbf{R}_Z(\psi)$.

In this work, the relative position is resolved in the horizontal frame \mathcal{F}_H , and all related variables should be properly transformed accordingly. First, the velocity \mathbf{v} is measured in \mathcal{F}_B and it needs to be expressed in \mathcal{F}_H . The rotation matrix transforming any vector in \mathcal{F}_B to \mathcal{F}_H is given as

$${}^H\mathbf{R}_B = \mathbf{R}_Y(\theta)\mathbf{R}_X(\phi) = \begin{bmatrix} c(\theta) & s(\theta)s(\phi) & s(\theta)c(\phi) \\ 0 & c(\phi) & -s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix}. \quad (1)$$

Second, the gyroscope measures the instantaneous angular velocity $\boldsymbol{\Omega}$ in \mathcal{F}_B , to which the Euler angle rate should be properly related, and the formula is given as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi)/c(\theta) & c(\phi)/c(\theta) \end{bmatrix} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}. \quad (2)$$

In particular, we are interested in computing the heading rate $\dot{\psi}$, to which the changing of the horizontal frame is related. Note that the conversion in (1) and (2) requires the roll (ϕ) and pitch (θ), both of which can be accurately estimated by IMU.

For any agent pair (i, j) , the peer-to-peer RL problem is defined as estimating the relative position of agent j with respect to agent i in agent i 's horizontal frame \mathcal{F}_H^i . The state vector is thus given as $\mathbf{x}_{ij} = [\psi_{ij}, \mathbf{p}_{ij}^\top]^\top$, where ψ_{ij} is the relative heading and $\mathbf{p}_{ij} = [x_{ij}, y_{ij}, z_{ij}]^\top$ is the relative position in \mathcal{F}_H^i . The control input of any agent i is defined as $\mathbf{u}_i = [\dot{\psi}_i, \mathbf{v}_i^\top]^\top$ where $\mathbf{v}_i = [v_{x,i}, v_{y,i}, v_{z,i}]^\top$ is the velocity

input in the body frame \mathcal{F}_H^i of agent i , and the control input for agent pair (i, j) is $\mathbf{u}_{ij} = \text{vec}\{\mathbf{u}_i, \mathbf{u}_j\}$. Then, the relative motion dynamics for agent pair (i, j) can be derived using Newton's law as

$$\dot{\mathbf{x}}_{ij} = \mathbf{g}(\mathbf{x}_{ij}, \mathbf{u}_{ij}) = \begin{bmatrix} \dot{\psi}_j - \dot{\psi}_i \\ \mathbf{R}(\psi_{ij})\mathbf{v}_j - \mathbf{v}_i - \dot{\psi}_i S \mathbf{p}_{ij} \end{bmatrix}, \quad (3)$$

where the matrices $\mathbf{R}(\psi_{ij})$ and S are given as

$$\mathbf{R}(\psi_{ij}) = \begin{bmatrix} c(\psi_{ij}) & -s(\psi_{ij}) & 0 \\ s(\psi_{ij}) & c(\psi_{ij}) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

C. Heavy-tailed UWB Measurement Noise & Delay Effects

Ideally, the true distance d_{ij}^* between agent pair (i, j) is given in terms of its corresponding relative position \mathbf{p}_{ij}^* as

$$d_{ij}^* = \|\mathbf{p}_{ij}^*\|. \quad (4)$$

However, the measured distance deviates from the true value due to many practical factors. First, the distance measuring scheme in UWB is two-way ranging (TWR), whose noise model can be ideally assumed to be Gaussian considering only line-of-sight (LoS) scenarios. A more realistic noise model accounting for both multi-pathing and NLoS effects leads to a heavy-tailed distribution which is a linear combination of a Gaussian distribution and a Gamma distribution and other additional terms for outliers. We adopt the PDF of the simplified model [24] for the noise ν as

$$f(\nu) = \frac{1}{1 + s_{\text{ht}}} G(\nu; s_{\text{ht}}\mu, \sigma) + \frac{s_{\text{ht}}}{1 + s_{\text{ht}}} \gamma(\nu; k, \lambda), \quad (5)$$

where $s_{\text{ht}} \geq 0$ is a scale factor that is chosen based on the real measurement data.

Another source of measurement error comes from communication delay, especially for the indirect distance measurements between neighbors. Due to the limited bandwidth, the distances cannot be measured and transmitted simultaneously. Therefore, for agent i which can only get the distance between its neighbors j and l through the communication between agent j or l , the instantaneous true distance $\|\mathbf{p}_{jl}^*\|$ at the transmitting time t_{tr} differs from the transmitted $\|\mathbf{p}_{jl}\|$ measured at time t_{ms} due to the unknown motion of agent j and l during the time interval $[t_{\text{ms}}, t_{\text{tr}}]$. Assume the length of the time interval is upper bounded by $\bar{\eta}$, and the relative speed between any pair of agents is also bounded by \bar{v} , we have $\mathbf{p}_{jl}^* = \mathbf{p}_{jl} + \mathbf{p}_e = \mathbf{p}_{jl} + (\alpha\bar{\eta}\bar{v})\mathbf{i}$ with $\alpha \in [0, 1]$ and \mathbf{i} a unit vector. For modeling simplicity, we further assume that \mathbf{i} has an equal probability of pointing in any direction and α is uniformly distributed on $[0, 1]$. Then, \mathbf{p}_e has equal probability pointing at any point in a sphere of radius $\bar{r} = \bar{\eta}\bar{v}$. As a result, the probability of having a distance error $\nu_d = s - d_{jl}$ is proportional to the area of the spherical cap of a sphere of radius $s \in [d_{jl} - \bar{r}, d_{jl} + \bar{r}]$ intersected by a sphere with radius \bar{r} , where $d_{jl} = \|\mathbf{p}_{jl}\|$ is the distance between agent j and l with measurement delay. The area of the spherical cap can be approximated as

$$a(\nu_d; d_{jl}, \bar{r}) = \pi[-(\nu_d^2 + 2\nu_d d_{jl} - \bar{r}^2)^2 + 4d_{jl}^2 \bar{r}^2], \quad (6)$$

and the 2-D visualization of the cap is given in Fig. 2. Following (6), the normalized PDF of ν_d is thus given as

$$f(\nu_d) = \frac{15}{16\pi\bar{r}^3(5d_{jl}^2 - \bar{r}^2)} a(\nu_d; d_{jl}, \bar{r}), \quad (\nu_d \in [-\bar{r}, \bar{r}]). \quad (7)$$

The severity of the delay effects hinges on the velocity of the MAV and the density of the swarm. On the one hand, if

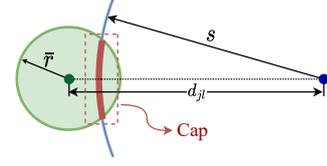


Fig. 2. 2-D visualization of the spherical cap which is marked in red.

the MAVs fly at high speed such that the relative velocity can be high enough, then \bar{v} can be large. On the other hand, if the swarm is too dense such that the number of neighbors of each agent is big, then $\bar{\eta}$ can be quite large due to limited bandwidth. In Fig. 3, we present some representative PDFs when $\bar{r} = 0.15[\text{m}]$.

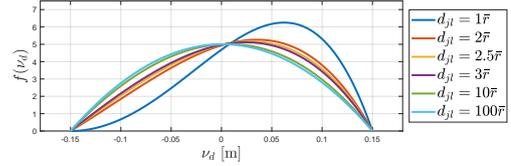


Fig. 3. PDFs of noise ν_d

Remark 1. The noise ν and ν_d are independent, which explains why they are considered separately instead of being combined into a single type of noise with a unified PDF.

Remark 2. The approximation in (6) is valid under the condition that $\bar{r} \ll d_{jl}$, which can be relaxed since that the noise distribution does not differ too much when $\bar{r} \leq 3d_{jl}$. In most cases, the condition $\bar{r} > d_{jl}$ holds given that the number of neighbors for any agent is limited such that $\bar{\eta}$ does not permit a large value and two MAVs do not fly in a dangerous mode in which the inter-agent distance is too short such that the random motion within a short time interval is likely to cause a collision (i.e., $d_{jl} \leq \bar{r}$).

III. COOPERATIVE RELATIVE LOCALIZATION

This section presents 3 different CRL models which serve as the main problem of interest in this work. Next, to shed light on the state estimation, we present the observability analysis for the built models using some elementary differential geometric tools.

A. Modeling of CRL Schemes

Having the peer-to-peer relative motion dynamics established as in (3), we are able to delve into the CRL modeling. Consider a swarm of N agent, and for agent i , the set of its neighbors is \mathcal{N}_i with $|\mathcal{N}_i| = N_i$. We explicitly express \mathcal{N}_i as $\{j_\alpha \in \mathbb{Z}_N^+ | j_\alpha \neq i, j_\alpha < j_{\alpha+1} (\alpha = 1, 2, \dots, N_i)\}$. To build an integrated dynamic model, we augment the state \mathbf{x}_{ij_α} of agent pair (i, j_α) , and the augmented state $\bar{\mathbf{x}}_i$ is given as

$$\bar{\mathbf{x}}_i = \text{vec}_{\alpha=1}^{N_i} \{\mathbf{x}_{ij_\alpha}\}. \quad (8)$$

We also augment the control input \mathbf{u}_i to form $\bar{\mathbf{u}}$ as

$$\bar{\mathbf{u}} = \text{vec}\{\mathbf{u}_i, \text{vec}_{\alpha=1}^{N_i} \{\mathbf{u}_{j_\alpha}\}\}. \quad (9)$$

Following (3), we can define $\bar{\mathbf{g}}_i^c(\bar{\mathbf{x}}_i, \bar{\mathbf{u}})$ similarly as

$$\bar{\mathbf{g}}_i^c(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}) = \text{vec}_{\alpha=1}^{N_i} \{\mathbf{g}(\mathbf{x}_{ij_\alpha}, \mathbf{u}_{j_\alpha})\}, \quad (10)$$

and the augmented system directly follows as

$$\dot{\bar{\mathbf{x}}}_i = \bar{\mathbf{g}}_i^c(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i). \quad (11)$$

For the measurement model of CRL, we first augment the direct distance model between agent i and each of its neighbors

$$\mathbf{h}_i^d(\bar{\mathbf{x}}_i) = \text{vec}_{\alpha=1}^{N_i} \{\|\mathbf{p}_{ij_\alpha}\|\}. \quad (12)$$

Then, the augmented direct measurement considering only the noise ν is given as

$$\mathbf{y}_i^d = \mathbf{h}_i^d(\bar{\mathbf{x}}_i) + \nu, \quad (13)$$

where $\nu = \text{vec}_{\alpha=1}^{N_i} \{\nu_\alpha\}$ is the augmented noise. On the other hand, the augmented indirect distances transmitted by agent j_α are given as

$$\mathbf{h}_i^{\text{id}}(\bar{\mathbf{x}}_i; j_\alpha) = \text{vec}_{l \in \mathcal{N}_i \cap \mathcal{N}_{j_\alpha}} \{\|\mathbf{p}_{lj_\alpha} - \mathbf{p}_{il}\|\}, \quad (14)$$

and the augmented indirect distances considering all its neighbors follows as

$$\mathbf{h}_i^{\text{id}}(\bar{\mathbf{x}}_i) = \text{vec}_{\alpha=1}^{N_i} \{\mathbf{h}_i^{\text{id}}(\bar{\mathbf{x}}_i; j_\alpha)\}. \quad (15)$$

The final noisy indirect measurement model is given as

$$\mathbf{y}_i^{\text{id}} = \mathbf{h}_i^{\text{id}}(\bar{\mathbf{x}}_i) + \nu_a \quad (16)$$

where $\nu_a = \text{vec}_{\beta=1}^{p_{\text{id}}} \{\nu_\beta + \nu_{d,\beta}\}$ with $p_{\text{id}} = \sum_{\alpha=1}^{N_i} |\mathcal{N}_i \cap \mathcal{N}_{j_\alpha}|$.

Moreover, we assume that all velocity inputs are imperfect due to actuator noise $\Delta \mathbf{u}_i$ for agent i , with which we also define $\Delta \mathbf{u}_{ij_\alpha} = \text{vec}\{\Delta \mathbf{u}_i, \Delta \mathbf{u}_{j_\alpha}\}$ and $\Delta \bar{\mathbf{u}}_i = \text{vec}\{\Delta \mathbf{u}_i, \text{vec}_{\alpha=1}^{N_i} \{\Delta \mathbf{u}_{j_\alpha}\}\}$. Referring to (11), (13) and (16), the full-CRL (fCRL) model with the augmented dynamics and both direct and indirect measurements is given as

$$\text{fCRL} : \begin{cases} \dot{\bar{\mathbf{x}}}_i = \bar{\mathbf{g}}_i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i + \Delta \bar{\mathbf{u}}_i) \\ \begin{bmatrix} \mathbf{y}_i^d \\ \mathbf{y}_i^{\text{id}} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_i^d(\bar{\mathbf{x}}_i) \\ \mathbf{h}_i^{\text{id}}(\bar{\mathbf{x}}_i) \end{bmatrix} + \begin{bmatrix} \nu \\ \nu_a \end{bmatrix} \end{cases} \quad (17)$$

For comparative study, we also introduce the following two models: i) the half-CRL (hCRL) model with the augmented dynamics and only direct measurements, ii) the non-CRL (nCRL) model which is simply a collection of the peer-to-peer dynamics in (3) for agent pairs (i, j_α) without state and input augmentation, $\forall j_\alpha \in \mathcal{N}_i$. The details are given below:

$$\text{hCRL} : \begin{cases} \dot{\bar{\mathbf{x}}}_i = \bar{\mathbf{g}}_i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i + \Delta \bar{\mathbf{u}}_i) \\ \mathbf{y}_i^d = \mathbf{h}_i^d(\bar{\mathbf{x}}_i) + \nu \end{cases}, \quad (18)$$

$$\text{nCRL} : \left\{ \begin{cases} \dot{\mathbf{x}}_{ij_\alpha} = \mathbf{g}(\mathbf{x}_{ij_\alpha}, \mathbf{u}_{ij_\alpha} + \Delta \mathbf{u}_{ij_\alpha}) \\ \mathbf{y}_{ij_\alpha} = \|\mathbf{p}_{ij_\alpha}\| + \nu_\alpha \end{cases} \right\}_{j_\alpha \in \mathcal{N}_i} \quad (19)$$

Remark 3. The 3 dynamic models (i.e., fCRL, hCRL, and nCRL) are all continuous-time nonlinear control systems. In addition, the built models are stochastic in nature due to actuator and measurement noise.

Remark 4. The nCRL model is equivalent to the peer-to-peer relative localization as investigated in [6], [17].

B. Knowledge from Differential Geometry

We present some basic geometric concepts following [25]. Consider a time-varying (TV) nonlinear control system

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \\ \mathbf{y} = [h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_p(\mathbf{x})]^\top, \end{cases} \quad (20)$$

where \mathbf{x} evolves in a \mathcal{C}^∞ - manifold \mathcal{M} of dimension n , $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input, $\mathbf{g}(\mathbf{x}, \mathbf{u}, t)$ is the TV control-coupled vector field in \mathcal{M} , and $h_i(\mathbf{x})$ are time-invariant (TI) scalar fields also defined on \mathcal{M} . Besides, all functions are \mathcal{C}^∞ (i.e., smooth/infinately differentiable) functions of their

arguments. The time t belongs to an open interval $\mathcal{I} \subseteq \mathbb{R}$, and we denote $\underline{\mathcal{M}} = \mathcal{I} \times \mathcal{M} \times \mathcal{U}$. To analyze the above system, we present some basic geometric tools.

Definition 1 (Lie Derivative). Given a scalar function $h(\mathbf{x}) : \mathcal{M} \mapsto \mathbb{R}$, and a vector field $\mathbf{g}(\mathbf{x}, \mathbf{u}, t) : \underline{\mathcal{M}} \mapsto \mathcal{M}$. The Lie derivative of the covector dh along the vector field $\mathbf{g}(\mathbf{x}, \mathbf{u}, t)$ is defined as

$$\mathcal{L}_{\mathbf{g}} dh = d(\mathcal{L}_{\mathbf{g}} h) = d(\langle \frac{\partial h}{\partial \mathbf{x}}, \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \rangle), \quad (21)$$

where dh , being the dual of $\mathbf{g}(\mathbf{x}, \mathbf{u}, t)$, is a special type of covector called exact differential, and $\langle \cdot, \cdot \rangle$ denotes the inner product of two vector fields. Further, a set of covectors dh_i can form a codistribution $\Xi = \text{span}\{dh_i\}$, and the Lie derivative of the codistribution is $\mathcal{L}_{\mathbf{g}} \Xi = \text{span}\{\mathcal{L}_{\mathbf{g}} \xi | \forall \xi \in \Xi\}$. Taking the Lie derivative on a codistribution is useful when recursively computing the observability codistribution, especially for systems with multiple output functions.

To analyze the time-varying system in (20), we introduce the augmented Lie derivative [26] for a covector dh as

$$\dot{\mathcal{L}}_{\mathbf{g}} dh = \mathcal{L}_{\mathbf{g}} dh + \frac{\partial dh}{\partial t}. \quad (22)$$

Besides, this new derivative permits a recursive operation as

$$\dot{\mathcal{L}}_{\mathbf{g}}^r dh = \dot{\mathcal{L}}_{\mathbf{g}} (\dot{\mathcal{L}}_{\mathbf{g}}^{r-1} dh), \quad (r \in \mathbb{Z}_+), \quad (23)$$

with the initial case being $\dot{\mathcal{L}}_{\mathbf{g}}^0 dh := dh$. Likewise, the augmented Lie derivative can also act upon codistributions. The above geometric tools will be used for the observability analysis by checking the observability rank condition [25]. Specifically, the observability codistribution Ξ^* can be recursively computed according to Algorithm 1. Moreover, we

Algorithm 1: Computation of the Observability Codistribution

Data: $h_i, \mathbf{g}(\mathbf{x}, \mathbf{u}, t)$
Result: Ξ_i^*
1 $k \leftarrow 0, ID \leftarrow 1$;
2 $\Xi^{(k)} = \text{span}\{dh_1, dh_2, \dots, dh_p\}$;
3 **while** $ID = 1$ **do**
4 $\Xi^{(k+1)} = \text{span}\{\Xi^{(k)}, \dot{\mathcal{L}}_{\mathbf{g}} \Xi^{(k)}\}$;
5 **if** $\Xi^{(k+1)} = \Xi^{(k)}$ **then**
6 $ID \leftarrow 0$
7 **end**
8 $k \leftarrow k + 1$
9 **end**
10 $\Xi_i^* \leftarrow \Xi_i^{(k)}$

have $\Xi^{(k)} \subseteq \Xi^{(k+1)}$, and the final converged codistribution is obtained for some $k^* \leq n - 1$ [25], [26]. Checking the observability rank condition is equivalent to checking the rank of the observability matrix $\mathcal{OB}^{(k)}$ derived from the codistribution $\Xi^{(k)}$, and $\mathcal{OB}^{(k)}$ is given as

$$\mathcal{OB}^{(k)} = \begin{bmatrix} \text{vec}_{i=1}^p \{\dot{\mathcal{L}}_{\mathbf{g}}^0 dh_i\} \\ \text{vec}_{i=1}^p \{\dot{\mathcal{L}}_{\mathbf{g}}^1 dh_i\} \\ \vdots \\ \text{vec}_{i=1}^p \{\dot{\mathcal{L}}_{\mathbf{g}}^k dh_i\} \end{bmatrix}, \quad (24)$$

where we only refer to the coefficients of the bases $\{dx_1, dx_2, \dots, dx_n\}$ of the covector field while the bases themselves are eliminated with slight abuse of notation. If $\text{rank}(\mathcal{OB}^{(k)}) = n$ for some k at (\mathbf{x}^*, t^*) with control being

\mathbf{u}^* , then the system is *weakly locally observable* at (\mathbf{x}^*, t^*) [27] with control being \mathbf{u}^* .

Remark 5. In this work, we focus on active state estimation where we also care how control inputs affect the observability of the system. Besides, the controls \mathbf{u} are treated as constant parameters in that we check observability only locally (i.e., we assume that the controls remain constant within any small time interval). In addition, though the controls \mathbf{u} are treated as parameters, the space we investigate is \mathcal{M} in terms of active state estimation, and we aim to identify possible observable and unobservable subspaces through observability analysis.

Remark 6. The observability analysis only helps to determine whether the internal state of the system as in (20) can be well reconstructed (i.e., the estimation error is exponentially bounded in means square and bounded with probability 1 [28]) by the filter that can be viewed as a stochastic realization (e.g., the Kalman filter and its variations) [29], whereas the performance of the filter (i.e., estimation accuracy, transient behavior) still depends on many other factors.

C. Observability Analysis

For the basic peer-to-peer localization model, we only have a single direct distance measurement. Noting the dynamics of the relative heading ψ_{ij} is independent of the position \mathbf{p}_{ij} (cf. 11), we thus only consider the position dynamics given that ψ_{ij} is also not our state variable of interest. After decoupling ψ_{ij} , the position dynamics should be treated as a TV nonlinear system with $\psi_{ij}(t)$ being the varying parameter. Following (3), the resulting peer-to-peer relative dynamics and augmented dynamics are given in (25) and (26), respectively.

$$\begin{cases} \mathbf{p}_{ij} = \mathbf{g}(\mathbf{p}_{ij}, \mathbf{u}_{ij}, t) = \mathbf{R}(\psi_{ij}(t))\mathbf{v}_j - \mathbf{v}_i - \dot{\psi}_{ij}S\mathbf{p}_{ij} \\ y = h(\mathbf{p}_{ij}) = \|\mathbf{p}_{ij}\| \end{cases}, \quad (25)$$

$$\begin{cases} \bar{\mathbf{p}}_i = \bar{\mathbf{g}}(\bar{\mathbf{p}}_i, \bar{\mathbf{u}}_i, t) \\ \mathbf{y}_i^d = \mathbf{h}_i^d(\bar{\mathbf{p}}_i) \\ (\mathbf{y}_i^{\text{id}} = \mathbf{h}_i^{\text{id}}(\bar{\mathbf{p}}_i)) \end{cases}, \quad (26)$$

where $\bar{\mathbf{p}}_i = \text{vec}_{\alpha=1}^{N_i}\{\mathbf{p}_{ij_\alpha}\}$, $\bar{\mathbf{u}}_i$ is given as in (9), $\bar{\mathbf{g}}(\bar{\mathbf{p}}_i, \bar{\mathbf{u}}_i, t) = \text{vec}_{\alpha=1}^{N_i}\{\mathbf{g}(\mathbf{p}_{ij_\alpha}, \mathbf{u}_{ij_\alpha}, t)\}$, and \mathbf{y}_i^d and \mathbf{y}_i^{id} are constructed similarly as in (17) (with details given in (12), (14) and (15)). We start with analyzing the peer-to-peer system given in (25), and the observability matrix $\mathcal{OB}^d(i, j)$ is given as

$$\mathcal{OB}^d(i, j) = \begin{bmatrix} \mathbf{p}_{ij}^\top \\ \frac{[\mathbf{R}(\psi_{ij}(t))\mathbf{v}_j - \mathbf{v}_i]^\top}{\dot{\psi}_{ij}\mathbf{v}_j^\top E - [\dot{\psi}_{ij}\mathbf{R}(\psi_{ij}(t))\mathbf{v}_j - \dot{\psi}_{ij}\mathbf{v}_i]^\top S} \end{bmatrix}, \quad (27)$$

where $E = \text{diag}\{0, 0, 1\}$, S is given as in (3), and all entries are multiplied by the non-zero term $\|\mathbf{p}_{ij}\|$ (i.e., two agents do not collide), which does not affect the rank of $\mathcal{OB}^d(i, j)$. Explicitly rewriting the above matrix lead to a more detailed expression of $\mathcal{OB}^d(i, j)$ as

$$\begin{bmatrix} x_{ij} & y_{ij} & z_{ij} \\ c(\psi_{ij})v_{x,j} - s(\psi_{ij})v_{y,j} - v_{x,i} & s(\psi_{ij})v_{x,j} + c(\psi_{ij})v_{y,j} - v_{y,i} & v_{z,j} - v_{z,i} \\ -\dot{\psi}_j s(\psi_{ij})v_{x,j} - \dot{\psi}_j c(\psi_{ij})v_{y,j} + \dot{\psi}_i v_{y,i} & \dot{\psi}_j c(\psi_{ij})v_{x,j} - \dot{\psi}_j s(\psi_{ij})v_{y,j} - \dot{\psi}_i v_{x,i} & v_{z,j}(\dot{\psi}_j - \dot{\psi}_i) \end{bmatrix}.$$

We further denote the relative velocity $\mathbf{R}(\psi_{ij}(t))\mathbf{v}_j - \mathbf{v}_i$ as $\mathbf{v}_{ij} = [v_{x,ij}, v_{y,ij}, v_{z,ij}]^\top$. Though a complete analysis of the observability is unattainable since finding all conditions under which the determinant of the observability matrix is 0 is difficult, we can still find some *unobservable* subspaces which have clear physical interpretation:

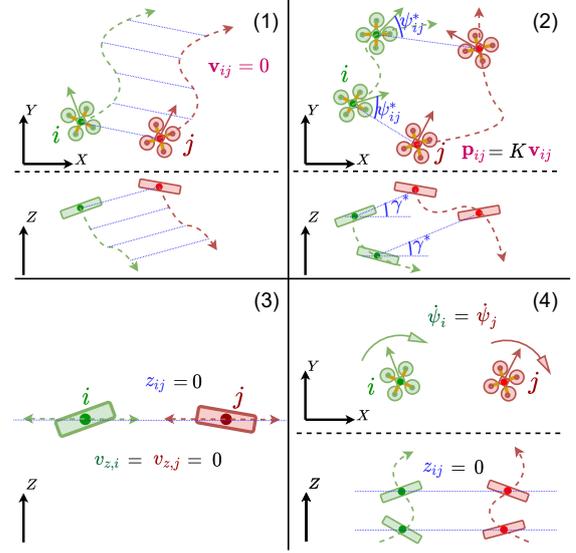


Fig. 4. Unobservable motions: case (1) (top left); case (2) (top right); case (3) (bottom left); case (4) (bottom right).

- 1) Relative velocity is zero (i.e., $\mathbf{v}_{ij} = \mathbf{0}$, **parallel motion**): $\mathcal{OB}^d(i, j)[2] = \mathbf{0}$.
- 2) Relative velocity is **aligned** with the relative position (i.e., the **relative motion is linear**): $\mathcal{OB}^d(i, j)[1] = K\mathcal{OB}^d(i, j)[2](K \neq 0)$.
- 3) Two agents move in a **fixed horizontal plane** ($z_{ij} = 0$), plus that the two agents i and j **do not have vertical velocity** ($v_{z,i} = v_{z,j} = 0$): $\mathcal{OB}^d(i, j)^\top[3] = \mathbf{0}$.
- 4) Two agents move in a **fixed horizontal plane** ($z_{ij} = 0$), plus that the two agents have the **same heading rate** ($\dot{\psi}_j - \dot{\psi}_i = 0$): $\mathcal{OB}^d(i, j)^\top[3] = \mathbf{0}$.

The above observability analysis directly suits the nCRL model due to its dynamics being fully decoupled for each agent pair $(i, j_\alpha), j_\alpha \in \mathcal{N}_i$.

Regarding the hCRL model, though \mathbf{u}_i perturbs the dynamics of each agent pair in an integrated way, both the dynamics $\mathbf{g}^c(\mathbf{p}_{ij_\alpha}, \mathbf{u}_{ij_\alpha}, t)$ and the measurement model $h_{ij_\alpha} = \|\mathbf{p}_{ij_\alpha}\|$ are decoupled in terms of the state $\mathbf{x}_{ij}(\mathbf{p}_{ij})$. Correspondingly, the resulting observability matrix for the hCRL has a *diagonal* structure as $\mathcal{OB}_{\text{hCRL}} = \text{diag}_{\alpha=1}^{N_i}\{\mathcal{OB}^d(i, j_\alpha)\}$ after some row permutations, from which we can easily conclude that the observability for the hCRL model can be analyzed in a *pair-wise* manner, making it the same as that for the nCRL model.

For fCRL model where indirect measurements are also present, the observable subspace can be expanded. Particularly, if the number of indirect measurements $p_{id} \geq 2N_i$ (i.e., each neighbor provides at least 2 indirect measurements on average and these measurements do not coincide), the total number of measurements (i.e., counting both direct and indirect ones) will be greater than the number of interested states $3N_i$, then the fCRL system is *fully observable* since $\mathcal{OB}_{\text{fCRL}}^{(0)} = \text{vec}_{i=1}^{3N_i}\{dh_i\}$ is already of full rank given that the agents do not collide.

The minimum number of neighboring agents that can achieve fully observable is $N_i = 5$. In generic cases where $p_{id} < 2N_i$, more detailed investigation is required, and we start with the simplest case where we deal with the 3 agent tuple (i, j_1, j_2) without loss of generality. After some row permutations, the observability matrix $\mathcal{OB}(i, j_1, j_2)$ is expressed as

$$\mathcal{OB}(i, j_1, j_2) = \begin{bmatrix} \frac{\mathcal{OB}^d(i, j_1)}{\|\mathbf{p}_{ij_1}\|} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \frac{\mathcal{OB}^d(i, j_2)}{\|\mathbf{p}_{ij_2}\|} \\ \frac{\mathcal{OB}^{id}(j_1, j_2)}{\|\mathbf{p}_{ij_1} - \mathbf{p}_{ij_2}\|} & -\frac{\mathcal{OB}^{id}(j_1, j_2)}{\|\mathbf{p}_{ij_1} - \mathbf{p}_{ij_2}\|} \end{bmatrix}, \quad (28)$$

where the partial observability matrix $\mathcal{OB}^{id}(j_1, j_2)$ corresponding to the indirect measurement $h(\mathbf{p}_{ij_1}, \mathbf{p}_{ij_2}) = \|\mathbf{p}_{ij_1} - \mathbf{p}_{ij_2}\|$ is given as

$$\mathcal{OB}^{id}(j_1, j_2) = \begin{bmatrix} \mathbf{p}_{ij_1}^\top - \mathbf{p}_{ij_2}^\top \\ [R(\psi_{ij_1})\mathbf{v}_{j_1} - R(\psi_{ij_2})\mathbf{v}_{j_2}]^\top \\ (\dot{\psi}_{ij_1}\mathbf{v}_{j_1} - \dot{\psi}_{ij_2}\mathbf{v}_{j_2})^\top E \\ -[\dot{\psi}_{ij_1}R(\psi_{ij_1})\mathbf{v}_{j_1} - \dot{\psi}_{ij_2}R(\psi_{ij_2})\mathbf{v}_{j_2}]^\top S \end{bmatrix}, \quad (29)$$

We then discuss how the observable subspace can be expanded by imposing extra stringent conditions on the previously identified unobservable subspace. In the following analysis, we only discuss the cases where the previously identified 4 unobservable conditions hold independently, whereas the cases in which multiple conditions are satisfied simultaneously will not be further investigated. In the following, we will frequently use the term *scaled velocity* which means the velocity scaled by the heading rate (e.g., $\dot{\psi}_j\mathbf{v}_j$ and $\dot{\psi}_j\mathbf{R}(\psi_{ij})\mathbf{v}_j$), and the scaled velocity terms are present in the observability subblocks (cf. (27) and (29)).

1) Parallel motion

- If either (i, j_1) or (i, j_2) is in parallel motion, then the system is always observable in that $\mathcal{OB}^{id}(j_1, j_2)[1] \neq 0$ always holds true.
- If both (i, j_1) and (i, j_2) are in parallel motion (i.e., all 3 agents are in parallel motion), then the system is still observable if the relative scaled velocity of pair (j_1, j_2) is non-zero (i.e., $\mathcal{OB}^{id}(j_1, j_2)[3] \neq 0$)

2) Relative velocity aligned with the relative position

- If either (i, j_1) or (i, j_2) falls in this aligning condition, then the system is always observable in that $\mathcal{OB}^{id}(j_1, j_2)[1] \neq 0$ always holds true.
- If both pair (i, j_1) and (i, j_2) satisfy this aligning condition, then the system is still observable if pair (j_1, j_2) does not meet this condition. Further, even if (j_1, j_2) also satisfies this aligning condition, then the system is still observable if the relative scaled velocity of pair (j_1, j_2) is non-zero (i.e., $\mathcal{OB}^{id}(j_1, j_2)[3] \neq 0$).

3) Fixed horizontal plane plus zero vertical velocity

If at least one of the two pairs (i, j_1) and (i, j_2) satisfies the condition that i and j_1 (j_2) moves in the same horizontal plane and j_1 (j_2) does not have vertical velocity, then the system is still observable if any one of the below conditions holds: i) j_1 and j_2 have altitude difference ii) j_1 and j_2 have different vertical velocity iii) the relative scaled vertical velocity of pair (j_1, j_2) is non-zero (i.e., $\mathcal{OB}^{id}(j_1, j_2)[3] \neq 0$).

For the general fCRL model of agent i and its neighbor set \mathcal{N}_i , its observability matrix $\mathcal{OB}(i, \mathcal{N}_i)$ is given as

$$\begin{bmatrix} d\mathbf{p}_{ij_1} & d\mathbf{p}_{ij_2} & \cdots & d\mathbf{p}_{ij_{N_i}} \\ \frac{\mathcal{OB}^d(i, j_1)}{\|\mathbf{p}_{ij_1}\|} & \mathbf{0}_{3 \times 3} & \cdots & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \frac{\mathcal{OB}^d(i, j_2)}{\|\mathbf{p}_{ij_2}\|} & \cdots & \mathbf{0}_{3 \times 3} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{3 \times 3} & \cdots & \cdots & \frac{\mathcal{OB}^d(i, j_{N_i})}{\|\mathbf{p}_{ij_{N_i}}\|} \\ \hline \frac{\mathcal{OB}^{id}(j_1, j_2)}{\|\mathbf{p}_{ij_1} - \mathbf{p}_{ij_2}\|} & -\frac{\mathcal{OB}^{id}(j_1, j_2)}{\|\mathbf{p}_{ij_1} - \mathbf{p}_{ij_2}\|} & \cdots & \mathbf{0}_{3 \times 3} \\ \frac{\mathcal{OB}^{id}(j_1, j_{N_i})}{\|\mathbf{p}_{ij_1} - \mathbf{p}_{ij_{N_i}}\|} & \mathbf{0}_{3 \times 3} & \cdots & -\frac{\mathcal{OB}^{id}(j_1, j_{N_i})}{\|\mathbf{p}_{ij_1} - \mathbf{p}_{ij_{N_i}}\|} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} d_{ij_1} \\ d_{ij_2} \\ \vdots \\ d_{ij_{N_i}} \\ d_{j_1 j_2} \\ d_{j_1 j_{N_i}} \\ \vdots \end{bmatrix},$$

where the upper part with only diagonal subblocks is fixed and the lower part depends on the type of indirect measurements. Specifically, the direct measurement d_{ij_α} and the indirect measurement $d_{j_\alpha j_\beta}$ contribute, respectively, to the following two blocks:

$$\begin{cases} d_{ij_\alpha} : \mathbf{b}_\alpha^{N_i} \otimes \frac{\mathcal{OB}^{id}(j_i, j_\alpha)}{\|\mathbf{p}_{ij_\alpha}\|} \\ d_{j_\alpha j_\beta} : (\mathbf{b}_\alpha^{N_i} - \mathbf{b}_\beta^{N_i}) \otimes \frac{\mathcal{OB}^{id}(j_\alpha, j_\beta)}{\|\mathbf{p}_{ij_\alpha} - \mathbf{p}_{ij_\beta}\|} \end{cases}. \quad (30)$$

Based on the special structure of the observability matrix, the general rule of thumb is that the more indirect measurements that agent j_α is coupled, the more the unobservable subspace corresponding to the state \mathbf{p}_{ij_α} diminishes.

IV. FILTERING BASED STATE ESTIMATION

This section provides the nonlinear filtering solution to the state estimation problem in CRL. We will give an introduction to the basic extended Kalman filter (EKF), with which we develop the MLVC-EKF and apply it to the CRL models. The convergence analysis of the fixed-point iteration involved in the posterior estimation will also be presented. Finally, we analyze the computational complexity of all 3 CRL schemes applied with both MLVC-EKF and the EKF.

A. Extended Kalman Filter

We first summarize the basic extended Kalman filter algorithm. To better serve our built CRL models, we consider a continuous-time stochastic nonlinear system of the form

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{w}) \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\nu} \end{cases}, \quad (31)$$

where $\mathbf{x} \in \mathcal{M} \subseteq \mathbb{R}^n$ is the state, $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^p$ is the measurement vector, $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input, and $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^m$ and $\boldsymbol{\nu} \in \mathcal{V} \subseteq \mathbb{R}^p$ are process noise and measurement noise, respectively. The system evolves on an open time interval \mathcal{I} . With sampling time T_s , the continuous-time system can be discretized using the Euler method and evaluated at $t_0 + kT_s$ ($k \in \mathbb{Z}_{k_m}^0$) such that $\{t_0 + kT_s\}_{k=0}^{k_m} \subset \mathcal{I}$. We simply use k as the index, and the corresponding discrete-time system is given as

$$\begin{cases} \mathbf{x}_k = \mathbf{x}_{k-1} + T_s \mathbf{g}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\nu}_k \end{cases}. \quad (32)$$

We then provide the EKF *recursion* for the discretized system as in (32). Given the corrected estimated state $\hat{\mathbf{x}}_{k-1}^+$ as well as its covariance \mathbf{P}_{k-1}^+ at time instant $k-1$ ($k \geq 1$), we then proceed to compute the posterior estimation $\hat{\mathbf{x}}_k^+$ and covariance \mathbf{P}_k^+ in two steps.

1) Prediction: Given the control input \mathbf{u}_{k-1} , compute the prior estimate $\hat{\mathbf{x}}_k^-$ and its covariance \mathbf{P}_k^- as

$$\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k-1}^+ + T_s \mathbf{g}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, \mathbf{0}) \quad (33a)$$

$$\mathbf{P}_k^- = \mathbf{A}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{A}_{k-1}^\top + \mathbf{B}_{k-1} \mathbf{Q}_{k-1} \mathbf{B}_{k-1}^\top, \quad (33b)$$

where $\mathbf{Q}_{k-1} = \hat{\mathbb{E}}[\mathbf{w}_{k-1} \mathbf{w}_{k-1}^\top]$ is the estimated process noise covariance matrix, and \mathbf{A}_{k-1} and \mathbf{B}_{k-1} are given as

$$\mathbf{A}_{k-1} = \mathbb{I}_n + T_s \left(\left. \frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{x}} \right|_{(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, \mathbf{0})} \right), \quad (34a)$$

$$\mathbf{B}_{k-1} = T_s \left(\left. \frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{u}} \right|_{(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, \mathbf{0})} \right). \quad (34b)$$

2) Correction by measurements: Correct the prior estimation based on the measurement \mathbf{y}_k , which outputs the *posterior* estimation $\hat{\mathbf{x}}_k^+$ and its covariance \mathbf{P}_k^+ . First, approximate \mathbf{y}_k using first-order linearization as

$$\mathbf{y}_k \approx \mathbf{h}(\hat{\mathbf{x}}_k^-) + \mathbf{H}_k (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \boldsymbol{\nu}_k, \quad (35)$$

where \mathbf{H}_k is given as

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^-}. \quad (36)$$

Then the Kalman update follows as

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)), \quad (37a)$$

$$\mathbf{P}_k^+ = (\mathbb{I}_n - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbb{I}_n - \mathbf{K}_k \mathbf{H}_k)^\top + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^\top, \quad (37b)$$

where $\mathbf{R}_k = \hat{\mathbb{E}}[\boldsymbol{\nu}_k \boldsymbol{\nu}_k^\top]$ is the estimated measurement noise covariance matrix, and the Kalman gain \mathbf{K}_k is given as

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^\top (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\top + \mathbf{R}_k)^{-1}. \quad (38)$$

B. Kernel-induced Extended Kalman Filter

To cope with heavy-tailed non-Gaussian measurement noise, we use correntropy as a new metric to define the estimation error. In information theoretical learning, correntropy measures the similarity between two random variables $X, Y \in \mathbb{R}$ as

$$C(X, Y) = \mathbb{E}[\kappa(X, Y)] = \int \kappa(x, y) dF_{XY}(x, y), \quad (39)$$

where $F_{XY}(x, y)$ is the joint distribution function and $\kappa(\cdot, \cdot)$ is a shift-invariant Mercer kernel [30]. In this work, inspired by the Versoria function [31], we design a novel Logarithmic-Versoria (LV) kernel as

$$\kappa(x, y) = L_\tau(e) = \frac{\tau}{\tau + \ln(1 + e^2)}, \quad (40)$$

where $e = x - y$ is the error, and $\tau > 0$ controls the *bandwidth* of the kernel. The Versoria function can exploit the non-Gaussian characteristics and provide better higher-order error information than the Gaussian kernel [31]. The newly designed LV kernel not only inherits the power of Versoria function but also exhibits smoother behavior for small errors. A brief visualization of different kernel functions is given in Figure 5, where we compare the Gaussian kernel function, the Versoria kernel function, and the Logarithmic-Versoria function all with the same bandwidth $\tau = 5$ (cf. (60b), (60a), and (40)).

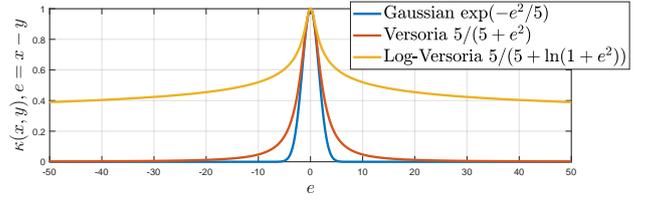


Fig. 5. Kernel Function Comparison

For the filtering problem of general stochastic systems where the analytical distribution of the state(output) is not known, it is practical to approximate the correntropy using the sampled-mean strategy as

$$\hat{C}(X, Y) = \frac{1}{D} \sum_{i=1}^D L_\tau(e(i)), \quad (41)$$

where $e(i) = x(i) - y(i)$ with $\{x(i), y(i)\}$ being the candidate data points.

The MLVC-EKF *recursion* also has the two-step structure as the standard EKF, and it only differs from the EKF in the state correction step whereas the prediction step exactly follows (33a). The details of the correction step in the MLVC-EKF is presented as follows.

2) Kernel-induced correction: Based on the linearized measurement equation in (36), we then leverage both the state error $\mathbf{e}_{\mathbf{x},k} := \hat{\mathbf{x}}_k^- - \mathbf{x}_k$ and the measurement noise $\boldsymbol{\nu}_k$ to obtain

$$\underbrace{\begin{bmatrix} \hat{\mathbf{x}}_k^- \\ \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-) + \mathbf{H}_k \hat{\mathbf{x}}_k^- \end{bmatrix}}_{:= \mathbf{z}_k} = \underbrace{\begin{bmatrix} \mathbb{I}_n \\ \mathbf{H}_k \end{bmatrix}}_{:= \mathbf{F}_k} \mathbf{x}_k + \underbrace{\begin{bmatrix} \mathbf{e}_{\mathbf{x},k} \\ \boldsymbol{\nu}_k \end{bmatrix}}_{:= \boldsymbol{\delta}_k}. \quad (42)$$

To normalize the error $\boldsymbol{\delta}_k$, we perform Cholesky decomposition to obtain $\mathbf{M}_{\mathbf{x},k} = \mathcal{CH}(\mathbf{P}_k^-)$ and $\mathbf{M}_{\mathbf{y},k} = \mathcal{CH}(\mathbf{R}_k)$. We further define $\mathbf{M}_k = \text{diag}\{\mathbf{M}_{\mathbf{x},k}, \mathbf{M}_{\mathbf{y},k}\}$, then (42) can be modified as

$$\underbrace{(\mathbf{M}_k)^{-1} \mathbf{z}_k}_{:= \mathbf{z}_k^*} = \underbrace{(\mathbf{M}_k)^{-1} \mathbf{F}_k}_{:= \mathbf{F}_k^*} \mathbf{x}_k + (\mathbf{M}_k)^{-1} \boldsymbol{\delta}_k. \quad (43)$$

For the regression model given in (43), we incorporate the Versoria loss function (cf. (41)) as

$$J(\mathbf{x}_k) = \frac{1}{D} \sum_{i=1}^D L_\tau \left(\underbrace{\mathbf{z}_k^*[i] - \mathbf{F}_k^*[i, :] \mathbf{x}_k}_{:= \mathbf{e}_k[i]} \right), \quad (44)$$

where $D := n + p$ is the dimension of the regression problem, with which the posterior estimation $\hat{\mathbf{x}}_k^+$ can be computed by solving the following optimization problem:

$$\hat{\mathbf{x}}_k^+ = \arg \max_{\mathbf{x}_k} J(\mathbf{x}_k). \quad (45)$$

The optimality of (45) is guaranteed by $\partial J(\mathbf{x}_k) / \partial \mathbf{x}_k = 0$, which suggests that $\hat{\mathbf{x}}_k^+$ satisfies the following equation:

$$\hat{\mathbf{x}}_k^+ = (\mathbf{F}_k^{*\top} \mathbf{L}_k \mathbf{F}_k^*)^{-1} \mathbf{F}_k^{*\top} \mathbf{L}_k \mathbf{z}_k^*, \quad (46)$$

where $\mathbf{L}_k = \text{diag}\{\mathbf{L}_{\mathbf{x},k}, \mathbf{L}_{\mathbf{y},k}\}$ with $\mathbf{L}_{\mathbf{x},k}$ and $\mathbf{L}_{\mathbf{y},k}$ being constructed as follows:

$$\begin{cases} \mathbf{L}_{\mathbf{x},k} = \text{diag}_{i=1}^n \{L_\tau^2(\hat{\mathbf{e}}_k[i]) (1 + \hat{\mathbf{e}}_k^2[i])^{-1}\} \\ \mathbf{L}_{\mathbf{y},k} = \text{diag}_{i=n+1}^D \{L_\tau^2(\hat{\mathbf{e}}_k[i]) (1 + \hat{\mathbf{e}}_k^2[i])^{-1}\} \end{cases}. \quad (47)$$

In (47), we have $\hat{\mathbf{e}}_k[i] = \mathbf{z}_k^*[i] - \mathbf{F}_k^*[i, :] \hat{\mathbf{x}}_k^+$. Moreover, (46) can be rewritten as a Kalman-type state update [23]

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)), \quad (48)$$

where the Kalman gain \mathbf{K}_k is given as

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{L},k}^- \mathbf{H}_k^\top (\mathbf{H}_k \mathbf{P}_{\mathbf{L},k}^- \mathbf{H}_k^\top + \mathbf{R}_{\mathbf{L},k})^{-1}, \quad (49)$$

where the *kernel-weighted* covariance matrices $\mathbf{P}_{\mathbf{L},k}^-$ and $\mathbf{R}_{\mathbf{L},k}$ are given as

$$\begin{cases} \mathbf{P}_{\mathbf{L},k}^- = \mathbf{M}_{\mathbf{x},k} \mathbf{L}_{\mathbf{x},k}^{-1} \mathbf{M}_{\mathbf{x},k}^\top \\ \mathbf{R}_{\mathbf{L},k} = \mathbf{M}_{\mathbf{y},k} \mathbf{L}_{\mathbf{y},k}^{-1} \mathbf{M}_{\mathbf{y},k}^\top \end{cases} \quad (50)$$

Since the right hand side of (48) implicitly depends on $\hat{\mathbf{x}}_k^+$, computing $\hat{\mathbf{x}}_k^+$ requires the *fixed-point* algorithm. The converged output of the fixed-point iteration will serve as the posterior estimation, and the following final step is the Kalman-type covariance update as in (37b). The overall MLVC-EKF is summarized in Algorithm 2.

Algorithm 2: MLVC-EKF State Estimation

Data: $\hat{\mathbf{x}}_{k-1}^+$, \mathbf{P}_{k-1}^+ , \mathbf{u}_{k-1} , \mathbf{y}_k , \mathbf{Q}_k , \mathbf{R}_k , T_s
Result: $\hat{\mathbf{x}}_k^+$, \mathbf{P}_k^+

- 1 Compute \mathbf{A}_{k-1} , \mathbf{B}_{k-1} (cf. (34));
- 2 Compute $\hat{\mathbf{x}}_k^-$, \mathbf{P}_k^- (cf. (33)), \mathbf{H}_k (cf. (36));
- 3 $\mathbf{M}_{\mathbf{x},k} = \mathcal{CH}(\mathbf{P}_k^-)$, $\mathbf{M}_{\mathbf{y},k} = \mathcal{CH}(\mathbf{R}_k)$;
- 4 $\text{Tag}_{\text{FP}} \leftarrow 1$, $t \leftarrow 0$, $\mathbf{x}_k^{(0)} \leftarrow \hat{\mathbf{x}}_k^-$, $\iota = n + 1$;
- 5 **while** Tag_{FP} **do**
- 6 Build \mathbf{z}_k^* , \mathbf{F}_k^* (cf. (42), (43));
- 7 $\mathbf{e}_k^{(t)} = \mathbf{z}_k^* - \mathbf{F}_k^* \mathbf{x}_k^{(t)}$;
- 8 $\mathbf{L}_{\mathbf{x},k}^{(t)} = \text{diag}_{i=1}^n \{L_\tau^2(\hat{\mathbf{e}}_k^{(t)}[i])[1 + (\hat{\mathbf{e}}_k^{(t)}[i])^2]^{-1}\}$;
- 9 $\mathbf{P}_{\mathbf{L},k}^{(t)} = \mathbf{M}_{\mathbf{x},k} \mathbf{L}_{\mathbf{x},k}^{(t)-1} \mathbf{M}_{\mathbf{x},k}^\top$;
- 10 $\mathbf{L}_{\mathbf{y},k}^{(t)} = \text{diag}_{i=1}^D \{L_\tau^2(\hat{\mathbf{e}}_k^{(t)}[i])[1 + (\hat{\mathbf{e}}_k^{(t)}[i])^2]^{-1}\}$;
- 11 $\mathbf{R}_{\mathbf{L},k}^{(t)} = \mathbf{M}_{\mathbf{y},k} \mathbf{L}_{\mathbf{y},k}^{(t)-1} \mathbf{M}_{\mathbf{y},k}^\top$;
- 12 $\mathbf{K}_k^{(t)} = \mathbf{P}_{\mathbf{L},k}^{(t)} \mathbf{H}_k^\top (\mathbf{H}_k \mathbf{P}_{\mathbf{L},k}^{(t)} \mathbf{H}_k^\top + \mathbf{R}_{\mathbf{L},k}^{(t)})^{-1}$;
- 13 $\mathbf{x}_k^{(t+1)} = \hat{\mathbf{x}}_k^- + \mathbf{K}_k^{(t)} (\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-))$;
- 14 $r = \|(\mathbf{x}_k^{(t+1)} - \mathbf{x}_k^{(t)}) / \mathbf{x}_k^{(t)}\|$;
- 15 **if** $r \leq \epsilon$ **then**
- 16 $\text{Tag}_{\text{FP}} \leftarrow 0$, $\hat{\mathbf{x}}_k^+ \leftarrow \mathbf{x}_k^{(t+1)}$, $\mathbf{K}_k^* \leftarrow \mathbf{K}_k^{(t)}$
- 17 **end**
- 18 **end**

19 $\mathbf{P}_k^+ = (\mathbb{I}_n - \mathbf{K}_k^* \mathbf{H}_k) \mathbf{P}_k^- (\mathbb{I}_n - \mathbf{K}_k^* \mathbf{H}_k)^\top + \mathbf{K}_k^* \mathbf{R}_k \mathbf{K}_k^{*\top}$

Remark 7. The covariance update at the end of the MLVC-EKF algorithm adopts the Joseph form to guarantee symmetry as well as positive definiteness of the covariance matrix in numerical computation, which is necessary for performing the Cholesky decomposition.

Remark 8. The fixed-point iteration works with temporary state variable $\mathbf{x}_k^{(t)}$, and it terminates when the norm of the incremental percentile $((\mathbf{x}_k^{(t+1)} - \mathbf{x}_k^{(t)}) / \mathbf{x}_k^{(t)})$ is smaller than a preset threshold ϵ . A smaller value of ϵ leads to a more accurate estimation but may increase the required number of fixed-point iterations. In practice, we normally choose ϵ within the range $[10^{-6}, 10^{-3}]$.

Having developed the MLVC-EKF state estimation algorithm, we now investigate the application details of CRL. Due to the limited communication bandwidth, each agent can only bind with its neighbors *sequentially*. As a consequence, at each time instant, the received inputs for propagating the dynamic model and the available measurements are limited. Thus, We define the *information package* $\mathcal{IP}(j_\alpha, k)$ as below:

Definition 2 (Information Package). For agent i performing CRL task with its neighboring set being \mathcal{N}_i , the received messages from agent j_α at iteration k is denoted as $\mathcal{IP}(j_\alpha, k)$ which contains the following details:

- $\mathcal{IP}(j_\alpha, k).\mathbf{u}$: The control inputs of agent j_α .
- $\mathcal{IP}(j_\alpha, k).\mathbf{y}$: For fCRL mode, the latest stored distances d_{ij_α} of its neighbors $l \in \mathcal{N}_{j_\alpha}$ and the newly measured distance d_{ij_α} . For hCRL/nCRL mode, only the newly measured distance d_{ij_α} .

Noting the input-affine structure of the CRL model, there exists a simple CRL mechanism that only performs state estimation upon receiving the information package from all its neighbors. Compared to promptly performing estimation whenever an information package arrives, this unified scheme is more computationally efficient in that the fixed-point iteration only needs to be executed once if MLVC-EKF is applied. With a properly designed communication protocol that ensures *persistent and fair* binding for each of the neighbors, this unified scheme is a feasible solution.

Another issue is the package dropout which frequently occurs in practice. If the measurement $\mathcal{IP}(j_\alpha, k).\mathbf{y}$ (partly) gets lost, then fewer measurements will be used to compute the posteriori. In extreme cases where no measurement is received from any neighbors, then the second correction step will simply be ignored whereas only prediction is carried out. Package dropout in terms of $\mathcal{IP}(j_\alpha, k).\mathbf{y}$ is more tricky, and we need a backup algorithm that computes *predicted* inputs using previously stored input data (i.e., extrapolation).

Remark 9. Algorithm 2 is a synthesized CRL mechanism tailored for all 3 CRL models (i.e., fCRL, hCRL, and nCRL). When proceeding with the algorithm, one should simply choose one of the 3 models.

Remark 10. For the unified CRL scheme, the sampling time T_s is the average binding interval for a pair of agents. With a properly designed communication protocol, T_s is identical for all pairs (i, j_α) .

Remark 11. For the indirect distance d_{ij_α} (cf. step 7 of Algorithm 3), it will be transmitted twice by agent j_α and l sequentially, but only the latest transmitted distance will be used as an effective measurement.

Remark 12. The details of the measurement noise covariance matrix \mathbf{R}_k (cf. step 12) and the actuator noise covariance matrix \mathbf{Q}_u (cf. step 14, 22) will be given in Section V.

C. Convergence of The Fixed-point Iteration

The core part of the MLVC-EKF is the kernel-based measurement update which provides the posterior estimation using the fixed-point iteration. The convergence of the fixed-point iteration is thus important for the successful implementation of the filter. In the following results, we omit the time index k and the superscript $+$ for simplicity, and we rewrite (46) in a compact form as

$$\hat{\mathbf{x}} = \mathbf{f}(\hat{\mathbf{x}}) \quad (51)$$

Following [23], the main results regarding the convergence of (51) are summarized in Theorem 1.

Theorem 1. Given $\rho > \xi$ (cf. (53)) and $\tau \geq \max\{\tau^*, \tau^\dagger\}$, where τ^* satisfies $\Upsilon(\tau^*) = \rho$ (cf. (54a)) and τ^\dagger satisfies

Algorithm 3: CRL Mechanism

Input: fCRL(cf. (17))/hCRL(cf. (18))/nCRL(cf. (18)), \mathcal{N}_i , $\hat{\mathbf{x}}_0^+$, \mathbf{P}_0^+ , T_s , $\{\mathbf{u}_{i,k}\}_{k=0}^{k_m-1}$, $\{\mathcal{I}\mathcal{P}(j_\alpha, k)\}_{k=1}^{k_m}$, $\forall j_\alpha \in \mathcal{N}_i$

Output: $\{\hat{\mathbf{x}}_k^+\}_{k=1}^{k_m}$, $\{\mathbf{P}_k^+\}_{k=1}^{k_m}$

- 1 $k \leftarrow 1$, mode $\leftarrow 0$;
- 2 **if** fCRL/hCRL **then**
- 3 mode = 1
- 4 **end**
- 5 **while** $k \leq k_m$ **do**
- 6 Build $\bar{\mathbf{y}}_{i,k}^{\text{d}}$ by augmenting d_{ij_α} from $\mathcal{I}\mathcal{P}(j_\alpha, k) \cdot \mathbf{y}$, $\forall j_\alpha \in \mathcal{N}_i$;
- 7 Build $\bar{\mathbf{y}}_{i,k}^{\text{id}}(j_\alpha)$ by augmenting d_{lj_α} from $\mathcal{I}\mathcal{P}(j_\alpha, k) \cdot \mathbf{y}$, $\forall l \in \mathcal{N}_i \cap \mathcal{N}_{j_\alpha}$;
- 8 $\bar{\mathbf{y}}_{i,k}^{\text{id}} \leftarrow \text{vec}_{\alpha=1}^{N_i} \{\bar{\mathbf{y}}_{i,k}^{\text{id}}(j_\alpha)\}$; /* For hCRL/nCRL, the generated $\bar{\mathbf{y}}_{i,k}^{\text{id}} = \emptyset$ */
- 9 $\mathbf{u}_{j_\alpha, k-1} \leftarrow \mathcal{I}\mathcal{P}(j_\alpha, k) \cdot \mathbf{u}$, $\forall j_\alpha \in \mathcal{N}_i$;
- 10 **if** mode = 1 **then**
- 11 $\mathbf{y}_k = \text{vec}\{\bar{\mathbf{y}}_{i,k}^{\text{d}}, \bar{\mathbf{y}}_{i,k}^{\text{id}}\}$;
- 12 Compute \mathbf{R}_k based on \mathbf{y}_k ;
- 13 $\bar{\mathbf{u}}_{i,k-1} = \text{vec}\{\mathbf{u}_{i,k-1}, \text{vec}_{\alpha=1}^{N_i} \{\mathbf{u}_{j_\alpha, k-1}\}^\top\}$;
- 14 $\mathbf{Q}_k = \mathbb{I}_{N_i+1} \otimes \mathbf{Q}_u$;
- 15 Compute $\hat{\mathbf{x}}_k^+$, \mathbf{P}_k^+ by MLVC-EKF (cf. Alg 2);
/* Use augmented dynamics (cf.(11)) */
- 16 **else**
- 17 Decomposition to obtain $\hat{\mathbf{x}}_{\alpha, k-1}^+$, $\mathbf{P}_{\alpha, k-1}^+$ (the inverse operation of step 25);
- 18 **for** $\alpha = 1, 2, \dots, N_i$ **do**
- 19 $\mathbf{y}_k \leftarrow d_{ij_\alpha}$;
- 20 Compute \mathbf{R}_k based on \mathbf{y}_k ;
- 21 $\mathbf{u}_{ij_\alpha, k-1} = \text{vec}\{\mathbf{u}_{i,k-1}, \mathbf{u}_{j_\alpha, k-1}\}$;
- 22 $\mathbf{Q}_k = \text{diag}\{\mathbf{Q}_u, \mathbf{Q}_u\}$;
- 23 Compute $\hat{\mathbf{x}}_{\alpha, k}^+$, $\mathbf{P}_{\alpha, k}^+$ by MLVC-EKF (cf. Alg 2); /* Use peer-to-peer dynamics (cf.(3)) */
- 24 **end**
- 25 $\hat{\mathbf{x}}_k^+ = \text{vec}_{\alpha=1}^{N_i} \{\hat{\mathbf{x}}_{\alpha, k}^+\}$, $\mathbf{P}_k^+ = \text{diag}_{\alpha=1}^{N_i} \{\mathbf{P}_{\alpha, k}^+\}$;
- 26 **end**
- 27 $k \leftarrow k + 1$
- 28 **end**

$\Phi(\tau^\dagger; \rho) = \zeta$ ($0 < \zeta < 1$) (cf. (54b)). Then for $\hat{\mathbf{x}} \in \{\mathbf{m} \in \mathbb{R}^n \mid \|\mathbf{m}\|_1 \leq \rho\}$, the following two conditions hold:

$$\|\mathbf{f}(\hat{\mathbf{x}})\|_1 \leq \rho, \quad (52a)$$

$$\|\nabla \mathbf{f}(\hat{\mathbf{x}})\|_1 \leq \zeta. \quad (52b)$$

According to the Banach fixed-point theorem [32], with a large enough kernel bandwidth τ and a good initialization $\hat{\mathbf{x}}^{(0)}$, the fixed-point iteration converges to the unique solution of (51). Detailed expression of ξ , $\Upsilon(\tau)$, and $\Phi(\tau)$ are given as follows.

$$\xi = \frac{\sqrt{n} \sum_{i=1}^D |\mathbf{z}^*[i]| \cdot \|\mathbf{F}^*[i]\|_1}{\lambda_{\min}(\sum_{i=1}^D \sigma_i(\mathbf{F}^*))}, \quad (53)$$

$$\Upsilon(\tau) = \frac{\sqrt{n} \sum_{i=1}^D |\mathbf{z}^*[i]| \cdot \|\mathbf{F}^*[i]\|_1}{\lambda_{\min}(\sum_{i=1}^D \tilde{L}_\tau(\bar{\mathbf{e}}_i(\mathbf{F}^*)) \sigma_i(\mathbf{F}^*))}, \quad (54a)$$

$$\Phi(\tau; \rho) = \frac{(\rho \|\sigma_i(\mathbf{F}^*)\|_1 + \|\mathbf{z}^*[i] \mathbf{F}^*[i]\|)}{\tau^2 \lambda_{\min}(\sum_{i=1}^D \tilde{L}_\tau(\bar{\mathbf{e}}_i(\mathbf{F}^*)) \sigma_i(\mathbf{F}^*))}, \quad (54b)$$

where $\sigma_i(\mathbf{F}^*) := \mathbf{F}^*[i]^\top \mathbf{F}^*[i]$, $\bar{\mathbf{e}}_i(\mathbf{F}^*) := |\mathbf{z}^*[i]| + \rho \|\mathbf{F}^*[i]\|_1$,

and $\tilde{L}_\tau(x) := L_\tau^2(x)/(1+x^2)$.

Proof. The proof follows the same procedure as that of the filters using the Gaussian kernel [23]. \square

D. Computational Complexity

For robotic applications, the filtering algorithm should have good real-time performance. The MLVC-EKF is more computationally expensive due to the various operations involved in the fixed point iteration. To shed light on the applicability of the MLVC-EKF, we will analyze its computational complexity and evaluate how it scales with the number of agents as well as the number of fixed-point iterations.

The computational complexity will be analyzed in terms of the number of floating-point operations. Specifically, we consider addition (subtraction), multiplication (division), trigonometric operation, matrix inversion, etc. The specific counting rule for the mathematical operations is summarized in Table I. Involved equations that are taken into account are (33)-(38), (42)-(43), and (47)-(50). The number of floating point

TABLE I
COUNTING RULE OF MATHEMATICAL OPERATIONS

Math Operation	Counting
$M^{-1}, \mathcal{CH}(M)$ ($M \in \mathbb{R}^{n \times n}$)	$\mathcal{O}(n^3)$
$M_1 \cdot M_2$ ($M_1 \in \mathbb{R}^{m \times n}, M_2 \in \mathbb{R}^{n \times p}$)	$\mathcal{O}(mnp)$
$M_1 \pm M_2$ ($M_1, M_2 \in \mathbb{R}^{m \times n}$)	$\mathcal{O}(mn)$
$\sin(a), \cos(a), \ln(a)$ ($a \in \mathbb{R}$)	$\mathcal{O}(1)$

operations is associated with the number of neighbors N_i , the number of fixed-point iterations $T(k)$, and the number of indirect measurements p_{id} . We consider the sufficient worst case in which full-observability is ensured such that $p_{\text{id}} = 2N_i$ and choose T_m as the averaged number of fixed-point iterations. Thus, the final results will be only dependent on N_i and T_m , and the details are presented in Table II, where we separately compute the complexity for each of the terms $\mathcal{O}(N_i)$, $\mathcal{O}(N_i^2)$ and $\mathcal{O}(N_i^3)$. From the results in Table II,

TABLE II
COMPUTATIONAL COMPLEXITY

CRL Scheme	$\mathcal{O}(N_i)$	$\mathcal{O}(N_i^2)$	$\mathcal{O}(N_i^3)$
EKF Estimation			
nCRL	888	0	0
hCRL	187	289	335
fCRL	224	309	597
MLVC-EKF Estimation			
nCRL	$975 + 250T_m$	0	0
hCRL	$189 + 56T_m$	$309 + 5T_m$	$415 + 195T_m$
fCRL	$223 + 72T_m$	$325 + 21T_m$	$569 + 401T_m$

the complexity can be high when the number of neighboring agents N_i is big, especially for hCRL and fCRL schemes where $\mathcal{O}(N_i^2)$ and $\mathcal{O}(N_i^3)$ terms are present due to the matrix multiplication and inversion operations on high dimensional matrices. Another factor that greatly influences the complexity is the average number of fixed-point iterations T_m which scales badly regarding $\mathcal{O}(N_i^3)$. In real experiments, it might be necessary to restrict the maximum allowed number of iterations when using kernel-induced EKF.

V. SIMULATION RESULTS & ANALYSIS

In this section, the simulation results will be presented to demonstrate the advantages of using the fCRL scheme combined with the MLVC-EKF. We will first describe the simulation environment and setup. Next, we compare different CRL methods, in which we investigate all three CRL models with the EKF and the MLVC-EKF. Finally, in the fCRL framework, we compare the kernel-induced Kalman filter with different kernels to show the advantages of the Logarithmic-Versoria kernel.

A. Modeling Setup & Parameters

In the following description, the unit of time, position (distance) and angle are unanimously denoted as *second* [s], *meter* [m] and *radius* [rad], respectively. We simulate a group of 5 agents ($N = 5$) with each of them flying the same type of nominal trajectory. The position trajectory is given as

$$\begin{cases} x(t) = x_c + R \cos(2\pi ft + \vartheta) \\ y(t) = y_c + R \sin(2\pi ft + \vartheta) \\ z(t) = z_c + R_z \sin(2\pi f_z t) \end{cases}, \quad (55)$$

where $t \in [0s, 30s]$ is the time, and $x(t)$, $y(t)$, $z(t)$ are all global positions in a preset common reference frame. The trajectories are designed such that the observability is stimulated infinitely often. As for the heading, each agent has a different initial heading ψ_0 , and it performs *intermittent* heading turning within the simulation horizon. Specifically, an agent changes its heading of a fixed angle ψ_t during time interval $[t_i, t_i + 2]$ at a constant heading rate. The parameters describing the nominal trajectory (i.e., both the positions and the heading) for each of the agents are summarized in Table III. The nominal trajectory described above still differs from the

TABLE III
NOMINAL TRAJECTORY PARAMETERS

Agent	1	2	3	4	5
Position Parameters					
x_c [m]	0	2	-2	-2	2
y_c [m]	0	2	2	-2	-2
z_c [m]	7	8	9	6	5
R [m]	1	1.2	0.8	1.3	0.7
R_z [m]	4	4.5	6	3.5	2
f [s ⁻¹]	0.3	0.4	0.2	0.5	0.1
f_z [s ⁻¹]	0.2	0.4	0.3	0.35	0.25
ϑ [rad]	0	$\pi/4$	$4\pi/3$	$-4\pi/3$	$-\pi/4$
Heading Parameters					
ψ_0 [rad]	0	$2\pi/5$	$3\pi/5$	$4\pi/5$	$\pi/5$
ψ_t [rad]	$\pi/6$	$\pi/6$	$-\pi/6$	$-\pi/6$	$\pi/6$
t_i [s]	3; 10; 20	6; 12; 15	4; 8; 11	5; 9; 12	7; 11; 25

real trajectory due to actuator noise. In this work, we assume the actuator noise $\Delta \mathbf{u}_i = [n_\psi, n_{v,x}, n_{v,y}, n_{v,z}]^\top$ obeys a zero-mean Gaussian distribution with a *constant* covariance matrix $\mathbf{Q}_u = \text{diag}\{\sigma_\psi^2, \sigma_v^2, \sigma_v^2, \sigma_v^2\}$, where $\sigma_\psi = 0.4[\text{rad} \cdot \text{s}^{-1}]$ and $\sigma_v = 0.25[\text{m} \cdot \text{s}^{-1}]$. Based on the nominal trajectory, we can easily compute the nominal velocities as

$$\begin{cases} v_x(t) = -2\pi f R \sin(2\pi ft + \vartheta) \\ v_y(t) = 2\pi f R \cos(2\pi ft + \vartheta) \\ v_z(t) = 2\pi f_z R_z \cos(2\pi f_z t) \end{cases}, \quad (56)$$

$$\dot{\psi}(t) = \begin{cases} 0.5\psi_t, t \in [t_i, t_i + 2] \\ 0, \text{ otherwise} \end{cases}. \quad (57)$$

The real trajectory of each agent will be generated, given the initial positions (x_c, y_c, z_c) and heading (ψ_0) , using the noisy velocity (i.e., the nominal velocity plus the actuator noise).

The PDF of the basic UWB noise follows (5), and we adopt the validated parameters from [24]. As for the delay effects, the PDF of ν_d depends on both d_{jl} and $\bar{r} = \bar{\eta}\bar{v}$, and it is measurement dependent and thus time-varying. In general, a smaller d_{jl} results in a larger variance, thus making ν_d more tricky to handle. For modeling simplicity, we take $d_{jl} = 3\bar{r}$ as the worst case to make the noise distribution independent of time, and the corresponding distribution will thus only depends on $\bar{\eta}$ and \bar{v} . The parameters governing the noise distributions are outlined in Table IV.

TABLE IV
MEASUREMENT NOISE PARAMETERS

UWB Noise ν			
s_{ht}	0.2	μ [m]	0.1
λ	3.5	k	2
σ [m]	0.1		
Delay Effects Noise ν_d			
$\bar{\eta}$ [s]	0.01	\bar{v} [m · s ⁻¹]	15

In the simulation, the filter run at the discrete time $k\Delta t$ with the sampling interval $\Delta t = 0.01[\text{s}]$. When doing state estimation, the filter receives the body velocities and heading rate derived from the nominal input for prediction and then uses the noisy distance measurements for state correction. In the following, we will elaborate on the initialization settings and the noise covariance settings of the filter.

First, the filter does not know the true initial state (i.e., the true relative position and relative heading). Given the true initial *global* state tuple $\{\psi_0, x_c + R \cos(\varphi), y_c + R \sin(\varphi), z_c\}$ of all the 5 agents, the true initial state $\mathbf{x}_{ij}(0)$ for any pair (i, j) can be computed, based on which the true state $\mathbf{x}_i(0)$ for the CRL task of agent i can be built by state augmentation. Then, we manually add initial offset $\Delta \mathbf{x}_{ij}(0) = [\Delta\psi_0, \Delta x_0, \Delta y_0, \Delta z_0]^\top$ to the initial state, where the heading uncertainty is simply $\psi_0 \sim U(-\bar{\psi}_e, \bar{\psi}_e)$ and the position uncertainty is designed as

$$\begin{cases} \Delta x_0 = \bar{r}_e \cos(\varrho) \cos(\varphi) \\ \Delta y_0 = \bar{r}_e \cos(\varrho) \sin(\varphi) \\ \Delta z_0 = \bar{r}_e \sin(\varrho) \end{cases}, \quad (58)$$

where $\varrho \sim U(-\pi/2, \pi/2)$ and $\varphi \sim U(0, 2\pi)$. In other words, the design rule in (58) ensures that the initial position error has a fixed length \bar{r}_e whereas its direction is unknown. Following the state augmentation modeling in Section IV, for an agent i and its neighbors \mathcal{N}_i , we first build $\Delta \mathbf{x}_i(0) = \text{vec}_{\alpha=1}^{\mathcal{N}_i}\{\Delta \mathbf{x}_{i\alpha}(0)\}$ where $\Delta \mathbf{x}_{i\alpha}(0)$ are repeatedly generated using the same *uncertainty level* pair $(\bar{\psi}_e, \bar{r}_e)$, then the initial state of the filter is given by $\hat{\mathbf{x}}^+(0) = \mathbf{x}_i(0) + \Delta \mathbf{x}_i(0)$. Correspondingly, the initial covariance is set as $\mathbf{P}^+(0) = \mathbb{I}_{N_i} \otimes \mathbf{P}_e$, where $\mathbf{P}_e = \text{diag}\{\bar{\psi}_e^2/3, \bar{r}_e^2/4, \bar{r}_e^2/4, \bar{r}_e^2/2\}$.

Apart from the initialization, the filters should have some prior knowledge of the noise. The velocity input uncertainty is determined by the onboard hardware (i.e., IMU, gyroscope) of the MAV itself, which means the actuator noise statistics can

be obtained *offline beforehand*. Therefore, we assume that the filter has access to the true actuator covariance matrix \mathbf{Q}_u . For nCRL scheme where pairwise localization is the foundation, we set $\mathbf{Q}_k = \text{diag}\{\mathbf{Q}_u, \mathbf{Q}_u\}$ (cf. step 22 of Algorithm 3), whereas for fCRL and hCRL schemes, we have $\mathbf{Q}_k = \mathbb{I}_{N_i+1} \otimes \mathbf{Q}_u$ (cf. step 14 of Algorithm 3).

The UWB noise ideally is Gaussian (i.e., $s_{\text{ht}} = 0$ in (5)) without accounting for practical issues (e.g., NLoS, multipath, transmission delay), and this ideal UWB noise information which is only related to the UWB device itself is also available *beforehand*. The remaining tricky part is the heavy-tailed component of the UWB noise which intrinsically is environment-dependent as well as task-dependent. As a consequence, the overall UWB noise information cannot be accurately obtained in general, and how to assign a proper measurement covariance matrix \mathbf{R}_k to the Kalman-type filters is difficult and may require additional adaptive or online learning techniques which significantly increase computational complexity. In this regard, we propose the following two initialization strategies for \mathbf{R}_k .

- **Smart:** The filter has access to the true heavy-tailed noise statistics, and \mathbf{R}_k is *well* initialized to balance the heavy-tailed component.
- **Inattentive:** The filter only has information about the basic Gaussian statistics of the UWB noise, and \mathbf{R}_k is *badly* initialized.

Based on the noise parameters given in Table IV, we can numerically obtain $\sigma_\nu^2 \approx 0.08$ and $\sigma_{\nu_d}^2 \approx 0.01$. The details of the measurement covariance matrix \mathbf{R}_k for both of the above strategies are summarized in Table V.

TABLE V
MEASUREMENT COVARIANCE MATRIX \mathbf{R}_k

	Smart	Inattentive
fCRL	$\text{diag}\{0.08\mathbb{I}_{N_i}, 0.09\mathbb{I}_{p_{\text{id}}}\}$	$0.01\mathbb{I}_{(N_i+p_{\text{id}})}$
hCRL	$0.08\mathbb{I}_{N_i}$	$0.01\mathbb{I}_{N_i}$
nCRL	0.08	0.01

In our simulation, we assume that for any agent $i \in \{1, 2, 3, 4, 5\}$, its neighbor set is $\mathcal{N}_i = \{1, 2, 3, 4, 5\} \setminus \{i\}$ (i.e., all 5 agents are closed neighbors to each other). Given the above condition, for fCRL scheme, the number of indirect measurements is $p_{\text{id}} = \binom{4}{2} = 6$. We select agent 1 to be the interested agent that is going to localize its neighbors.

B. Comparison of Different CRL Methods

For each of the 3 localization schemes (i.e., fCRL, hCRL, nCRL), it can be combined with either the MLVC-EKF or the EKF. Thus, we have $2 \times 3 = 6$ different comparable CRL methods in total. For a given scheme xCRL, if it is combined with MLVC-EKF, the corresponding method will be abbreviated as xCRL(MLVC) (e.g., fCRL(MLVC) stands for the method based on the fCRL scheme using the MLVC-EKF estimation algorithm), otherwise, it will be simply denoted as xCRL if its embedded filter is the EKF.

To compare the different CRL methods, we run a Monte Carlo simulation in which we vary both the initialization setting and the noise sequences. We first set 6 uncertainty level pairs $\{(\bar{\psi}_e^q, \bar{r}_e^q)\}$ with $(\bar{\psi}_e^q, \bar{r}_e^q) = (q\pi/18, q/2)$, ($q =$

$1, 2, \dots, 6$). The unit of $\bar{\psi}_e^q$ and \bar{r}_e^q is [rad] and [m], respectively. Then, for a specific uncertainty level pair, we run 20 trials where each trial has its unique specific initialization and it is simulated with randomly generated noise sequences (for both actuator noise and measurement noise). Therefore, for a given CRL method, we run $6 \times 20 = 120$ trials in total.

For each trial, we will compute both the averaged heading error and the averaged position error, and the mathematical expression of the four considered error metrics are given as

$$\text{TR-ER}_\psi = \frac{1}{4N_s} \sum_{\alpha=1}^4 \sum_{k=1}^{k_0} |\psi_{1j_\alpha}(k) - \hat{\psi}_{1j_\alpha}(k)|, \quad (59a)$$

$$\text{SS-ER}_\psi = \frac{1}{4N_s} \sum_{\alpha=1}^4 \sum_{k=k_0+1}^{N_s} |\psi_{1j_\alpha}(k) - \hat{\psi}_{1j_\alpha}(k)|, \quad (59b)$$

$$\text{TR-ER}_p = \frac{1}{4N_s} \sum_{\alpha=1}^4 \sum_{k=1}^{k_0} \|\mathbf{p}_{1j_\alpha}(k) - \hat{\mathbf{p}}_{1j_\alpha}(k)\|, \quad (59c)$$

$$\text{SS-ER}_p = \frac{1}{4N_s} \sum_{\alpha=1}^4 \sum_{k=k_0+1}^{N_s} \|\mathbf{p}_{1j_\alpha}(k) - \hat{\mathbf{p}}_{1j_\alpha}(k)\|, \quad (59d)$$

where $N_s = 30/T_s = 3 \times 10^3$ is the total number of discretized time stamps, $k_0 = 10^3$ is the separation time stamp which separates the entire simulation horizon into the first 0s – 10s *transient* interval and the last 10s – 30s *steady-state* interval. Following the previous convention, $j_\alpha \in \mathcal{N}_1 = \{2, 3, 4, 5\}$ denotes the neighbors of agent 1, the variables without hat (i.e., $\psi_{1j_\alpha}(k)$ and $\mathbf{p}_{1j_\alpha}(k)$) are *ground truth* derived from the real trajectory and evaluated at kT_s , and the variables with hat (i.e., $\hat{\psi}_{1j_\alpha}(k)$ and $\hat{\mathbf{p}}_{1j_\alpha}(k)$) are the estimated state using a specific CRL method.

First, we carried out the Monte Carlo simulation where the embedded filter of each of the CRL methods is in *smart* mode. For a given CRL method, the corresponding performance evaluation quantities TR-ER $_\psi$, SS-ER $_\psi$, TR-ER $_p$ and SS-ER $_p$ were computed for all the 120 trials. The results are shown through the violin statistical plots in Fig. 6, where Fig. 6a and Fig. 6b summarize the TR-ER $_\psi$ (SS-ER $_\psi$) data and TR-ER $_p$ (SS-ER $_p$) data, respectively.

According to 6, it can be clearly observed that, for both transient and steady-state performance, the fCRL gives the best estimation results among the 3 CRL schemes for having lower average position and heading error in general. In addition, for all three CRL schemes, the MLVC-EKF exhibits strong power to reject outliers, especially for the fCRL scheme where more measurements are present. To demonstrate the improvements of using fCRL combined with the MLVC-EKF, we apply the bootstrapping technique to further compare the results between various comparable CRL methods. Taking SS-ER $_p$ as an example, the basic steps of the bootstrapping analysis are outlined below, and a similar procedure applies also to TR-ER $_p$, TR-ER $_\psi$, and SS-ER $_\psi$.

- 1) Fuse the SS-ER $_p$ data of two CRL methods (i.e., denoted as \mathcal{A}_p^* and \mathcal{B}_p^* , respectively) to form a hybrid data set of size $120 \times 2 = 240$.
- 2) From the hybrid data set, build data set \mathcal{A}_p and \mathcal{B}_p , both being of size 240, by resampling with replacement.
- 3) Compute the average of \mathcal{A}_p and \mathcal{B}_p , respectively. The results $\bar{\mathcal{A}}_p$ and $\bar{\mathcal{B}}_p$ are further used to calculate $\Delta p = |\bar{\mathcal{A}}_p - \bar{\mathcal{B}}_p|$.

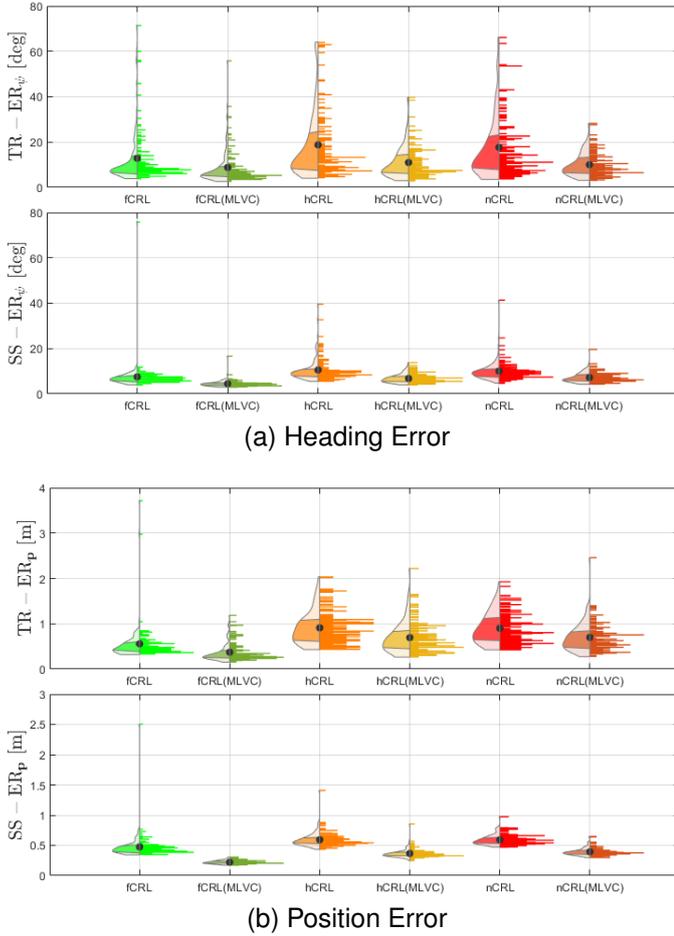


Fig. 6. Violin Plots of the Monte Carlo simulation for all 6 CRL methods with all filters in **smart** mode, (a) Averaged Heading Error (b) Averaged Position Error. For each subfigure (a) and (b), from left to right: 1) fCRL, 2) fCRL(MLVC), 3) hCRL, 4) hCRL(MLVC), 5) nCRL, 6) nCRL(MLVC). For each violin plot, the averaged error metric with respect to the trials is marked as a black dot, the left part is the approximated PDF of the data with the two boundaries of the shaded area representing the 25% and 75% quantile, and the right part is the histogram.

- 4) Repeatedly execute Step 2 and 3 for N_{bs} times to form the final data set $\{\Delta \mathbf{p}(i)\}_{i=1}^{N_{bs}}$.
- 5) Compute the nominal difference $\Delta \mathbf{p}^* = |\bar{A}_{\mathbf{p}}^* - \bar{B}_{\mathbf{p}}^*|$.
- 6) Compute the p -value of $\{\Delta \mathbf{p}(i)\}_{i=1}^{N_{bs}}$ evaluated at $p = \Delta \mathbf{p}^*$, and the p -value represents the *confidence level* of whether the original two data set $\mathcal{A}_{\mathbf{p}}^*$ and $\mathcal{B}_{\mathbf{p}}^*$ differs.

In this work, we set $N_{bs} = 10^4$, and the bootstrapping results for evaluating the steady-state performance are summarized in Fig. 7, where the plots for heading error statistics and the position error statistics are, respectively, presented in *blue* and *purple*. For each bootstrapping plot, the nominal average difference is marked with a *red* vertical dashed line, and the p -value is highlighted in *red* if it is smaller than one (i.e., not 100% confident). Based on the bootstrapping results in Fig. 7, it can be further concluded that, in terms of the steady-state heading estimation performance, there is no significant improvement when using hCRL which simply brings correlation compared to the standard nCRL regardless of the used filter, whereas the position estimation is clearly improved if the MLVC-EKF is applied since the confidence level for comparing nCRL(MLVC) and hCRL(MLVC) is 99.84%.

Next, to see how the CRL methods performs when the filters

are initialized with bad matching covariance matrices \mathbf{R}_k , we did the Monte Carlo simulation for all 6 CRL methods with *inattentive mode* filters. The violin plots are given in Fig. 8, and the bootstrapping plots are given in Fig. 9. Based on the results in Fig. 8 and Fig. 9, we can draw similar conclusions that i) the fCRL scheme outperforms the hCRL/nCRL scheme in general, ii) the MLVC-EKF outperforms the EKF regardless of the underlying CRL scheme, iii) the hCRL scheme does not provide a statistically significant advantage over the nCRL scheme.

However, when comparing the magnitude of each of the error metrics presented in Fig. 6 and Fig. 8, we can observe that the performance of the inattentive mode filter is worse than that of its smart mode counterpart for all six CRL methods. To highlight the difference between the filters of the smart mode and the inattentive mode, we summarize the average error metrics $\overline{\text{TR-ER}}_{\psi}$, $\overline{\text{SS-ER}}_{\psi}$, $\overline{\text{TR-ER}}_{\mathbf{p}}$, and $\overline{\text{SS-ER}}_{\mathbf{p}}$ with respect to all 120 trials in Table VI and Table VII for transient and steady-state performance, respectively. In both Table VI and Table VII, the percentage increase of the error metrics of CRL method using the inattentive mode filter with respect to that of the same method using smart filter is also computed and highlighted in red color, from which it is clear that the MLVC-EKF is more *robust* than the EKF in terms of handling outliers in cases where a good prior knowledge of the measurement noise statistics is unattainable.

TABLE VI
TRANSIENT PERFORMANCE (SMART(S) & INATTENTIVE(I))

	fCRL	hCRL	nCRL
Heading $\overline{\text{TR-ER}}_{\psi}$ [deg]			
EKF	(S)12.8269	(S)18.7618	(S)17.6867
	(I)21.8339 70.22%	(I)35.1224 87.20%	(I)33.1865 87.64%
MLVC-EKF	(S) 8.8004	(S)10.9913	(S)10.0108
	(I) 13.8284 57.13%	(I)14.7124 33.86%	(I)13.0007 29.87%
Position $\overline{\text{TR-ER}}_{\mathbf{p}}$ [m]			
EKF	(S)0.5614	(S)0.9191	(S)0.8925
	(I)0.9397 67.39%	(I)1.6198 76.24%	(I)1.5380 72.32%
MLVC-EKF	(S) 0.3719	(S)0.6947	(S)0.6991
	(I) 0.5080 36.60%	(I)0.8420 21.20%	(I)0.8351 19.45%

To better visualize the estimation performance and to see the advantages of using fCRL with MLVC-EKF in detail, we also present the state trajectory plots and the top-view plots of some representative trials. To avoid dense plots, we only show the results comparing the baseline CRL method (i.e., nCRL with EKF) and the proposed most advanced CRL method (i.e., fCRL with MLVC-EKF).

For the top view plots, we present the instantaneous shot at $t = 0\text{s}$, $t = 5\text{s}$, $t = 15\text{s}$, and $t = 20\text{s}$, and the trajectory of the ground truth relative position over a short period following each of the shown time instants is also presented. To also show the relative position along the z-axis, the *size of the markers (agents)* is scaled accordingly. Specifically, the size of

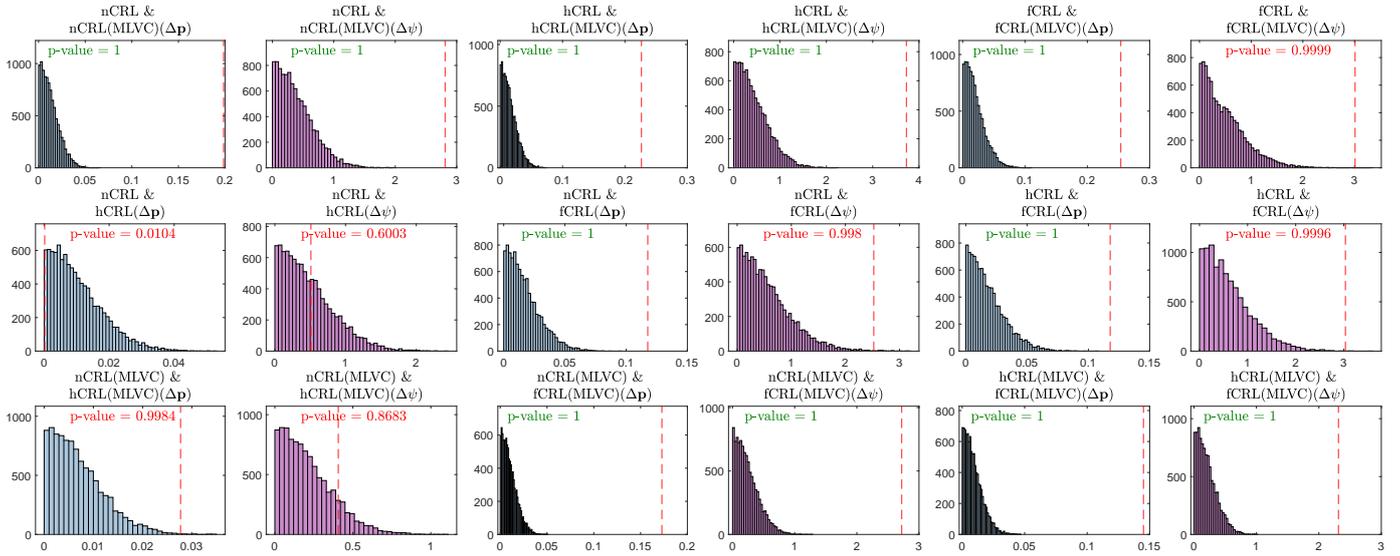


Fig. 7. Bootstrapping statistical analysis for evaluating the steady-state performance of all comparable CRL methods with **smart mode** filters

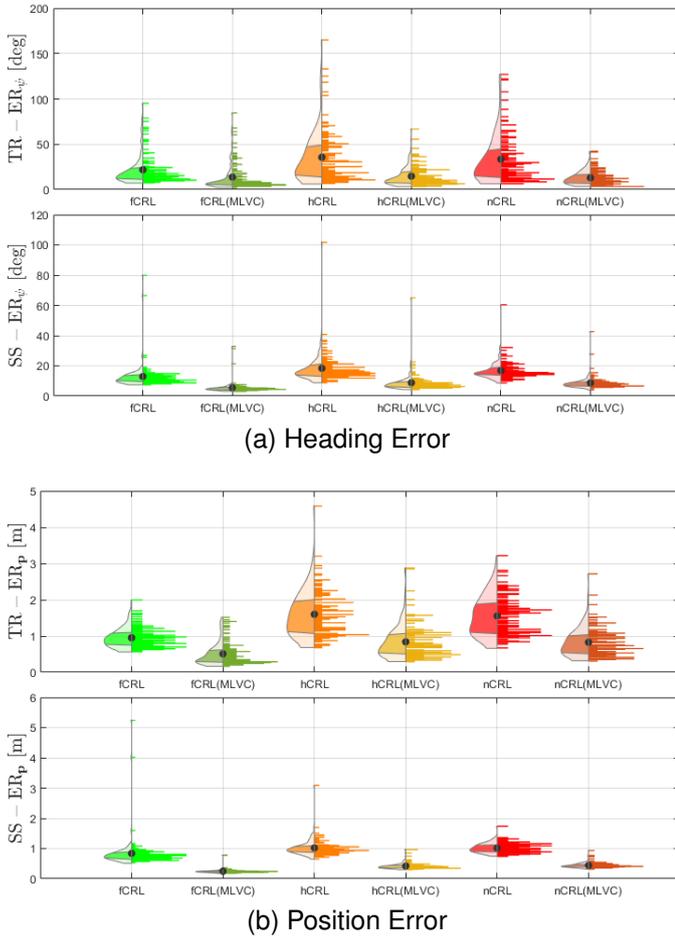


Fig. 8. Violin plots of the Monte Carlo simulation for all 6 CRL methods with all filters in **inattentive** mode, (a) Averaged Heading Error (b) Averaged Position Error. For each subfigure (a) and (b), from left to right: 1) fCRL, 2) fCRL(MLVC), 3) hCRL, 4) hCRL(MLVC), 5) nCRL, 6) nCRL(MLVC). For each violin plot, the averaged error metric with respect to the trials is marked as a black dot, the left part is the approximated PDF of the data with the two boundaries of the shaded area representing the 25% and 75% quantile, and the right part is the histogram.

TABLE VII
STEADY-STATE PERFORMANCE (SMART(S) & INATTENTIVE(I))

	fCRL	hCRL	nCRL
Heading $\overline{SS-ER}_\psi$ [deg]			
EKF	(S)7.5088 (I)12.8731 71.44%	(S)10.5429 (I)18.3855 74.39%	(S)10.1980 (I)16.3452 60.28%
MLVC-EKF	(S) 4.4992 (I) 5.4510 21.15%	(S)6.6804 (I)8.8255 32.11%	(S)7.2203 (I)8.7587 21.31%
Position $\overline{SS-ER}_p$ [m]			
EKF	(S)0.4758 (I)0.8487 78.37%	(S)0.5913 (I)1.0393 75.77%	(S)0.5843 (I)1.0189 74.38%
MLVC-EKF	(S) 0.2213 (I) 0.2488 12.43%	(S)0.3663 (I)0.4041 10.21%	(S)0.3987 (I)0.4622 15.93%

the black circle (i.e., agent 1) is the default size, and if another agent is higher (lower) than agent 1, it has a relatively bigger (smaller) marker. In addition, the relative heading angle can also be directly viewed in this top-view figure by observing the bar-shaped heading indicator attached to each of the markers (agents). Fig. 10 and 11 give the top-view plots for CRL methods with smart mode filters and inattentive mode filters, respectively. The state trajectories are given in Fig. 12 and Fig. 13, where all the four states (i.e., heading ψ_{ij} , x_{ij} , y_{ij} , and z_{ij}) are given separately, and we only show the state relating to the localization for agent 2 for simplicity.

From both the state trajectory and the top-view plots, it can be observed that, compared to the nCRL method, the fCRL(MLVC) method converges faster in the transient interval and also provides a more accurate estimation in the steady-state interval. Moreover, the trajectory of nCRL method exhibits frequent undesirable *sharp peaks* and it also deviates from the ground truth trajectory even after convergence, espe-

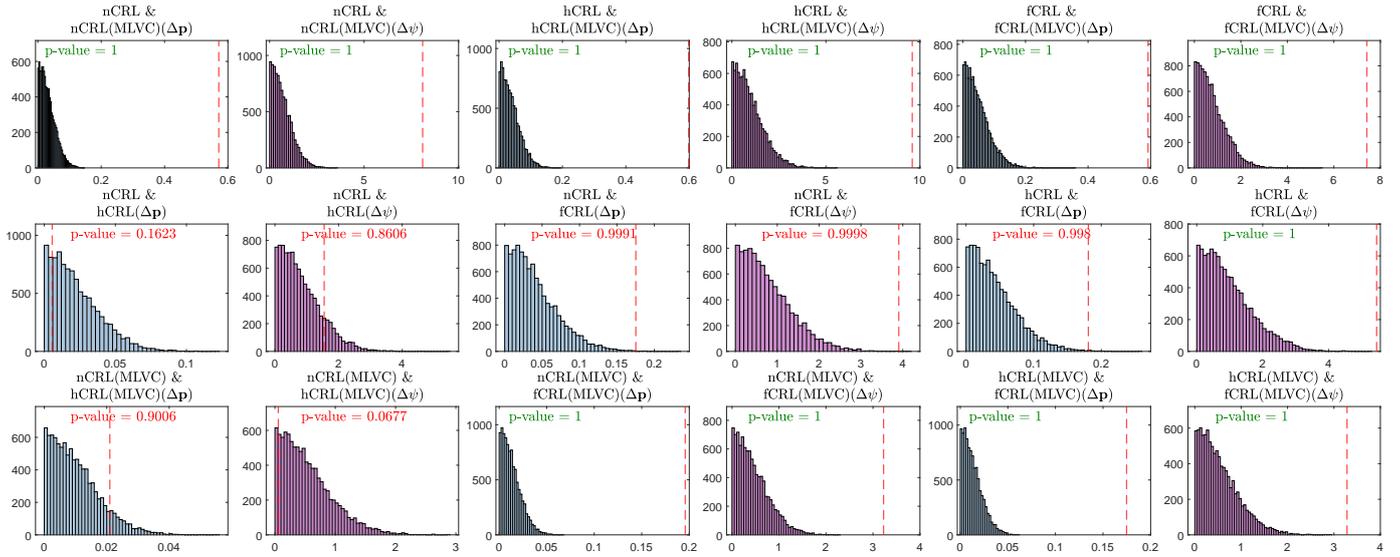


Fig. 9. Bootstrapping statistical analysis for evaluating the steady-state performance of all comparable CRL methods with **inattentive mode filters**

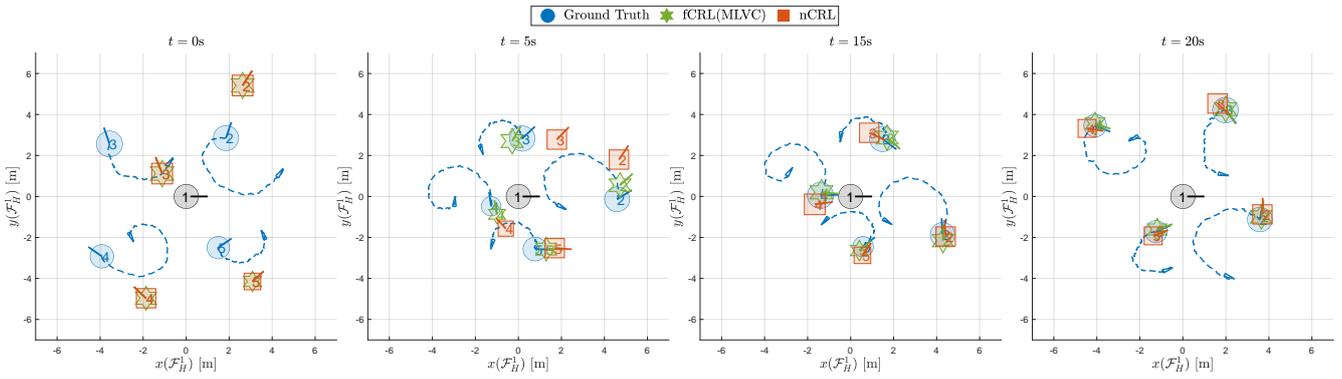


Fig. 10. Top view of the localization performance, instantaneous shot at $t = 0s, 5s, 15s, 20s$ (from left to right). The host agent 1 is the black circle always located at $(0, 0)$, the ground truth agents is the blue circles, the estimated agents using fCRL(MLVC) are the green hexagram, and the estimated agents using nCRL are the red squares. (**smart mode filters**)

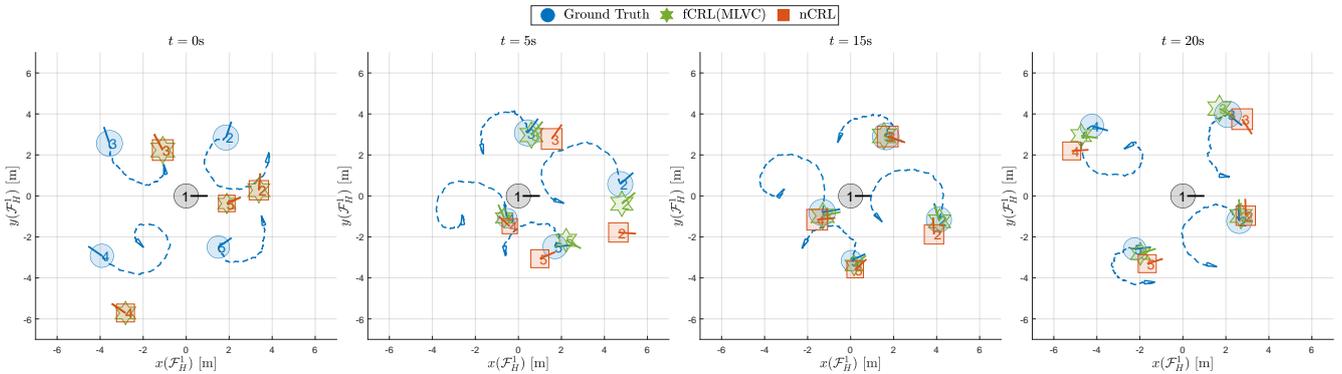


Fig. 11. Top view of the localization performance, instantaneous shot at $t = 0s, 5s, 15s, 20s$ (from left to right). The host agent 1 is the black circle always located at $(0, 0)$, the ground truth agents is the blue circles, the estimated agents using fCRL(MLVC) are the green hexagram, and the estimated agents using nCRL are the red squares. (**inattentive mode filters**)

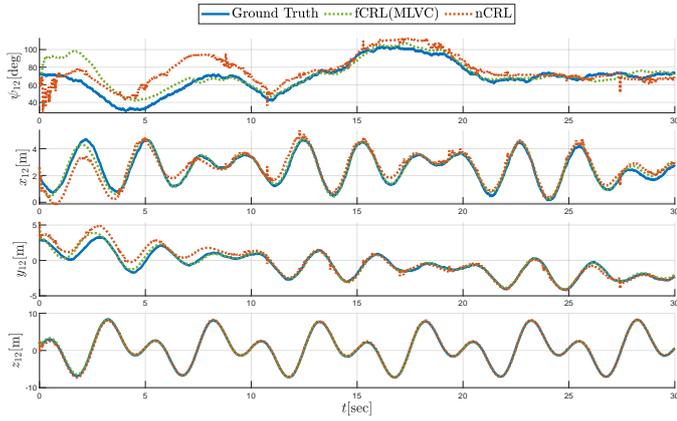


Fig. 12. State trajectory of the localization for agent 2 (smart mode filters)

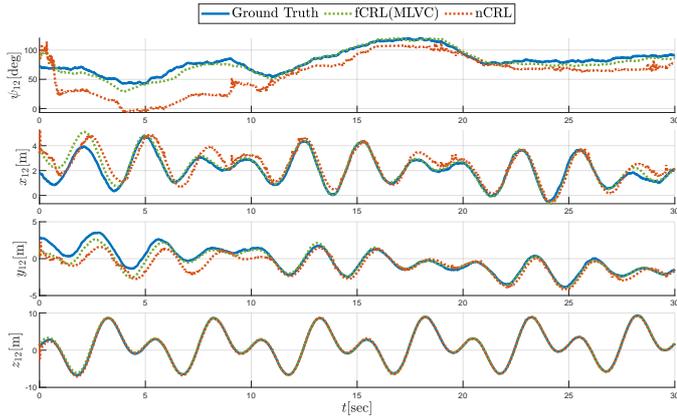


Fig. 13. State trajectory of the localization for agent 2 (inattentive mode filters)

cially in cases where the inattentive mode filters are used.

During the simulation, we also recorded the run time of different CRL methods. For a given scheme, the run time of each iteration k during trial i is denoted as $\Delta t(i, k)$, and this time cost is recorded for all time instant and for all 120 trials. The average time cost $\overline{\Delta t}$ used for evaluation is computed as

$$\overline{\Delta t} = \frac{1}{200N_s} \sum_{i=1}^{200} \sum_{k=1}^{N_s} \Delta t(i, k),$$

where $N_s = 30/T_s = 3 \times 10^3$, and the results for all 6 CRL methods are given in Table VIII. Compared with the baseline nCRL method, though using fCRL/hCRL and incorporating MLVC-EKF both increase the computational time. Specifically, the computational time increases by about 2 ~ 3% when bringing more measurements (i.e., from hCRL to fCRL), and it dramatically increases by about 260% when considering the correlation (i.e., from nCRL to hCRL). As for incorporating MLVC-EKF, the computational time increases by about 11%, 15% and 57% for fCRL, hCRL and nCRL, respectively.

TABLE VIII
AVERAGE RUN TIME (UNIT: 10^{-1} ms)

	fCRL	hCRL	nCRL
EKF	6.2822	6.0855	2.3901
MLVC-EKF	7.0136	6.9214	3.7712

C. Comparison of different kernels

In this section, we are going to compare the performance of the kernel-induced EKF with different kernels. We only consider applying the filters to the fCRL scheme, and we stick to the *smart mode* filters in terms of setting measurement covariance matrix \mathbf{R}_k . We will compare the designed LV kernel (cf. (40)), the Versoria kernel, and the Gaussian kernel. The Versoria kernel and Gaussian kernel are given as

$$V_\tau(e) = \frac{\tau}{\tau + e^2}, \quad (60a)$$

$$G_\tau(e) = \exp\left(-\frac{e^2}{\tau}\right), \quad (60b)$$

where τ is the bandwidth of each of the two kernels. For the Gaussian kernel and Versoria kernel, the corresponding maximum correntropy filter will degenerate to the standard EKF and loses its ability to handle non-Gaussian heavy-tailed noise if $\tau \rightarrow \infty$. One advantage of the LV kernel is that the corresponding filter never degenerates and the designer can simply choose a large enough bandwidth if he/she would like to avoid the divergence issue of the fixed point iteration without checking the conditions given in 1. In general, choosing a smaller bandwidth leads to a more robust filtering performance while also a larger number of iterations required in the fixed point iteration on average [23]. To compare the performance of the filters when the kernel-based measurement update is effective, we set $\tau = 5.0$ for all three kernels which also guarantees the convergence of the fixed-point iteration as stated in Theorem 1. We follow the Monte Carlo simulation setting as that of the comparison for different CRL methods, and we also use the same error metrics as in (59). The three methods are abbreviated as fCRL(MLVC), fCRL(MVC), and fCRL(MGC) for the CRL methods using the LV kernel, the Versoria kernel, and the Gaussian kernel, respectively. The violin statistical plots are given in Fig. 14, where Fig. 14a and Fig. 14b summarize the TR- ER_ψ (SS- ER_ψ) and TR- ER_p (SS- ER_p) data, respectively. Likewise, the bootstrapping plots are given in 15. According to Fig. 14 and Fig. 15, there is no much difference between the CRL methods using different kernels, especially for steady-state performance. Another important aspect that requires evaluation in kernel-based EKF is the number of fixed-point iterations. For each of the three methods, there are 3×10^3 time stamps for each trial, thus there are $3 \times 10^3 \times 120 = 3.6 \times 10^5$ data for the number of fixed-point iterations in total for the 120 trials. The histogram of the 3.6×10^5 numbers for each of the three methods is presented in Fig. 16, where the average number of the fixed-point iteration is also given. From the three histograms in Fig. 16, it is clear that the LV kernel outperforms the Versoria kernel and the Gaussian kernel for having fewer number of fixed-point iterations, which is highly desired for running the estimation algorithm online with limited computational resources.

A possible compromise is to restrict the fixed-point iteration to run *only once* at each time instant k , which would further reduce the computational complexity and is more suitable for running the estimation algorithm online. To this end, another round of Monte Carlo simulation was carried out where the methods fCRL(MLVC), fCRL(MVC), fCRL(MGC) are compared and the number of fixed-point iterations for all three methods is uniformly set to one. In addition, the results of the fCRL method (i.e., fCRL with the EKF), serving as the

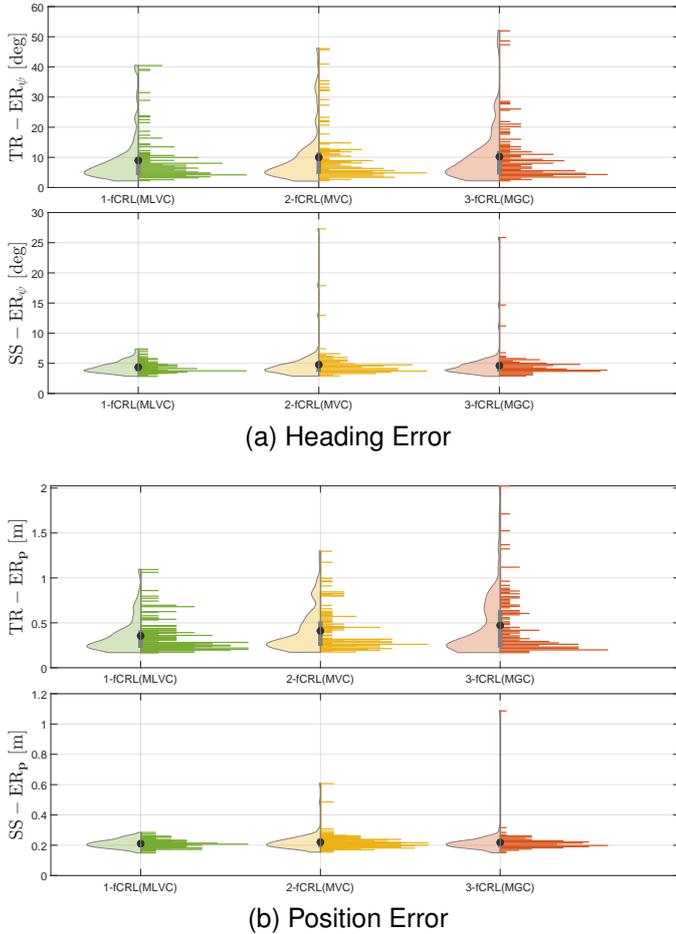


Fig. 14. Violin Plots of the Monte Carlo simulation for the 3 CRL methods with the filters in **smart** mode, (a) Averaged Heading Error (b) Averaged Position Error. For each subfigure (a) and (b), from left to right: 1) fCRL(MLVC) 2) fCRL(MVC) 3) fCRL(MGC). For each violin plot, the averaged error metric concerning the trials is marked as a black dot, the left part is the approximated PDF of the data with the two boundaries of the shaded area representing the 25% and 75% quantile, and the right part is the histogram.

baseline method, are also included for comparison purposes. The corresponding violin plots are shown in Fig. 18, and the accompanying bootstrapping results are given in Fig. 17, from which it is clear that the fCRL(MLVC) outperforms the other methods in terms of the position estimation for both the transient and steady-state intervals.

VI. CONCLUSION & FUTURE WORK

In this work, a new CRL scheme is proposed for the localization task of an agent which aims to localize all its neighbors in its body frame. The new CRL scheme can account for the correlation induced by the same velocity input and benefit from indirect distance measurements between the neighbors. Observability analysis using the augmented Lie derivative is carried out on the CRL model, which shows that bringing additional indirect measurements expands the observable subspace. To handle the heavy-tailed UWB noise, the kernel-induced Kalman filter with a novel-designed LV kernel is applied to the state estimation problem. Sufficient conditions for the convergence of the fixed-point iteration in the filtering algorithm are derived. The advantages of using the proposed CRL method in combination with the kernel-induced EKF are demonstrated through a comparative study.

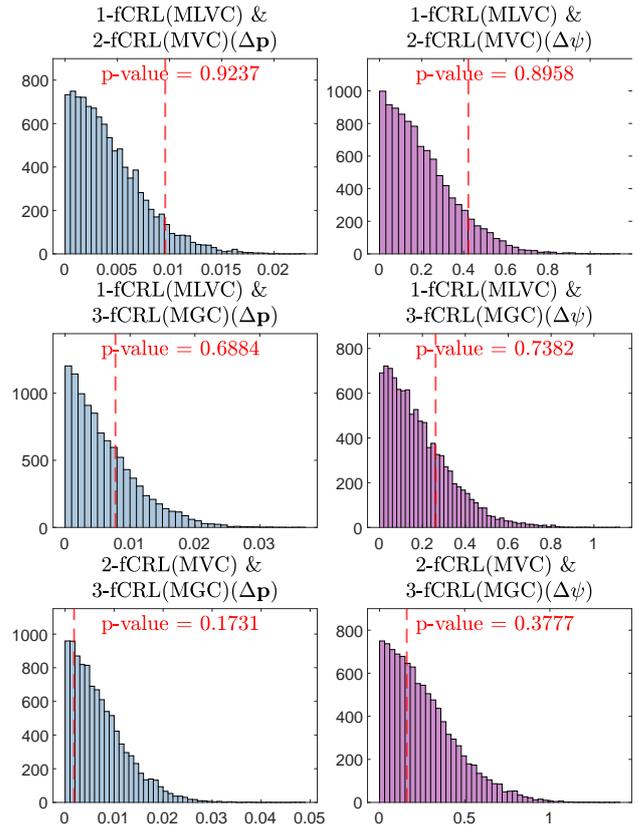


Fig. 15. Bootstrapping statistical analysis for steady-state performance of the methods fCRL(MLVC), fCRL(MVC), and fCRL(MGC).

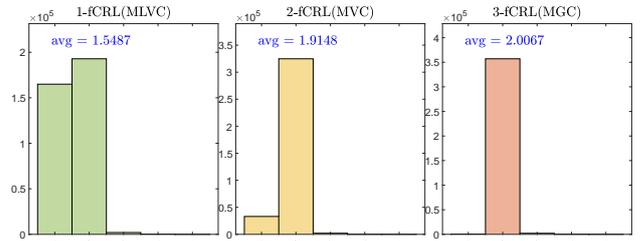


Fig. 16. Histogram of the number of fixed-point iterations. From left to right: 1) fCRL(MLVC) 2) fCRL(MVC) 3) fCRL(MGC).

Simulation results show that the proposed localization method outperforms its other variations in terms of improved estimation accuracy, robustness against measurement outliers, and insensitivity to measurement covariance matrix initialization. Moreover, the proposed filtering with the LV kernel also outperforms those using other kernels in extreme cases when the fixed-point iteration degenerates to the one-step kernel-induced measurement update.

REFERENCES

- [1] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip *et al.*, "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments," *IEEE Robotics & Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014.
- [2] A. Bahr, M. R. Walter, and J. J. Leonard, "Consistent cooperative localization," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 3415–3422.
- [3] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction

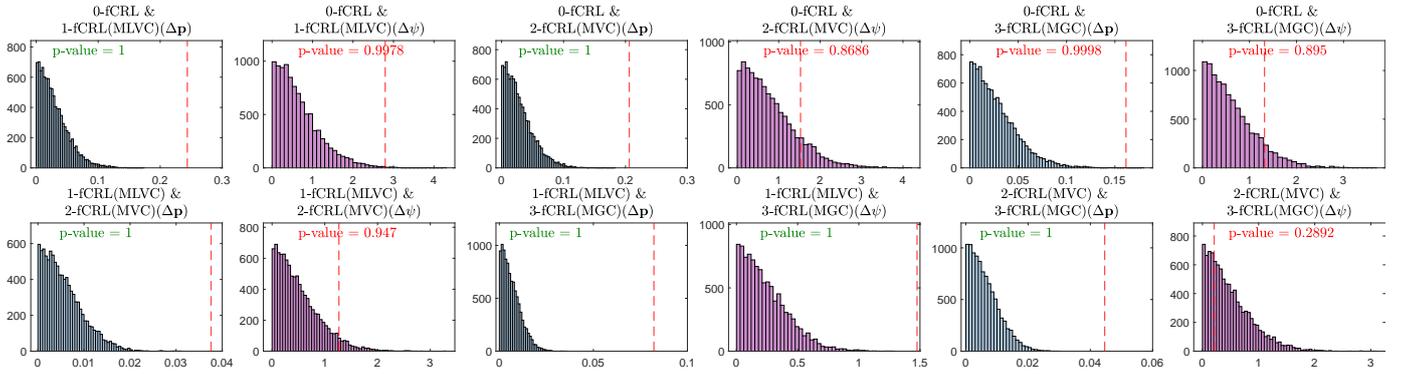


Fig. 17. Bootstrapping statistical analysis for steady-state performance of the methods fCRL, fCRL(MLVC), fCRL(MVC), and fCRL(MGC). For fCRL(MLVC), fCRL(MVC), and fCRL(MGC), the number of allowed fixed-point iterations is restricted to one.

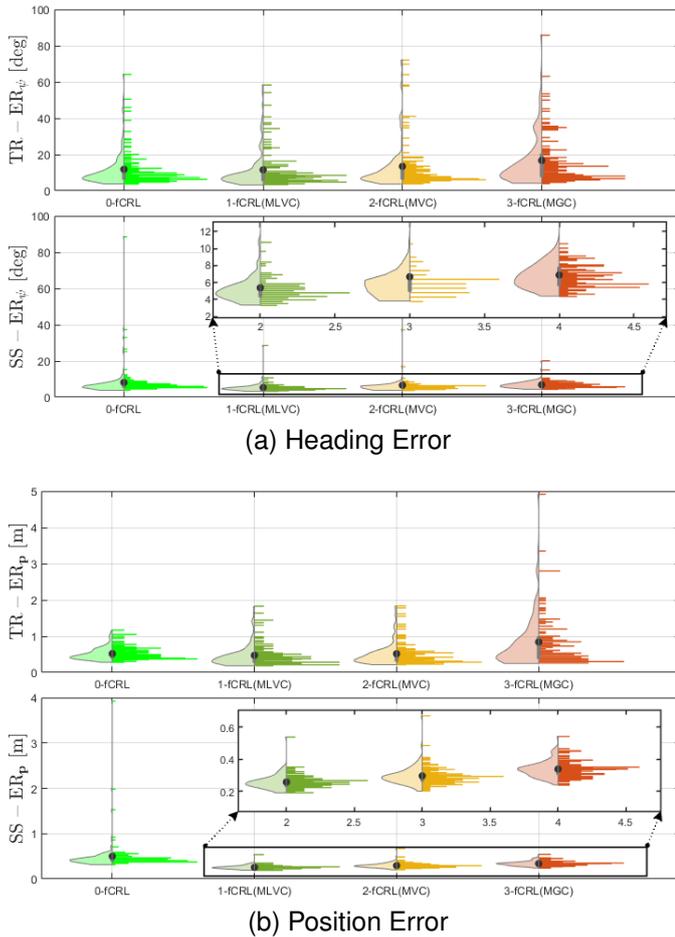


Fig. 18. Violin Plots of the Monte Carlo simulation for the kernel-induced EKF-based fCRL localization methods with only one fixed-point iteration (a) Averaged Heading Error (b) Averaged Position Error. For each subfigure (a) and (b), from left to right: 1) fCRL, 2) fCRL(MLVC) 3) fCRL(MVC) 4) fCRL(MGC). For each violin plot, the averaged error metric concerning the trials is marked as a black dot, the left part is the approximated PDF of the data with the two boundaries of the shaded area representing the 25% and 75% quantile, and the right part is the histogram.

- with flying machines,” *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 46–64, 2014.
- [4] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, “Uav-enabled intelligent transportation systems for the smart city: Applications and challenges,” *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.
- [5] B. P. Duisterhof, S. Li, J. Burgués, V. J. Reddi, and G. C. de Croon, “Sniffy bug: A fully autonomous swarm of gas-seeking nano quadcopters in cluttered environments,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9099–9106.
- [6] M. Coppola, K. N. McGuire, K. Y. Scheper, and G. C. de Croon, “On-board communication-based relative localization for collision avoidance in micro air vehicle teams,” *Autonomous robots*, vol. 42, no. 8, pp. 1787–1805, 2018.
- [7] 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, 2019.
- [8] X. Dong, Y. Zhou, Z. Ren, and Y. Zhong, “Time-varying formation tracking for second-order multi-agent systems subjected to switching topologies with application to quadrotor formation flying,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5014–5024, 2016.
- [9] S.-M. Kang, M.-C. Park, and H.-S. Ahn, “Distance-based cycle-free persistent formation: Global convergence and experimental test with a group of quadcopters,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 380–389, 2016.
- [10] S. Dudzik, “Application of the motion capture system to estimate the accuracy of a wheeled mobile robot localization,” *Energies*, vol. 13, no. 23, p. 6437, 2020.
- [11] 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, 2020.
- [12] L. Ruan, G. Li, W. Dai, S. Tian, G. Fan, J. Wang, and X. Dai, “Cooperative relative localization for uav swarm in gnss-denied environment: A coalition formation game approach,” *IEEE Internet of Things Journal*, 2021.
- [13] T. M. Nguyen, A. H. Zaini, K. Guo, and L. Xie, “An ultra-wideband-based multi-uav localization system in gps-denied environments,” in *2016 International Micro Air Vehicles Conference*, vol. 6, 2016, pp. 1–15.
- [14] J. Chudoba, M. Saska, T. Báča, and L. Přeučil, “Localization and stabilization of micro aerial vehicles based on visual features tracking,” in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2014, pp. 611–616.
- [15] R. Jurevičius, V. Marcinkevičius, and J. Šeibokas, “Robust gnss-denied localization for uav using particle filter and visual odometry,” *Machine Vision and Applications*, vol. 30, no. 7, pp. 1181–1190, 2019.
- [16] S. Güler, M. Abdelkader, and J. S. Shamma, “Peer-to-peer relative localization of aerial robots with ultrawideband sensors,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, pp. 1981–1996, 2020.
- [17] S. Li, M. Coppola, C. De Wagter, and G. C. de Croon, “An autonomous swarm of micro flying robots with range-based relative localization,” *arXiv preprint arXiv:2003.05853*, 2020.
- [18] J.-S. Lee, Y.-W. Su, and C.-C. Shen, “A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi,” in *IECON 2007-33rd Annual Conference of the IEEE Industrial Electronics Society*. Ieee, 2007, pp. 46–51.

- [19] 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*, vol. 6, no. 3, pp. 4955–4961, 2021.
- [20] Y. Liu, Y. Wang, J. Wang, and Y. Shen, "Distributed 3d relative localization of uavs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11 756–11 770, 2020.
- [21] N. Trawny, X. S. Zhou, K. X. Zhou, and S. I. Roumeliotis, "3d relative pose estimation from distance-only measurements," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 1071–1078.
- [22] B. Jiang, B. D. Anderson, and H. Hmam, "3-d relative localization of mobile systems using distance-only measurements via semidefinite optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 1903–1916, 2019.
- [23] B. Chen, X. Liu, H. Zhao, and J. C. Principe, "Maximum correntropy kalman filter," *Automatica*, vol. 76, pp. 70–77, 2017.
- [24] S. Pfeiffer, C. De Wagter, and G. C. De Croon, "A computationally efficient moving horizon estimator for ultra-wideband localization on small quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6725–6732, 2021.
- [25] A. Isidori, *Nonlinear control systems, 3rd ed.* London, Springer Verlag, 1995.
- [26] A. Martinelli, "Extension of the observability rank condition to time-varying nonlinear systems," *IEEE Transactions on Automatic Control*, 2022.
- [27] A. Martinelli and R. Siegwart, "Observability analysis for mobile robot localization," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 1471–1476.
- [28] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen, "Stochastic stability of the discrete-time extended kalman filter," *IEEE Transactions on Automatic control*, vol. 44, no. 4, pp. 714–728, 1999.
- [29] J. H. van Schuppen, *Control and System Theory of Discrete-Time Stochastic Systems*. Springer, 2021.
- [30] J. C. Principe, *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010.
- [31] F. Huang, J. Zhang, and S. Zhang, "Maximum versoria criterion-based robust adaptive filtering algorithm," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 10, pp. 1252–1256, 2017.
- [32] R. P. Agarwal, M. Meehan, and D. O'regan, *Fixed point theory and applications*. Cambridge university press, 2001, vol. 141.

Part II: Literature Study

Nomenclature

Abbreviations

Abbreviation	Definition
CDMA	Code-division Multiple Access
CEE	Centered Error Entropy
CKF	Cubature Kalman Filter
CL	Relative Localization
CRL	Cooperative Relative Localization
CRLB	Cramér–Rao Lower Bound
DPF	Dual Particle Filter
EKF	Extended Kalman Filter
FDMA	Frequency Division Multiple Access
KF	Kalman Filter
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LOS	Line-of-sight
LP	Linear Programming
LS	Least Square
MAC	Medium Access Control
MAI	Multiple Access Interference
MAP	Maximum A Posteriori
MAV	Micro Aerial Vehicle
MEE	Minimum Error Entropy
MHE	Moving Horizon Estimation
MHEKF	Multi-Hypothesis Extended Kalman Filter
ML	Maximum Likelihood
MPF	Mixture Particle Filter
MMSE	Minimum Mean Squared Error
NLOS	None-line-of-sight
NLS	Nonlinear Least Square
PF	Particle Filter
QCQP	Quadratic Constrained Quadratic Programming
RBPF	Rao-blackwellized Particle Filtler
RF	Radio-frequency (Telecommunication)

Abbreviation	Definition
RL	Relative Localization
RLT	Reformulation Linearization Technique
RSS	Received Signal Strength
SCKF	Square-root Cubature Kalman Filter
SDP	Semidefinite Programming
SNR	Signal-to-noise Ratio
SVD	Singular-value Decomposition
TDMA	Time-division Multiple Access
TDOA	Time Difference of Arrival
TOA	Time of Arrival
TWR	Two Way Ranging
UAV	Unmanned Aerial Vehicle
UWB	Ultra-wideband
UKF	Unscented Kalman Filter
WLS	Weighted Least Square

Symbols

Symbol	Definition
$\mathbb{1}_{\{P\}}$	Indicator Function with proposition P
$\ \cdot\ $	2-Norm of a vector space
$\text{Var}(\cdot)$	Variance of a given random variable
$M[i, j]$	element of the i -th row and j -th column of the matrix M
$M[i, :]$	the i -th row of matrix M
$M[:, j]$	the j -th column of matrix M
$< (\leq)$	element-wise less (less or equal to)
$> (\geq)$	element-wise greater (greater or equal to)
$\text{vec}(a_1, a_2, \dots, a_n)$	column vector with the i -th element being a_i
$\text{diag}(A_1, A_2, \dots, A_n)$	diagonalized square matrix with diagonal square blocks being A_i
\mathcal{F}^x	sigma-algebra of the random variable on the general sample space Ω
$(F^{x_1}, F^{x_2}, \dots, F^{x_n} G) \in \text{CI}$	random variables x_i are conditional independent given the sigma-algebra G
$Q \succ 0$	Matrix Q is strictly positive definite
$Q \succeq 0$	Matrix Q is positive definite
$Q \prec 0$	Matrix $-Q$ is strictly positive definite
$Q \preceq 0$	Matrix $-Q$ is positive definite

Symbol	Definition
$tr(M)$	trace (i.e., sum of the diagonal elements) of matrix M
\mathbb{R}	set of the real numbers
\mathbb{R}_+	set of the positive real numbers
$\mathbb{R}_s^{n \times n}$	set of symmetric matrices of size n in \mathbb{R}
$\mathbb{R}_{spd}^{n \times n}$	set of strictly positive definite matrices of size n in \mathbb{R}
$\mathbb{R}_{sdps}^{n \times n}$	set of symmetric strictly positive definite matrices of size n in \mathbb{R}

List of Tables

3.1	Comparison of Tightly Coupled and Loosely Coupled CL	16
3.2	Summary of Existing CRL solutions	19
4.1	Physical Quantity for System Modeling, where i, j are the agent identity indices following the convention in 3.4.2.	24
4.2	Comparison of EKF and UKF	28
4.3	Comparison of Different Filters	36

Introduction

1.1. Background

Swarm behaviors are ubiquitous in natural world animals, social insects like termite building dwelling, sardine forming bait balls to ward off predators, and a flock of birds navigating annual migration are all concrete examples. Inspiration from these creatures motivates the development of swarm robotics or multi-agent systems. Recent advances in ad hoc networking and miniaturization of electro-mechanical devices nurture successful applications in terrain mobile robots [16], underwater vehicles [6] and aerial vehicles [25]. Indeed, larger groups demonstrate their superiority to solve complex tasks (e.g., urban surveillance [22], search and rescue [71], detection and exploration [9, 35]), where efficiency, robustness, and autonomy are of major consideration, which in turn facilitates future researches on advanced swarm intelligence to better serve these requirements.

In general, swarm intelligence, from a navigational perspective, can be sequentially categorized into localization, perception, prediction, planning, and control. Though control is the final behavioral manifest after completion of its previous phases, the overall performance is determined by the integration of all the tasks, among which localization is the cornerstone whose accuracy has a great impact on all the following tasks, especially on planning and control. For swarm robotics where successful cooperation among agents is of great importance, a more paramount issue is relative localization (RL) which answers the question *where are my teammates*. RL provides essential feedback for collision avoidance [25] and formation control [50] which are fundamentals for many higher-level tasks ranging from crop monitoring [68] to parcel transportation [80]. Therefore, RL is a worthy research topic for swarm robotics.

This review focuses on a special type of micro air vehicles (MAVs), the micro quadrotors. MAVs are generally between 0.1 to $0.5m$ in length and 0.1 to $0.5kg$ in mass [83], their tiny structure directly leads to advantages such as agile behavior and capability to work in environments with limited space (e.g., small indoor workshop and underground caves), while also brings extra challenges in RL for MAVs in terms of accuracy, limited resources, and energy efficiency.

One simple but famous framework of RL is making each agent in the swarm aware of its global position in a shared frame, then relative positions can be computed based on the communicated global information among agents. Typical solutions within this framework use global positioning system (GPS) [32, 78], optical motion tracking systems [34] and anchor-dependent positioning systems with fixed beacons or ground stations [38, 120, 110]. Concerning GPS, whereas dramatic research has been carried out to improve its accuracy, the most satisfactory error range still is between $0 - 10$ meters [33], which is not suitable for MAVs where a centimeter-level of accuracy is required. Besides, the localization accuracy of GPS systems depends heavily on the measurement conditions of the client's surroundings [33], the accuracy degrades substantially in cluttered urban areas or indoor areas where MAVs, however, have great potential to perform tasks. For the above reasons, it is necessary to turn to solutions without GPS, and various researches have been published using infrared sensors [117], microphones [141], Bluetooth [25] and also ultra-wideband (UWB) [110]. Even with all these alternatives being found to replace GPS, a large group of existing solutions, see [131, 110, 166, 120, 38], are still intrinsically infrastructure dependent ones (i.e., optical motion tracking and anchor-based approaches) regardless of the

communication hardware being used. Infrastructure-dependent solutions are disadvantageous in that setting up the operation area brings extra cost, and some harsh or unknown environments (e.g., fire scenes and forests) even render setting up impossible. Additionally, communication between ground stations or fixed beacons causes increased communication workload and thus high energy consumption during operation, which also thwarts applying them to MAVs whose flight time is quite short (i.e., 10 to 20 minutes with a battery mass 30% of the total mass [83]) using Lithium batteries. Consequently, it is imperative to develop GPS-denied and infrastructure-free RL solutions for swarm MAVs.

Regarding GPS-denied and infrastructure-free localization for swarm MAVs, onboard sensors play more dominant roles. Ideally, one would place advanced sensors to tackle the problem, related approaches are localization using visual odometry with cameras [23, 75]. Nonetheless, due to the tiny structure and limited battery supply of MAVs, using sensors with high computational cost (i.e., visual information processing) also becomes problematic. As a result, the next-generation RL solutions for swarm MAVs aim to use limited onboard sensors without visual information. For this purpose, making good use of basic onboard sensors such as inertial-measurement-unit (IMU) is beneficial, the main trend is to fuse IMU with other ranging sensors such as UWB [50, 95, 94] to compensate for the noise and bias issues of IMU measurements. UWB is a low-power wireless radio communication technology that can also provide distance measurements at the centimeter level, which makes it exceptionally suitable for the RL of swarm MAVs. Furthermore, since no external components (i.e., anchors and ground stations) are present, inter-agent communication becomes indispensable, a popular trend is to combine communication and sensing, which also motivates using UWB to serve RL. Taking into account both communication capability and ranging accuracy, UWB takes precedence over infrared sensors, microphone and Bluetooth [87]. In conclusion, it is of great interest to develop RL solutions for swarm MAVs in GPS-denied and infrastructure-free environments using only IMU and UWB communication.

1.2. Related Work

One big family of localization for general multi-agent systems relies on sensor fusion which mainly hinges on filtering theory, typical solutions include the Kalman filter (KF) [94, 130, 119, 56, 51, 81, 91, 166, 106], the particle filter (PF) [48, 46, 154] and the moving horizon estimation (MHE) [152, 113, 133, 102, 153]. However, most of them are only effective for 2-D localization, and experiments are carried out for mobile robots. For those aiming for 3-D localization, most of the solutions use GPS, visual sensors, and/or fixed beacons.

Two up-to-date and representative solutions which well fit the desired problem (i.e., 3-D RL for swarm MAVs in GPS-denied and infrastructure-free environments using IMU and UWB communication) are [130] and [48]. The former uses two UWB units on some of the MAVs in the group to use attitude information [130], while the latter uses 3 UWB units on one MAV to extract both distances and bearing information for RL of a pair of MAVs [48]. Albeit they solve the RL problem in GPS-denied and infrastructure-free environments without the assistance of visual sensors, their approaches place more than one UWB unit on one drone, which is both *uneconomic* and energy *inefficient*. For the method proposed in [130], even only two MAVs require more than one UWB unit, their solution lacks robustness since the malfunction of any of these two agents will lead to failure of the RL. On the other hand, to achieve RL for swarm MAVs with the configuration in [48], many agents in the group are required to have 3 UWB units, which is costly and undesirable.

To achieve robust, energy-efficient, and economic RL, we further restrict our RL problem to the case where each MAV is only equipped with one UWB unit, and thus only *distance* measurements are attainable for a pair of drones [162]. Outstanding solutions are provided in [94], however, it trivially solves 3-D localization with laser-based altimeter assisted, which fails to handle uneven ground conditions. Though the distance is ambiguous information for the relative position, in terms of swarm robotics, multiple distances are available between each pair of agents. Indeed, *swarm though itself is challenging, it also brings extra opportunities if the potential of cooperation is exploited*. The idea of cooperative localization (CL) originates from [84], and the concept of CL is firstly officially presented in [116]. Toward CL, the main idea is a group of agents employs relative measurements concerning each other to *jointly estimate* the poses (i.e., both position and orientation) of all team members without external facilities (e.g., GPS, fixed beacons, and ground stations). CL has three components which are 1) cooperative localization algorithm 2) inter-agent sensing 3) wireless communication. This review mainly deals with CL algorithm, inter-agent sensing and wireless communication using UWB yet will

be briefly introduced, but *no* technological details will be treated regarding the latter two aspects and the final thesis following this review will also not attempt to fill in gaps in terms of improving sensing accuracy or communication protocol design.

CL algorithms can be broadly categorized as centralized CL and decentralized CL. Centralized CL requires a fusion center (FC) or a leader agent to collect information from all agents and then perform most of the computation and finally broadcast the outcome and commands to the team. Typical early achievements of centralized CL can be found in [119, 67, 118]. Obviously, for centralized CL, not only robustness is sacrificed, but communication and computational costs also scale poorly concerning the number of agents [167]. To avoid single point failure and also achieve high energy efficiency, most of the recent research is devoted to decentralized CL. Decentralized CL can be further categorized into tightly coupled and loosely coupled. Tightly coupled CL requires *all to all* communication at the time of each information exchange [81], whereas loosely coupled CL permits more communication sparsity. Many contributions have been made to distribute the computation for tightly coupled CL (see [108, 119, 143, 89]), but the *all to all* communication requirement still cannot be relaxed, which makes it unsuitable for UWB based localization where each agent can only pair to only one another agent at a time. Therefore, loosely coupled CL is preferable in our case.

The major issue involved in loosely coupled CL is how to properly account for intrinsic correlations across the whole team when measurements only involve part of the agents. Early examples which attempt to solve this problem are [9] and [17]. In [9], a bank of (extended Kalman filter) EKF is maintained for each agent, which significantly increases computational complexity and requires large memory storage. [17] assumes relative pose is attainable, which does not fit our problem since UWB can only provide relative range measurements. One popular and effective technique to ensure estimation consistency for EKF based approaches is covariance intersection, and many new results have been developed [90, 20, 158, 151, 92]. Though they serve as concrete contributions for loosely coupled CL, many of them exploit relative pose measurements and their algorithms are developed and validated for 2-D mobile robots. There is no published work about CRL in 3-D cases using only range measurements, and thus applying covariance intersection to RL is still a blank.

Focusing on CRL, the first infrastructure-free solution is given in [64], but it assumes relative pose is viable for each agent through camera or laser sensors. More recent examples on CRL are [51, 50, 120, 111]. In [51], EKF is used for the localization of swarm MAVs but it only works for 2-D localization (i.e., all MAVs fly in the same horizontal plane). [50] uses a consensus-based distributed algorithm with convergent proofs provided without relying on filtering theory. However, [50] also only works for 2-D and it further assumes uniform orientation of the swarm, which restricts its applicability for general swarms. A similar solution based on a consensus algorithm is given in [120] where a coalition game framework is used, however, this work relies on fixed reference beacons. The only exceptional work is [111] where RL is solved actively with part of MAVs performing orbital trajectories, which is also restrictive for general swarm MAVs if performing operational tasks is placed at the highest priority.

Another direction for CRL is based on graph rigidity theory. Since only distance measurements are available in our framework, we are interested in distance rigidity theory. For localization applications, classical contributions have been made in [104, 8, 4]. Though these early examples concentrate on exact localization with fixed nodes whose positions are known, they still inspire the RL algorithm in that the basic mathematical foundations remain the same. The most common way to solve distance rigidity problem for localization is using semidefinite programming (SDP) [134], which is further developed to have decentralized(distributed) computation capability [14, 149]. Recent useful solutions for 3-D relative localization are [72, 24, 103, 115, 96]. The frameworks in [103, 115] are also UWB based but visual sensors are used. [72] provides conditions on the minimum number of required distance measurements but it fails to account for intrinsic constraints imposed by UWB communication in our case. [24] sheds light on Lie group system representation which is possibly helpful for 3-D localization. Another example of realizing system transformation is [55] where barycentric coordinates are used. Analysis of localization error is presented in [96] where Cramér–Rao lower bound (CRLB) is focused. Moreover, a novel clock rigidity theory tailored for UWB based sensor network localization is presented in [156]. Nonetheless, All the above-mentioned references are purely based on rigidity theory, the only exceptional work is [155] which combines rigidity theory and KF, though it aims for exact localization instead of RL. In conclusion, an integrated solution that combines rigidity theory and filtering theory for 3-D relative localization that uses only distance measurements and also fits the UWB communication framework is still a gap that needs to be filled.

1.3. Outline of the Review

The rest of the review is organized as follows. Before delving into CL and CRL algorithms, we will first present some knowledge about UWB communication in Chapter 2 where UWB ranging principles and its relating communication protocols for localization purposes are briefly introduced. In Chapter 3, an overview of the CL problem will be given where examples and algorithms will not be restricted to only relative localization but also include exact localization. This kind of convention of combining CL and CRL is also valid for the following chapters in that CL has a strong relation with CRL and thus is beneficial to be investigated to provide useful insights. The core parts come with Chapter 4 and 5. In Chapter 4, filtering based localization techniques are elaborated. The following Chapter 5 covers graph theoretical approaches for localization. Finally, Chapter 6 concludes the review with summarizing remarks and future research directions provided.

2

Ultra-wideband Ranging for Localization

In this chapter, we present some basic knowledge on UWB ranging technique used for positioning and localization. We will first give the basic working principles of UWB ranging which sheds light on why UWB has high position estimation accuracy. Besides, essential factors relating to estimation error of UWB ranging will also be briefly visited. We then present the communication protocols used in UWB ranging focusing on some of the recent developments for localization. In the last section, we will analyze the distance measurement model using UWB ranging.

2.1. Ranging with Ultra-wideband

Ultra-wideband (UWB) signals, as the name suggests, are known for its large bandwidth compared to conventional narrow-band/wide-band signals. UWB is defined, according to Federal Communication Commission, to have an *absolute* bandwidth of at least $500MHz$ or a fractional bandwidth of larger than 20% [42], in which the upper frequency and lower frequency are determined by the $-10dB$ bandwidth rule. This large bandwidth in frequency domain leads to, in the time domain, a very short duration waveform which usually is on the order of nanosecond. Thus, commonly used UWB signals transmit *impulse pulses* with *low duty cycle*, which results the alternative name impulse radio (IR) UWB. The information is conveyed by the timings or the polarities of the pulses [42]. In addition, the large bandwidth brings several advantages for positioning/localization, such as accurate position estimation, high-speed data-transmission, penetration through obstacles as well as low power transceiver designs, which are all desired properties for micro robotics applications.

2.1.1. Two-step Positioning

Depending on whether the (relative) position is estimated from the signal traveling between the agents or not, estimation schemes mainly fall under categories *one-step direct* positioning or *two-step indirect* positioning. One-step positioning refers to estimating position directly from the signals, whereas two-step positioning first extracts positioning related parameters which are then used to estimate the (relative) position in the second step. Two-step positioning methods are more commonly used due to its low complexity in signal processing [44], our interested problem of distance-based cooperative relative localization (CRL) is also two-step in nature.

There are various positioning related parameters, such as time-of-arrival (TOA), time-difference-of-arrival (TDOA) and received signal strength (RSS). Applying a combination of different parameters for more accurate positioning has also received much attention [122]. In our framework where each MAV agent can only equip one UWB unit and peer-to-peer relative localization (RL) is a prerequisite, distance measurements become the only available *exteroceptive* information. Therefore, TOA stands out as a core parameter that perfect matches our framework. The basic principle of distance estimation using TOA intrinsically is calculating the distance as the product of the time-of-flight (τ) and speed of light in certain medium (e.g., air, water). Correspondingly, to obtain unambiguous TOA-based distance estimation requires clock synchronization between two agents or applying two-way ranging (TWR) protocols [88]. Consider a signal $s(t)$ transmitted from agent A, the received signal at another agent B

is modeled as

$$r(t) = s(t - \tau) + n(t) \quad (2.1)$$

where $n(t)$ is assumed as white Gaussian noise. The receiver generally employs a correlator/matched filter (MF) to extract the estimated time delay ($\hat{\tau}$) [44], which is formally given as

$$\hat{\tau}_{\text{TOA}} = \arg \max_{\hat{\tau}} \int r(t)s(t - \hat{\tau})dt \quad (2.2)$$

In practice, many advanced techniques (e.g., adaptive local signal copy with feedback control, first-path detection) are also used on the receiver side [44].

2.1.2. Error Sources of UWB Ranging

Accuracy limits of TOA estimation can be quantified using Cramér–Rao lower bound. For the signal as modeled in (2.1), the bound is given as [45]

$$\sqrt{\text{Var}(\hat{\tau})} \geq (2\sqrt{2}\pi\sqrt{\text{SNR}\Delta w})^{-1} \quad (2.3)$$

where SNR represents the signal-to-noise ratio and Δw is the bandwidth of the signal. (2.3) also reveals that large bandwidth has better estimation accuracy. In practical UWB positioning systems, the four major error sources are 1) multipath propagation, 2) multiple-access interference (MAI), 3) non-line-of-sight (NLOS) propagation, 4) high time resolution of UWB signals [44]. As a fact, multipath propagation is not a major issue for UWB ranging due to its large bandwidth, MAI can be mitigated through communication protocol design or nonlinear filters [123]. The most challenging tasks are mitigation of NLOS bias and accounting for clock inaccuracy or drift. Since the research focus of the thesis corresponding to this report is on cooperative localization algorithms, details relating to the NLOS or erroneous clock issues will not be further treated here.

2.2. Communication Protocols for Ultra-wideband Ranging

This section gives a short introduction to communication protocols used in localization with UWB ranging. As mentioned before in 2.1.1, clock synchronization and TWR are two alternatives to obtain reliable distance measurements. Clock synchronization is less preferable due to high time resolution of UWB signal poses challenges, on the other hand, TWR is a more flexible solution which enables arbitrary pair of nodes to band and perform distance measurements at any time [31]. However, TWR brings challenges regarding proper access control to a shared communication channel, which necessitates communication protocol design for TWR-based UWB ranging. To reduce the processing time of the receiver, medium access control (MAC) mechanism is exploited to coordinate the measurements across all agents at a given location. Typical MAC-based solutions are code-division multiple access (CDMA), frequency-division multiple access (FDMA) and time-division multiple access (TDMA).

CDMA [58], being a relatively old MAC strategy, uses spread spectrum technique to avoid interference in the same band. CDMA still requires time synchronization and thus contradicts the major advantage of TWR. As for FDMA [107] where the bandwidth of a single channel is divided into multiple sub-channels for different agents, it is obvious its performance degrades in terms of accuracy due to reduced bandwidth (c.f., (2.3)). Though improvements have been made for FDMA [162, 129], it is still not satisfactory enough [167]. TDMA [79, 99] is a promising solution to control access among multiple agents in TWR, it divides the access into different time-slots which are (dynamically) assigned to each agents. TDMA has attracted great attention recently and many variations of TDMA has been proposed, such as TDMA for distributed network [15], TDMA with adaptive slot assignment [136, 124], etc.

For swarm MAVs, a good communication protocol should satisfy the following requirements

Requirements of Communication Protocol

- If two drones are within range, then they must be able to communicate.
- The protocol should work in a dynamically changing network.
- The protocol should lead to energy efficient communication.

2.3. Distance Measurement Model using Ultra-wideband

No sensor can perfectly capture the value of the physical quantity of interest, so the UWB measurement will always be corrupted by some additional noise. In NLOS scenarios, bias also contributes to the measurement inaccuracy. Denote the relative position as $r \in \mathbb{R}^n$ where $n \in \{2, 3\}$ and the measured distance as z , then the measurement model is given as

$$z = \|r\| + b + v_z \quad (2.4)$$

where b is the bias in NLOS scenarios and v_z is the general measurement noise. In line-of-sight (LOS) scenarios, b is assumed as zero and only v_z is effective.

2.3.1. Bias identification and Mitigation

In NLOS conditions, a *positive* bias exists due to the extra time travelled by the UWB signal between the transmitter and the receiver when penetrating obstacles. To obtain accurate measurements, developing strategies to identify and mitigate the bias are necessary. Bias identification can be broadly categorized as offline methods and online methods. Offline methods rely on numerous experiments and tests to extract the quantitative characteristics of measurements, respectively, in LOS and NLOS conditions, and typical solutions are range-based methods [128], location-based methods [73] and statistics-based methods [53, 59]. Offline methods are not so preferable due to their statistical nature which limits their applicability when dealing with tasks in unknown environments. Recently, an online identification solution which compares the received signal power and first-path power demonstrates satisfactory results, though a threshold need to be chosen manually in advance.

Most recent bias mitigation techniques are machine-learning-based [159, 147, 166]. Machine learning requires large set of training data and thus requires numerous experiments beforehand. Future trends on online bias mitigation with sparse data sources are of great interest, an iterative filter is recently proposed [60] in which the bias is estimated in an active manner with control integrated.

2.3.2. Noise Modeling

Many theoretical and application-oriented projects, where sensors and actuators are analyzed, assume the noises are white and Gaussian. However, experiments show that the noise in UWB distance measurements is generally not white Gaussian noise [85, 86] and it is dependent on the relative pose. Correspondingly, Gaussian process are used to model the measurement and learning-based techniques are used to compute the parameters of the Gaussian process. It's also possible to model the noise using different distributions (i.e., Cauchy distribution [82], Gamma distribution [121]).

3

Overview of Cooperative Localization

In this chapter, an overview of CL for multi-agent systems will be given in terms of the basic categorization of CL algorithms. In this report, CL refers to cooperative absolute localization, a research area that have a rather long history and provides many useful concepts and ideas for recent research dealing directly with CRL which estimates relative pose without knowing absolute pose information in a common reference frame. Specifically, CL frameworks can be broadly categorized as centralized and decentralized ones, and decentralized frameworks can be further divided into tightly coupled and loosely coupled ones depending on whether *all to all* communication is satisfied at any time when information exchange among agents occurs. This chapter starts with several early examples where the idea of CL was first introduced and implemented. Subsequently, centralized CL will be treated with representative solutions of CL algorithms provided. Based on the insights gained from centralized algorithms, we then move on to decentralized ones, most of which are developed to achieve equal performance to their centralized counterparts. For decentralized CL, tightly coupled and loosely coupled solutions will be introduced in detail and compared in terms of communication cost, computational complexity as well as localization accuracy. After explaining the above essential property of general CL algorithms, in the final section, several representative CRL solutions will be presented, and then some discussions on the potential of using CRL to benefit the relative localization of systems with dynamic network topology will be provided.

3.1. Early Days of Cooperative Localization

The idea of cooperative localization starts from the problem of the absolute positioning of mobile robots [84]. Traditionally, given the initial position of robots in a common reference frame, their subsequent positions can be obtained using an onboard dead-reckoning system (e.g., IMU). However, positions calculated in this way are often inaccurate due to noise and bias issues intrinsically brought by the onboard sensors, and the positioning error accumulates in long-time operations. To improve the positioning accuracy without using external facilities (e.g., GPS, fixed beacons), a pioneering work published in 1994, where robots are treated as beacons themselves, provides new opportunities [84], and the punchline working principle is given below.

Principles of Cooperative Positioning [84]

1. Divide the entire robot team into two groups (i.e., Group A and Group B).
2. Fix all robots in Group A, let robots in Group B moving for a certain time interval ΔT .
3. Stop all robots in Group B, and for each robot in Group B, obtain the relative distance and relative bearing angles (i.e., horizontal angle and elevation angle) with respect to all robots in Group A.
4. Use the relative information to get the position of each robot in Group B.
5. Fix robots in Group B, and move robots in Group A and repeat step 3 and step 4 now for members in Group A.

6. Repeat step 2 to step 5 until all the robots reach their desired positions.

The working principle provided by Kurazume and Nagata [84], though lacks concrete experiments and sensing device configurations, provides inspirations for infrastructure-free localization. We summarize the highlights as follows.

General CL Highlight

- Divide the entire team into several groups
- Make part of the agents as beacons for the other robots that need to be localized

The term cooperative localization was first introduced by Rekleitis et al. in 2000 [116]. Their work inherits the same principles from the work by Kurazume and Nagata [84], whereas their task requires only two robots for terrain exploration. Besides, they provide practical solutions by using the camera to get relative poses. However, their framework assumes that two robots can always see each other, which imposes extra constraints for motion planning and prevents its capability to work in environments with many obstacles.

These early examples are indeed pioneering, yet they fail to take into account the constraints brought by sensing devices and to achieve localization with limited relative measurements. More importantly, their proposed solutions replace the deck-reckoning systems instead of adding supplements to improve the accuracy, and thus filtering theory was not applied there.

3.2. Centralized Cooperative Localization

Exploiting also on-board sensors and reducing inter-agent sensing complexity became a popular research direction for CL. In this section, we further investigate early improvements in this direction and we focus on a particular type of CL algorithms called *centralized* CL. By fusing both on-board sensors and inter-agent relative measurements, CL requires quite amount of computations. The term centralized means that the computations are carried out, in a centralized manner, either by a leader robot in the team or an external fusion center. Specifically, the leader robot or the fusion center, *at each time step*, gathers information from all agents, perform the computation online by using its own resources, then broadcasts the information (i.e., location and control commands) back to each agent. Follow the problem formulated as in [119], centralized CL using KF essentially employs *state augmentation* which collects all the states of each individual agent to build an *integrated* dynamical system at *group level*.

For a group of N agents, let $x_i(k) \in \mathbb{R}^{n_{x_i}}$ and $u_i(k) \in \mathbb{R}^{n_{u_i}}$ denote the state and control input of agent i ($i \in \mathbb{Z}_N^+$), respectively. Besides, let $w_i(k) \in \mathbb{R}^{n_{x_i}}$ and $\gamma_i(k) \in \mathbb{R}^{n_{u_i}}$ be the input noise and system noise, respectively. The original continuous-time nonlinear system dynamics of each agent i is given as

$$\dot{x}_i = f_i(x_i, u_i; w_i, \gamma_i), \quad (3.1a)$$

$$z_i = h_i(\bar{x}, v_i), \quad (3.1b)$$

where $\bar{x} = [x_1^\top, x_2^\top, \dots, x_N^\top]^\top$ is the augmented state. From (3.1), it can be observed that the dynamics of each agent is decoupled but the measurement is coupled. For instance, the measurement model for agent i measures the relative distance between agent j and itself can be expressed as

$$z_{i,j} = \|x_i - x_j\|^2 + v_i. \quad (3.2)$$

Correspondingly, expressing the group dynamics in a compact form requires state augmentation with *vectorized* dynamic equation and output equation. Typically, the augmented system is of the following form

$$\dot{\bar{x}} = \bar{f}(\bar{x}, \bar{u}; \bar{w}, \bar{\gamma}), \quad (3.3a)$$

$$\bar{z} = \bar{h}(\bar{x}, \bar{v}), \quad (3.3b)$$

where $\bar{u} = [u_1^\top, u_2^\top, \dots, u_N^\top]^\top$ and $\bar{\dagger} = [\dagger_1^\top, \dagger_2^\top, \dots, \dagger_N^\top]^\top$ ($\dagger \in \{w, v, \gamma\}$) are the augmented control input and the augmented noise input, respectively.

The centralized CL employing augmented system dynamics (3.3) requires information about inputs $u_i^*(k)$ of all agents. Moreover, even the dynamics of agents are decoupled, cross correlation will be brought in by additional relative measurements among agents. Consequently, distributing the computation is not a trivial task.

Centralized CL Summary

- Working Principles
 1. Gather information (i.e., inputs, local estimates) from all agents.
 2. Build *augmented system dynamics*
 3. Perform computation in the fusion center/leader robot according to the augmented dynamics
 4. Broadcast the estimated state to each agent
- Disadvantages
 1. Not robust regarding single point failure (i.e., failure of the leader robot or the fusion center).
 2. Increased communication cost between each agent and the fusion center/(local) leader robots.
 3. Performance scales poorly for large group teams.

At this point, we only present the preliminary steps on the construction of system dynamics for a group/subgroup of agents to illustrate the fundamental idea of *state augmentation*. Details on cross correlation and its relating distributed computation require knowledge about KF which will be treated in 4.1.1 with rigorous mathematical foundations.

3.3. Decentralized Cooperative Localization

In this section, we turn our focus to decentralized CL algorithms which are developed to resolve the disadvantages (i.e., high communication cost and lacking robustness) of centralized CL. The term *decentralized* refers to decentralized computation which means that *the computation is divided into sub-tasks and assigned to each of the agents*. Therefore, the role of each agent is equally important and each agent is only *partially responsible* for the overall localization algorithm of the entire team. Still, the algorithm for decentralized CL is cooperative in that agents need to exchange information and maintain consistency.

Elimination of the fusion center/leader agent is a major improvement of decentralized CL compared to centralized CL. Correspondingly, communication among agents becomes more important. In terms of temporal connectivity structure, decentralized CL can be further categorized into tightly coupled and loosely coupled algorithms. Tightly coupled CL requires *all-to-all* communication either by *directly* building simultaneous communication links among agents or by *indirectly* achieving *all-to-all* communication by transmitting a large amount of information containing not only the relative measurements between the pair of agents that involved in this communication stream but also other (possibly outdated) relative measurements obtained from other pairs of agents during previous streams. On the other hand, loosely coupled CL, though suffers from conservative estimation (i.e., estimation with covariance matrix having relatively big eigenvalues), has the advantage of imposing no connectivity condition on the team. We summarize the main characteristics of tightly coupled and loosely coupled CL in the following Table 3.1.

	Tightly Coupled CL	Loosely Coupled CL
Communication	<i>all-to-all</i> communication	<i>no</i> communication constraints
Correlation	full correlation	partial correlation
Estimation	optimistic	conservative
Computational Load	Low	High

Table 3.1: Comparison of Tightly Coupled and Loosely Coupled CL

3.3.1. Tightly Coupled Algorithms

The most straightforward fashion of tightly coupled CL is making each agent be an independent centralized working station, which results in *multicentralized* CL framework wherein each agent broadcasts its information (i.e., control, state estimates, variance estimates) to all other agents in the team (entire team or local team) [143], then each agent can perform centralized computation as a fusion center since it obtains information from all other interested agents. By the same token, each agent is a leader in this straightforward solution. *multicentralized* CL has a demanding computational cost for each agent and also requires a large amount of simultaneous communication among agents.

Since cross-covariance propagation is the source of difficulty for distributed computation, several methods have been developed to handle this issue without relying on a multicentralized CL framework. We summarize some of the representative works as follows.

Representative Tightly Coupled Algorithms

- **Singular-Value Decomposition (SVD) based decentralized computation** [119]:
Split covariance matrix for the corresponding pair of agents between which the relative measurements are taken. Then each agent only handle its own portion of computation, whose results are further combined in update stage of the EKF algorithm.
Communication constraints: *Direct all-to-all* communication at update stages
- **Interim Master decentralized CL (IMDCL)** [81]:
Whenever a relative measurement is obtained, the agent making this relative measurement is identified as the *interim master* which produces locally some intermediate variables and broadcasts those intermediate variables to all other interested agent either directly or indirectly (i.e., through intermediate agent(s)). At update stage of EKF, each agent makes use of the intermediate variables to compute the state estimation without extra communication.
Communication constraints: *Indirect all-to-all* communication (i.e., *spanning tree* rooted at the *interim master*)
- **Replace EKF with maximum a posteriori (MAP) Estimation** [108]:
In the least square formulation of the MAP optimization problem, the special structure of the matrix enables the distribution of the computation of the solution by assigning corresponding rows of the matrix to each agent.
Communication constraints: *Direct all-to-all* communication at update stages

3.3.2. Loosely Coupled Algorithms

Basic loosely coupled CL only updates state estimation for the pair of agents involved in the relative measurements while neglecting possible benefits from past relative measurements. In EKF based loosely coupled algorithms, an exact account of the cross-covariance terms of the entire team is not maintained during any update phase triggered by obtaining new relative measurements. Correspondingly, the localization consistency is lost according to the above basic formulation. Typical improvements for EKF based loosely coupled CL to maintain consistency have been made, which are summarized as follows.

Representative Loosely Coupled Algorithms

- **Interleaved Update** [9]:
Each agent keeps a bank of EKFs of size 2^N of maximum with N being the total number of agents. Each EKF has a well defined data structure to record all the agents that have contribution to its correlation using the identity index of the agents. Every time an agent broadcast its local information to others, the estimated states and covariance of all EKFs as well as the record are transmitted.
Advantage(s): Consistent estimation with only distance or bearing measurements
Disadvantage(s): Increased computational complexity; large memory demand; large transmitted information bits
- **Covariance Intersection** [74]:
Intuitively, covariance intersection estimate unknown cross covariance by taking the inter-

section of a pair of auto covariances. Details of this method will be elaborated in 4.1.3, we here only summarize its advantages and disadvantages. Advantage(s): Consistent estimation; Small memory demand; computation can be distributed Disadvantage(s): Only compatible with filters in which variances are accessible

3.4. Cooperative Relative Localization

This section will briefly introduce several representative algorithms for CRL, and our major concern is *direct* CRL algorithms. Here, the term *direct* means solving relative localization (positioning) without relying on absolute localization results, whereas *indirect* means computing relative pose using result of absolute localization.

Since relative poses can be trivially computed given a set of the absolute poses in global coordinates, absolute localization problems have been investigated intensively before *direct* relative localization came to focus. Indirect methods can hardly be infrastructure-free since at least one agent needs to interact with the global reference frame through either GPS or fixed beacons, otherwise the absolute position is unattainable. Therefore, direct methods are more economic and more suitable for infrastructure-free localization.

The main concepts (i.e., centralization/decentralization, loosely coupled/tightly coupled) in localization problems were initially proposed for absolute localization. However, these essential concepts, being related to universal aspects (i.e., computational framework, communication condition), are not restricted to absolute localization, and thus can also be applied to relative localization.

Apart from centralization/decentralization and loosely coupled/tightly coupled categorizations, we are also interested in whether a given solution works for 2-D/3-D as well as whether it is infrastructure-free/infrastructure-dependent. Moreover, we do not consider solutions that benefit from GPS or visual sensors. Given the above conditions, all interested categories are given as follows:

1. **computation & communication framework:** 1) centralized, 2) tightly coupled decentralized, 3) loosely coupled decentralized
2. **localization dimension:** 1) 2-D only, 2) 2-D/3-D, 3) 2-D & altimeter
3. **infrastructure dependency:** 1) infrastructure-free, 2) infrastructure dependent
4. **measurement type:** 1) distance only, 2) bearing only, 3) hybrid (distance and bearing) 4) relative pose
5. **estimated variables:** 1) relative position, 2) relative pose (i.e., both position and orientation)

3.4.1. Representative Solutions

CRL starts from an early work where relative poses are attainable for each robot [64] and the cooperation simply means combining pose measurements from different agents to reduce the measurement uncertainty. Since obtaining relative pose itself brings challenges to on-board sensors, solutions using only distance measurements and rigidity theory [104] gained favor. However, solely relying on rigidity theory to solve RL treats agents simply as nodes in a graph, which fails to provide orientation estimation. To estimate relative pose, ego-motion information obtained from inertial sensors (e.g., IMU) becomes popular, and a pioneering contribution, which exploits both IMU and relative distance, formulates the relative pose estimation as a least square optimization problem [142]. The first consensus-based RL was proposed for source localization where sampling effects of measurements were also considered [18], yet their framework is infrastructure-dependent. Recently, infrastructure-free consensus-based RL has been developed [50], but it only works for 2-D scenarios and it assumes a common reference frame. A similar game-theoretic solution [120] has also been proposed, it still requires ground stations and thus is also infrastructure-dependent.

Consensus-based solutions lead a popular trend in distributed CRL. Back in the early days, two algorithms focusing on distributed computation were developed for CRL [26]. In this work, one solution relies on degenerate motion assumption, while the other relies on motion control and requires part of the agents to be static. Therefore, both distributed solutions have limited applicability. Another direction for distributed CRL is achieving peer-to-peer RL, solutions using single link range measurements [30, 137, 51] has also been developed. Nevertheless, those early peer-to-peer solutions mean the cooperation of a pair of agents instead of benefiting from a large group.

Most of the existing solutions, regardless of applications (i.e., aerial vehicles [18, 50], mobile robots [64, 104], mobile users [95]), only solve RL in 2-D scenarios. For 3-D infrastructure-free RL which tasks using MAVs greatly desire, altimeters are pervasively used [25, 51]. The only exceptions solving 3-D RL without altimeter also fail to account for relative orientation [96, 72] since they purely rely on graph optimization. There are also solutions aiming for swarm aerial vehicles that place more than one UWB unit on each agent [130, 48], which makes them unsuitable for MAVs. We summarize and categorize representative relevant works on CRL in Table 3.2.

	Computation & Communication	Dimension	Infrastructure Dependency	Measurement	Estimated Variables
Sensor Network (Rigidity Theory) [104]	Decentralized Tightly Coupled	2-D	Free	Distance	Relative Position
Ground Vehicle (Weighted LS) [142]	(Not Applicable)	2-D	Free	Distance	Relative Pose
UAV (Consensus)[18]	Decentralized Loosely Coupled	2-D	Dependent	Distance	Relative Position
Sensor Network (Particle Filter) [64]	Decentralized Tightly Coupled	2-D	Free	Relative Pose	Relative Pose
UAV (EKF) [112]	(Not Applicable)	3-D	Free	Distance & Bearing	Relative Position
Sensor Network (NLS) [26]	Decentralized Tightly Coupled	2-D	Free	Distance	Relative Pose
Mobile Vehicle (Graph Theory)[30]	(Not Applicable)	3-D	Free	Distance	Relative Position
UAV (Graph Theory) [137]	(Not Applicable)	2-D	Free	Distance	Relative Position
UAV & UWB (EKF) [51]	Decentralized Loosely Coupled	2-D Altimeter	Free	Distance	Relative Position
UAV & Bluetooth (EKF) [25]	Decentralized Loosely Coupled	2-D Altimeter	Free	Distance	Relative Pose
Mobile User (Particle Filter)[95]	Centralized	2-D	Free	Distance	Relative Position
UAV (Consensus) [57]	Decentralized Loosely Coupled	2-D	Free	Distance	Relative Position
UAV & UWB (Consensus) [50]	Decentralized Loosely Coupled	2-D	Free	Distance	Relative Position
Mobile Vehicle & UWB (EKF) [19]	Decentralized Loosely Coupled	2-D	Free	Distance	Relative Pose
UAV (Graph Theory) [96]	Decentralized Tightly Coupled	3-D	Free	Distance	Relative Position
UAV (Game Theory) [120]	Decentralized Tightly Coupled	3-D	Dependent	Distance	Relative Position
Sensor Network (Rigidity Theory) [72]	Decentralized Tightly Coupled	3-D	Dependent	Distance	Relative Position
UAV (MHE + NLS) [27]	Decentralized Loosely Coupled	3-D	free	Distance	Relative Pose

Table 3.2: Summary of Existing CRL solutions

Due to the limited onboard computational capability of MAVs, solutions using recursive filtering are more favorable. Moreover, filtering-based solutions do not require a tightly coupled communication setting in that they solve peer-to-peer relative localization as a prerequisite [94, 19].

The simplest recursive filtering technique for nonlinear systems is the EKF which provides suboptimal estimation for the linear dynamics of the *incremental states*. Though the EKF has shown good performance for many engineering applications, it does not have a solid theoretical foundation regarding stability and convergence. One *necessary* condition for the successful implementation of the EKF is having nonlinear observability [25, 94, 19, 66, 65], a property that was originally proposed to design observers for deterministic linear systems. Another factor that influences the performance of the EKF is initialization. In general, if the initial estimation of the filter deviates a lot from the true state, then

the EKF can easily diverge. The main contribution of Chakraborty et al. [19] is using a bank of EKFs with different initialization guesses to tackle the initialization sensitivity of the EKF, and they call this technique the multi-hypothesis EKF (MHEKF).

In contrast to recursive filters where only one single distance measurement is used for estimation, the batch filters that use multiple measurements for estimation in each step are also powerful for CRL, and they provide more stability guarantees in practice [142, 26, 27]. One notable batch filter is the MHE, and it also has other variant names such as the sliding window estimator [27]. The MHE and many of its variant estimators intrinsically formulate the estimation problem as an optimization problem, such as nonlinear least-square (NLS) [26, 27] that can be solved by *Gauss-Newton* algorithm or more advanced *Levenberg-Marquart* algorithm for improved convergence.

There are also solutions where CRL is solved directly at the *group level* without solving peer-to-peer RL, two classes or representative solutions are those using distributed optimization (e.g., consensus [50], and game theory [120]) and graph rigidity theory [104, 137, 96]. Existing solutions fall in the *group level* category all assume that a global reference frame is available for all agents [50] or simply treating the robots as sensor nodes [104, 137, 96]. Since we are interested in pose estimation (i.e., relative position estimation in body frame), they are not so applicable compared to filtering-based solutions. Still, for distributed optimization approaches, adding the heading angle or the general attitude as the extra learnable state may also lead to a valid solution to our interested problem. And in terms of rigidity theory, it may serve as a complementary tool if used in combination with nonlinear filters.

3.4.2. Cooperation for Relative Localization of Dynamic Networks

In swarm robotics, all the agents in the swarm together form a network when the communication links are built. Notably, the communication link between any pair of agents may not be valid all the time (i.e., a link may be valid for an instant and then be destroyed after a while), which renders the whole network dynamic in the sense of validity of the communication links. Another factor contributing to the time-varying nature of the network is the number of participating agents. Specifically, incoming agents joining the group and existing agents leaving the group also make the network dynamic.

We present some knowledge of graph theory for modeling swarm robotics. Each agent in the swarm is regarded as a *node* and a valid sensing/communication link is an edge connecting its corresponding nodes. Then the *sensing topology* of the swarm can be described by a *directed graph* $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, 2, \dots, N\}$ is the *node set* with N being the total number of agents and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the *edge set*. An ordered pair $(i, j) \in \mathcal{E}$ means that agent i can measure its distance to agent j . Correspondingly, the *undirected graph* $\bar{\mathcal{G}} = \{\mathcal{V}, \bar{\mathcal{E}}\}$, being the counterpart of its directed graph \mathcal{G} , describes the *measurement(distance) topology* of the swarm network in which $(i, j) \in \bar{\mathcal{E}}$ if and only if $(j, i) \in \bar{\mathcal{E}}$. Subsequently, we refer the dynamic behavior of the communication links as *dynamic edge* and that of the participating agents as *dynamic node*. For convenience, we assign, for each graph $\mathcal{G}(\bar{\mathcal{G}})$, an square matrix $M_e \in \{0, 1\}^{N \times N}$ to its edge set $\mathcal{E}(\bar{\mathcal{E}})$. The relation between $\mathcal{E}(\bar{\mathcal{E}})$ and its corresponding matrix M_e is established according to the following rule

$$M_e[i, j] = \mathbb{1}_{(i, j) \in \mathcal{E}}. \quad (3.4)$$

Further denote the mapping from edge set to its corresponding matrix as f_e (i.e., $M_e = f_e(\mathcal{E})$), then apparently the mapping f_e is *bijective*. Therefore, it is safe to completely use the *edge matrix* M_e to describe the coupling of the swarm whenever needed. In our framework where UWB ranging is used, we denote the maximum communication range between any of the two agents as \bar{d} . The agents are moving in n -dimensional Euclidean space \mathbb{R}^n ($n \in \{2, 3\}$) and their exact positions are denoted as $p_i \in \mathbb{R}^n$, where $i \in \mathcal{V}$. Additionally, we define *range matrix* M_r for a given node set \mathcal{V} according to the following rule

$$M_r[i, j] = \mathbb{1}_{\|p_i - p_j\| \leq \bar{d}}. \quad (3.5)$$

According to (3.4) and (3.5), we have $M_e[i, j] \leq M_r[i, j]$. As a result, we define the *range graph* as a 3-tuple $\mathcal{G} = \{\mathcal{V}, M_e, M_r\}$ where M_e characterizes the validity of sensing/communication and M_r characterizes ability of sensing/communication. In the sequel, we will simply use *graph* to indicate *range graph*. Any sub-group of agents is denoted as \mathcal{S} ($\mathcal{S} \subseteq \mathcal{V}$). Given \mathcal{S} , it also has a corresponding directed graph and undirected graph denoted respectively as $\mathcal{G} = \{\mathcal{S}, M_e, M_r\}$ and $\bar{\mathcal{G}} = \{\mathcal{S}, M_e, M_r\}$. Following the discrete-time convention in 3.2, a graph is a function of time $t \in T$ (i.e., $\mathcal{G}(t) = \{\mathcal{S}(t), M_e(t), M_r(t)\}$). A specific time interval is denoted as $T[t_1, t_2] := \{t_1, \dots, t_2\}$. To distinguish between the above mentioned

two dynamic behaviors (i.e., dynamic node and dynamic edge), we need several formal definitions for the network.

Definition 1 (Compact Graph) *Given a group of agents $S \subseteq \mathcal{V}$ and its relating graph \mathcal{G} , it is compact if $M_r \mathbf{1} \geq \mathbf{1}$. A compact graph is denoted as \mathcal{G}_c*

Definition 2 (Loosely/Tightly Compact Graph) *Given a compact graph \mathcal{G}_c , it is loosely compact if $M_r < \mathbf{1}$, and it is tightly compact if $M_r = \mathbf{1}$.*

Definition 3 (Fixed-node Graph) *Given a group of agents $S \subseteq \mathcal{V}$ and its relating graph \mathcal{G} , it is fixed-node with respect to time interval $T[t_1, t_2]$ if $S(t) = S^*, \forall t \in T[t_1, t_2]$*

Subsequently, we present the definition of *node joining(leaving)* and *edge changing* as follows.

Definition 4 (Node Joining of Compact Graph) *Given a compact graph $\mathcal{G}_c(t_1)$ at $t_1 \in T$, if $\exists t_2 \in T, t_1 < t_2$ such that $M_r(t_2) \mathbf{1} \geq \mathbf{1}$ (i.e., graph is still compact at t_2) and $S(t_1) \subset S(t_2)$, then we define this variation as node joining.*

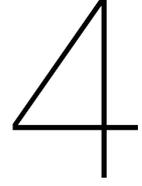
Definition 5 (Node leaving of Compact Graph) *Given a compact graph $\mathcal{G}_c(t_1)$ at $t_1 \in T$, if $\exists t_2 \in T, t_1 < t_2$ such that $M_r(t_2) \mathbf{1}$ has at least one 0 entry. then we define this variation as node leaving. Remove the nodes corresponding to the 0 entries will generate a compact graph again.*

Definition 6 (Edge Changing of Fixed-node Compact Graph) *Given a fixed-node compact graph $\mathcal{G}_c = \{S^*, M_e(t), \mathbf{1}\}$ on $T[t_1, t_2]$, if $\exists t_a, t_b \in T[t_1, t_2], t_a < t_b$ such that $M_e(t_a) \neq M_e(t_b)$, then we define this variation as edge changing*

Given the above definitions, we now present some insights on how cooperation benefits the RL problem for dynamic networks

Insights for CRL for dynamic networks

- For agents forming a loosely compact graph, there exists at least a pair of agent which cannot build a communication link and measure inter-agent distance directly and thus their relative pose cannot be estimated directly. However, they are indirectly connect by other agents, then if they respectively estimate their relative poses to the *media* agent(s), their relative pose can also be obtained.
- For agents forming a general compact graph, the relative pose of any pair of agents may be estimated both directly and indirectly, then fusing the direct and indirect estimation will increase the estimation accuracy.



Filtering Theory Based Cooperative Relative Localization

This chapter focuses on CRL solutions using different filtering techniques. In current literature, dominant filters are the Kalman filter (KF), particle filter, and moving horizon estimation (MHE). In this chapter, we will provide mathematical foundations for each of these filters and analyze some of the representative examples in the existing literature. At last, a comparative analysis will be presented in which we summarize the advantages and limitations of the different filters in terms of various aspects, e.g., convergence, initialization impact, computational complexity, etc.

4.1. Kalman Filter Based Cooperative Relative Localization

This section deals with KF as well as some of its classical variations. Some Novel extensions in existing literature will also be covered. We will first explain the basic working principles of the extended Kalman filter (EKF) which is a widely used nonlinear filtering algorithm. Then, we briefly summarize the observability issue using KF, focusing on stochastic nonlinear systems. Covariance intersection, being a powerful tool for computing unknown correlations, will be introduced from a CRL perspective. We end this section by analyzing the initialization impact on estimation performance using EKF, and several advanced extensions of EKF will also be presented.

4.1.1. Kalman Filter Introduction

The KF, also known as *linear quadratic estimation*, is an information extraction algorithm that uses a system model and sequence of measurements to estimate unknown variables in the presence of noise. It is named after Rudolf E. Kálmán, who is one of the founders of KF theory [77]. The algorithm has a recursive nature, which forces the discretization of general continuous system equations before applying the algorithm. Though a continuous version of KF (i.e., Kalman-Bucy filter [13]) also exists, we only focus on discrete-time KF in this report.

Following the general system described in (3.1), we consider noise-affine nonlinear system of the form

$$\dot{x} = f(x, u; \gamma) + g_w(x)w, \quad (4.1a)$$

$$z = h(x) + g_v(x)v. \quad (4.1b)$$

The real-time sampling interval is chosen to be δT , with which the set of discretized time instant is $T_{\text{dis}} = \{0, \delta T, 2\delta T, \dots\}$. For simplicity, we use the time index set $T = \{0, 1, 2, \dots\}$ to replace T_{dis} . Applying Euler discretization, the discrete-time nonlinear system dynamics can be obtained from (3.1) as

$$x(k+1) = x(k) + \delta T f(x(k), u(k); \gamma(k)) + \delta T g_w(x(k))w(k) := F(x(k), u(k); \gamma(k)) + G_w(x(k))w(k), \quad (4.2a)$$

$$z(k) = h(x(k)) + g_v(x(k))v(k), \quad (4.2b)$$

where $\ddagger(k), \ddagger \in \{x, u, w, \gamma, v\}$ represents the variable at time instant $k \in T$. To apply the EKF, we linearize both the system dynamics (4.2a) and the output equation (4.2b) at $x_e := (\hat{x}^+(k), u^*(k); \gamma = 0, w = 0, v = 0)$, where $\hat{x}_i^+(k)$ is the posteriori state estimation and $u^*(k)$ is the input computed by the controller. The final discrete-time linear system is given as

$$\begin{aligned}\tilde{x}(k+1) &= \left. \frac{\partial F}{\partial x} \right|_{x_e} \tilde{x}(k) + \left. \frac{\partial F}{\partial u} \right|_{x_e} (\tilde{u}(k) + \gamma(k)) + G_w(\hat{x}^+(k))w(k) \\ &= (I_{n_x} + \left. \frac{\partial f}{\partial x} \right|_{x_e}) \tilde{x}(k) + \left. \frac{\partial F}{\partial u} \right|_{x_e} (\tilde{u}(k) + \gamma(k)) + G_w(\hat{x}^+(k))w(k) \\ &:= A(k)\tilde{x}(k) + B(k)(\tilde{u}(k) + \gamma(k)) + G_w(\hat{x}^+(k))w(k)\end{aligned}\quad (4.3)$$

And the linearized output equation is given as

$$\begin{aligned}\tilde{z}(k) &= \left. \frac{\partial h}{\partial x} \right|_{x_e} \tilde{x}(k) + g_v(\hat{x}^+(k))v(k) \\ &= H(k)\tilde{x}(k) + g_v(\hat{x}^+(k))v(k).\end{aligned}\quad (4.4)$$

Further, we assume that the input increments $\tilde{u}(k)$ is negligible compared to the actuator noise $\gamma(k)$. Besides, denote $G_w(\hat{x}^+(k))$ and $g_v(\hat{x}^+(k))$ as $G_w(k)$ and $D(k)$, respectively. Then (4.3) can be simplified to

$$\tilde{x}(k+1) = A(k)\tilde{x}(k) + B(k)\gamma(k) + G_w(k)w(k), \quad (4.6a)$$

$$\tilde{z}(k) = H(k)\tilde{x}(k) + D(k)v(k). \quad (4.6b)$$

we assume that the input noise $\gamma(k)$, process noise $w(k)$ and measurement noise $v(k)$ are independent and Gaussian distributed (i.e., $(F^{\gamma(0:T-1)}, F^{w(0:T-1)}, F^{v(0:T-1)}) \in \text{CI}$). Denote $\eta(k) = [\gamma(k)^\top, w(k)^\top]^\top$ and $G(k) = [B(k), I]$, the resulting system equation is given as

$$\tilde{x}(k+1) = A(k)\tilde{x}(k) + G(k)\eta(k), \quad (4.7a)$$

$$\tilde{z}(k) = H(k)\tilde{x}(k) + D(k)v(k). \quad (4.7b)$$

We further denote the parameters of the Gaussian distribution as $\eta(k) \in G(0, Q(k))$ and $v(k) \in G(0, R(k))$, The EKF algorithm is given as follows.

EKF Algorithm

Given the initial mean $\hat{x}^+(0)$ and the initial covariance $P_x^+(0)$ of the state x . For $k = 0, 1, 2, \dots$,

1. State Propagation

$$\hat{x}^-(k+1) = F(\hat{x}^+(k), u^*(k)) \quad (4.8)$$

2. State Covariance Propagation

$$P_x^-(k+1) = A(k)P_x^+(k)A(k)^\top + G(k)Q(k)G(k)^\top \quad (4.9)$$

3. Kalman Gain Computation

$$K(k) = P_x^-(k+1)H(k)^\top (H(k)P_x^-(k+1)H(k)^\top + D(k)R(k)D(k)^\top)^{-1} \quad (4.10)$$

4. Posterior State Update

$$\hat{x}^+(k+1) = \hat{x}^-(k+1) + K(k)(z^* - h(\hat{x}^+(k))) \quad (4.11)$$

5. Posterior State Covariance Update

$$P_x^+(k+1) = (I - K(k)H(k))P_x^-(k+1) \quad (4.12)$$

where z^* is the real-time measurement obtained from sensors.

Before applying EKF to perform estimation, a problem needs to be formulated where discrete-time system equations are built. We first present possibly useful physical quantities for system modeling in Table 4.2. Four types of variables are defined: 1) known prior variables are available after experimental setup, 2) computed variable means they can be algebraically computed directly using measured variables or known prior variables, 3) Measured variable are obtained using sensors that are only available during online experiments, 4) estimated Variables are those needed to be estimated using filters running online. The modeling framework in some of the representative solutions are summarized as

Physical Quantity	Notation	Variable Type
Exact Position	p_i	Estimated/Computed Variable
Exact Height (Altitude)	h_i	Measured Variable
Initial Position	p_i^0	Known Prior Variable
Initial Relative Position	Δp_{ij}^0	Estimated/Computed Variable
Displacement	Δp_i	Estimated Variable
Relative Displacement	Δp_{ij}	Estimated Variable
Relative Distance	d_{ij}	Measured Variable
Relative Position	$\Delta p_{ij} = p_j - p_i$	Estimated Variable
Exact Heading	ϕ_i	Measured Variable
Relative Heading	$\Delta \phi_{ij} = \phi_j - \phi_i$	Estimated/Computed Variable
Transformation Matrix	$R_{ij}(\phi_i, \phi_j)$	Computed Variable
Velocity	v_i	Measured Variable
Relative Velocity	Δv_{ij}	Estimated/Computed Variable

Table 4.1: Physical Quantity for System Modeling, where i, j are the agent identity indices following the convention in 3.4.2.

follows along with their limitations.

Representative RL Solutions Using KF

- **3-D Relative Position Estimation of UAV Using UWB Range Measurements and Altimeter[51]**
 - **State:** $\Delta p_{ij}^0, \Delta p_{ij}$
 - **Measurement:** $d_{ij}, \Delta p_{ij}$
 - **Comments**
Relative heading is not estimated and thus all relative position estimation is assumed in the same global reference frame. Moreover, it is not clearly stated how the exact relative displacement is measured.
- **3-D Relative Pose Estimation of UAV Using Bluetooth Range Measurements and Altimeter[25]**
 - **State:** $\Delta p_{ij}, v_i, v_j, \phi_i, \phi_j, h_i, h_j$
 - **Measurement:** $d_{ij}, v_i, v_j, \phi_i, \phi_j, h_i, h_j$
 - **Comments**
Relative pose is estimated in local coordinate, which is good, however, altimeter is used and thus it is still a 2-D RL.
- **2-D Relative Pose Estimation of Ground Mobile Robots Using UWB Range Measurements [19]**
 - **State:** $\Delta p_{ij}, \Delta \phi_{ij}$
 - **Measurement:** d_{ij}
 - **Comments**
Good solution that directly estimates relative pose with only inter-agent distance measurement, but it is for 2-D localization.

4.1.2. Observability Analysis

After building the system model, we then need to analyze whether the filter can perform state estimation. More specifically, given the limited measurements and the chosen states, can all the interested states be reconstructed using the measurements. This question is answered by evaluating the rank of the observability matrix. For deterministic linear time-invariant (LTI) systems, the observability matrix is also invariant (i.e., both time-independent and state-independent) [76]. For forward Gaussian stochastic LTI systems, the equivalence between the stochastic observability matrix being full-rank and the system being stochastic observable is guaranteed only if the stochastic supportability holds.

For general deterministic nonlinear systems of input-affine form, controllability and observability are defined similarly to those of the LTI systems, yet the controllability/observability matrix is constructed using the *Lie derivatives* [69]. Consequently, the controllability/observability matrix for input-affine nonlinear systems is state-dependent and thus controllability/observability can only hold *locally* in general. For forward Gaussian stochastic nonlinear systems of input-affine form, the equivalence between the stochastic observability matrix being full-rank and the system being stochastic observable is also established only if the stochastic supportability holds. All existing localization solutions (e.g., [100, 119, 94]) analyze the observability *incorrectly* by directly borrowing the rank condition of the observability matrix for deterministic nonlinear systems [62], given the system model for localization is stochastic in nature.

4.1.3. Covariance Intersection

For loosely coupled CRL, only achieving peer-to-peer RL fails to provide convincing results since consistency is lost in most cases, and thus we need cooperation among agents to ensure estimation consistency as well as to improve estimation accuracy.

Consider a demonstrative scenario where a set of 3 agents $\{i, j, k\}$ performs local RL. For simplicity, we only consider relative position estimation at this time. Assume that agent i estimates the relative position Δp_{ij} and Δp_{ik} , and agent k also makes estimation Δp_{kj} . Agent k broadcasts its estimation Δp_{kj} and its heading ϕ_k to agent i such that agent i can recalculate Δp_{kj}^i in its local coordinate. Therefore, the relative position of agent j with respect to agent i estimated by agent i has a *direct* estimation Δp_{ij} and an *indirect* estimation $\Delta p_{ik} + \Delta p_{kj}^i$. Since Δp_{ij} , Δp_{ik} , Δp_{kj} are obtained independently, $\Delta p_{ij} \neq \Delta p_{ik} + \Delta p_{kj}^i$ in general, which causes estimation inconsistency. As a conclusion, fusing direct and indirect estimation is imperative. The major challenge is that the correlation between the direct estimation and indirect estimation is hard to compute and it is not 0 in general by virtual of the motion of agent i itself.

Covariance intersection [74] has been developed to perform information fusion with unknown correlations, the basic algorithm is given as follows.

Covariance Intersection Algorithm

Given random variable a and b as estimations for the same physical quantity with estimated mean being \bar{a} and \bar{b} , respectively. The estimated covariance of a and b are denoted as P_{aa} and P_{bb} , Then a consistent [70] fused estimation c can be computed according to the following equations

$$\begin{aligned} P_{cc}^{-1} &= \lambda P_{aa}^{-1} + (1 - \lambda) P_{bb}^{-1} \\ P_{cc}^{-1} \bar{c} &= \lambda P_{aa}^{-1} \bar{a} + (1 - \lambda) P_{bb}^{-1} \bar{b} \end{aligned} \quad (4.13)$$

where \bar{c} is the estimated mean of c , P_{cc} is the covariance of the fused estimation c , and λ is a tunable weighting parameter.

Covariance intersection has proved to be effective for decentralized sensor fusion, which perfectly fits the problem of decentralized CRL. Existing CL examples exploiting covariance intersection are all used for exact localization [49, 17, 90, 20, 92, 158] where some focus on implementation of the traditional algorithm [17, 49, 90, 20] and the others [92, 158] made contributions on developing variations of covariance intersection algorithm that have improved performance.

4.1.4. Initialization & Extensions

The EKF is widely applied for localization due to its computational efficiency. However, since the EKF only exploits first-order dynamics of the system and produces a suboptimal solution, its estimation performance greatly degrades with initialization parameters being poorly chosen. Therefore, it is beneficial to investigate the initialization impact on the EKF. To get rid of the curse of initialization, extensive contributions have been made to cover higher dynamics of the system and cater to non-Gaussian noise, which will also be elaborated on afterward.

We first highlight the initialization impact in terms of 1) measurement noise covariance matrix $R(0)$, 2) Initial state estimate $\hat{x}^+(0)$ and its corresponding covariance $P_x^+(0)$, 3) augmented process noise covariance matrix $Q(k)$.

Initialization for EKF

- **Measurement Noise Covariance $R(k)$**

$R(k)$ is assumed to be constant in most applications (i.e., $R(k) = R$), and it mainly comes from device imperfections with details readily available from the manufacturer's documentation. A common used form is $R = \text{diag}(k_R \sigma_i^2)$, where σ_i is the nominal standard deviation of the i th measurement and $k_R \geq 1$ is a scaling parameter to model possible device degradation. In general, choosing larger R or k_R makes the filter more *robust to measurement noise*, yet choosing too large values would *underestimate the effect of measurement*, thus making the Kalman gain $K(k)$ *vanish* in limit case.

- **Initial State Estimate $\hat{x}^+(0)$ and Covariance $P_x^+(0)$**

In general, $\hat{x}^+(0)$ and $P_x^+(0)$ are rarely known in practice with sufficient accuracy. However, it can be estimated using the upper ($x(0)_{\max}$) and lower ($x(0)_{\min}$) bound of the initial state $x(0)$, both of which can in practice be reasonably estimated. One simple and consistent framework [127] is formally given as

$$\begin{aligned}\hat{x}^+(0) &= 0.5(x(0)_{\max} + x(0)_{\min}), \\ P_x^+(0) &= 0.25 \text{diag}((x(0)_{\max} - x(0)_{\min})^\top (x(0)_{\max} - x(0)_{\min})).\end{aligned}\tag{4.14}$$

- **Augmented Process Noise Covariance $Q(k)$**

Choosing a reasonable $Q(k)$ is the most challenging part in initialization phase. As identified in 4.2, we have actuator noise $\gamma(k)$ and process noise $w(k)$. First, we can safely handle the actuator noise $\gamma(k)$ the same way as we deal with sensor noise whose uncertainty level is provided in the documentation of the device. As for process noise $w(k)$, its major correlating factors are parametric uncertainty and structure uncertainty. For micro quadrotors, its system dynamics is well established using classical mechanics, thus leading us to confidently only consider parametric uncertainty. The modified process equation is then given as

$$x(k+1) = F(x(k), u^*(k), p_x^*) + B(k)\gamma(k) + B_p(k)\delta p(k),\tag{4.15}$$

where p_x^* is the nominal parameter value, and $B_p(k) = \left. \frac{\partial F}{\partial p} \right|_{\hat{x}^+(k), u^*(k), p_x^*}$, and the covariance of $\delta p(k)$ can be similarly computed as in 4.14 with known bounds p_{\max} and p_{\min} .

Notably, the KF provides the only optimal solution for linear systems, the EKF extends the algorithm to nonlinear systems but it only approximates the optimal solution taking into account only the *first-order* impact [150]. The above flaw of the EKF motivates the development of the UKF algorithm.

UKF specifies the state distribution using a minimal set of carefully chosen sample points (called *sigma points*), with which the *3rd* order impact of the posterior mean and covariance of the state is captured [150] for any nonlinearity. Given random variable $x \in \mathcal{R}^{n_x}$ with its mean \bar{x} and covariance matrix P_x , the sigma points and their corresponding weights are designed as

Sigma Points Generation of UKF

- **Sigma Points**

$$\begin{aligned}
 s_0 &= \bar{x} \\
 s_i &= \bar{x} + \alpha\sqrt{n_x + \kappa}L[:, i], \quad i = 1, 2, \dots, n_x \\
 s_i &= \bar{x} - \alpha\sqrt{n_x + \kappa}L[:, i - n_x], \quad i = n_x + 1, n_x + 2, \dots, 2n_x
 \end{aligned} \tag{4.16}$$

- **Weights for Sigma Points**

- **Weights of the mean**

$$\begin{aligned}
 W_0^m &= [\alpha^2(n_x + \kappa) - n_x]/[\alpha^2(n_x + \kappa)] \\
 W_i^m &= 1/[2\alpha^2(n_x + \kappa)], \quad i = 1, 2, \dots, 2n_x
 \end{aligned} \tag{4.17}$$

- **Weights of the covariance**

$$\begin{aligned}
 W_0^c &= W_0^m + (1 - \alpha^2 + \beta) \\
 W_i^c &= W_i^m, \quad i = 1, 2, \dots, 2n_x
 \end{aligned} \tag{4.18}$$

where L is the square root of the covariance P_x (i.e., $P_x = LL^\top$), α is a tuning variable controlling the spread of the points and β is another tuning variable that helps incorporating prior knowledge of x .

Before applying the algorithm, we first construct $x_a(k) = [x(k)^\top, u(k)^\top, w(k)^\top, v(k)^\top]^\top$. In the following UKF algorithm, we only need to work with the augmented state $x_a(k)$.

UKF Algorithm

Initialization

$$\begin{aligned}
 \hat{x}_a^+(0) &= [\hat{x}^+(0)^\top, u^*(k)^\top, 0, 0]^\top \\
 P_a^+(0) &= \text{diag}\{P_x^+(0), P_\gamma(0), Q(0), R(0)\}
 \end{aligned} \tag{4.19}$$

For $k = 0, 1, 2, \dots$,

1. State and Covariance Propagation

$$\begin{aligned}
 S_a(k) &= \left[\hat{x}_a^+(k), \hat{x}_a^+(k) \pm \alpha\sqrt{n_a + \kappa}L_a \right] \text{ (c.f., (4.16) to (4.18))} \\
 S_a(k+1) &= F(S_a(k)) \text{ (apply to each column of } S_a(k) \text{ c.f., (4.2))} \\
 \hat{x}_a^-(k+1) &= S_a(k+1) \text{vec}\{W_0^m, W_1^m, \dots, W_{n_a}^m\} \\
 P_a^-(k+1) &= \sum_{i=0}^{2n_a} W_i^c (S_a(k+1)[:, i] - \hat{x}_a^-(k+1))(S_a(k+1)[:, i] - \hat{x}_a^-(k+1))^\top \\
 Y_a(k) &= h^*(S_a(k)) \text{ (apply to each column of } S_a(k)) \\
 \hat{y}(k) &= Y_a(k) \text{vec}\{W_0^m, W_1^m, \dots, W_{n_a}^m\}
 \end{aligned} \tag{4.20}$$

2. Kalman Gain Calculation

$$\begin{aligned}
 P_{yy}(k) &= \sum_{i=0}^{2n_a} W_i^c (Y_a(k)[:, i] - \hat{y}(k))(Y_a(k)[:, i] - \hat{y}(k))^\top \\
 P_{x-y}(k) &= \sum_{i=0}^{2n_a} W_i^c (S_a(k+1)[:, i] - \hat{x}_a^-(k+1))(Y_a(k)[:, i] - \hat{y}(k))^\top \\
 K(k) &= P_{x-y}(k)P_{yy}(k)^{-1}
 \end{aligned} \tag{4.21}$$

3. Update with New Measurement

$$\begin{aligned}\hat{x}_a^+(k+1) &= \hat{x}_a^-(k+1) + K(k)(z^* - \hat{y}(k)) \\ P_a^+(k+1) &= P_a^-(k+1) - K(k)P_{yy}(k)K(k)^\top\end{aligned}\quad (4.22)$$

where measurement model $h^*(x, v) := h(x) + Dv$ is defined as a general measurement equation whose effective attributes are state x and measurement noise v .

As shown in the above algorithm, the UKF scales poorly when the system dimension is high and it also requires computing the square root of a high-dimension matrix. On the other hand, apart from its ability to capture higher nonlinearity, UKF also eliminates the computation of Jacobian and Hessian matrices. We summarize and compare the major properties of the EKF and the UKF in the following Table 4.2. As pointed out in Table 4.2, the UKF uses many sigma points equal to $2(n_x + n_u + n_w +$

	EKF	UKF
Matrix Computation	Closed-form Jacobian and Heissian	Matrix Square Root
Dimension Scalability	Good	Poor
High Nonlinearity	No	Yes
Estimation Accuracy	Fine	Good
Sensitivity to Initialization	Sensitive	Not Sensitive

Table 4.2: Comparison of EKF and UKF

$n_v) + 1$, which significantly increases computational complexity and desired memory storage. Another drawback is the computation of the square root of a given matrix of high dimensions. To overcome the above-mentioned issues, the cubature Kalman filter (CKF) and its slight extension square-root CKF (SCKF) was developed [5]. The CKF (SCKF) also needs to propagate a bunch of data points of size $2(n_x + n_u + n_w + n_v)$, thus making it less preferable for online implementation. Notably, the idea of propagating the square root of a given matrix is effective for online computation and hardware coding.

Another direction to improve the KF is to expand its capability for handling non-Gaussian noise. Recent contributions leverage other types of error metrics (e.g., centered error entropy (CEE) [161], minimum error entropy (MEE) [21, 29]) instead of the commonly used minimum mean squared error (MMSE).

4.2. Particle Filter Based Cooperative Localization

In this section, we put our attention on the PF and some of the RL solutions using the PF. We will first present an introduction to the statistical theory behind the PF and then move on to some illustrative localization examples.

4.2.1. Introduction to Particle Filter

The PF is closely related to the Bayesian filtering theory for Markov processes whose theoretic principles will be illustrated first as preparation. In the sequel, we use $Z(k) = \{z(0), z(1), \dots, z(k)\}$ to denote the concatenated sequence of measurement up to time instant k . For simplicity, we do not distinguish actuator noise ($\gamma(k)$) and process noise ($w(k)$) and only use the augmented process noise $\eta(k)$ as the *process noise*. Given a discrete-time dynamical system following process equation (4.2) and the general output equation in (4.20), we have a discrete-time Markov process of the following form:

$$\begin{aligned}x(k+1) &= F(x(k), u(k), \eta(k)), \\ y(k) &= h(x(k), v(k)).\end{aligned}\quad (4.23)$$

Since $\eta(k)$ is stochastic, the state $x(k)$ is also a stochastic process that can be assigned with a probability distribution function (pdf) at each time instant k . In this section, $p(\cdot)$ will be the notation for any pdf functions. Alongside with Markov property, we have the following relation:

$$\begin{aligned}p(x(k+1)|X(k), Z(k)) &= p(x(k+1)|x(k)) \\ p(z(k)|X(k)) &= p(z(k)|x(k))\end{aligned}\quad (4.24)$$

A slightly modified Bayesian rule that fits our problem would be

$$p(A|B, C) = \frac{p(B|A, C)p(A|C)}{p(B|C)} = \frac{p(B|A, C)p(A|C)}{\int p(B|A, C)p(A|C)dA} \quad (4.25)$$

The Bayesian filter attempts to make the posterior state estimation by computing the conditional pdf $p(x(k)|Z(k))$, which is interpreted as the distribution information of the current state given all the previous information of the measurement up to the current instant. The recursive algorithm of the Bayesian filter is illustrated as follows.

Bayesian Filter Algorithm

1. Initialization:

Given the first measurement, and the pdf of the measurement noise $v(0)$, compute the posterior state estimation $p(x(0)|Z(0))$.

2. State Propagation (Prior Estimation):

Given the posterior state estimation $p(x(k)|Z(k))$ at step k , an initial estimate of the next state is given as

$$\begin{aligned} p(x(k+1)|Z(k)) &= \int p(x(k+1), x(k)|Z(k))dx(k) \\ &= \int p(x(k+1)|x(k), Z(k))p(x(k)|Z(k))dx(k) = \int p(x(k+1)|x(k))p(x(k)|Z(k))dx(k) \end{aligned} \quad (4.26)$$

where $p(x(k+1)|x(k))$ is obtained from the process equation

3. Measurement Incorporation (Posterior Estimation):

In the Bayesian rule (4.25), let $A = x(k+1)$, $B = z(k+1)$, and $C = Z(k)$, we obtain

$$\begin{aligned} p(x(k+1)|Z(k+1)) &= \frac{p(z(k+1)|x(k+1), Z(k))p(x(k+1)|Z(k))}{\int p(z(k+1)|x(k+1), Z(k))p(x(k+1)|Z(k))dx(k+1)} \\ &= \frac{p(z(k+1)|x(k+1))p(x(k+1)|Z(k))}{\int p(z(k+1)|x(k+1))p(x(k+1)|Z(k))dx(k+1)} \end{aligned} \quad (4.27)$$

where $p(x(k+1)|Z(k))$ is provided by the results of prior estimation.

In practice, the integral in (4.26) and (4.27) can only be computed under strong assumptions and may also increase computational complexity if the state dimension is high. Therefore, numerical methods need to be developed to fix this problem, which leads to the famous PF. The basic idea of the particle filter is approximating the probability distribution using a set of discrete particles. We present the details of the algorithm as follows:

PF Derivation (without Sampling)

1. Initialization:

Given the first measurement, and the pdf of the measurement noise $v(0)$, choose a set of N_s samples $x_i^+(k)$ that obeys the initial pdf $p(x(0)|Z(0))$. In general, each dimension l of $x_i(k)$ requires m_l sample points, thus the total number of samples is $N_s = \prod_{l=1}^{n_x} m_l$.

2. State Propagation (Prior Estimation):

Given the approximated posterior state estimation $p(x(k)|Z(k)) = \sum_{i=1}^{N_s} \alpha_i^k \delta(x(k) - x_i^+(k))$ at step k with $\sum_{i=1}^{N_s} \alpha_i^k = 1$ and $\delta(\cdot)$ being the generalized Dirac pulse function, the prior

estimate of the next state is given as

$$\begin{aligned}
p(x(k+1)|Z(k)) &= \int p(x(k+1)|x(k))p(x(k)|Z(k))\mathbf{d}x(k) \\
&= \int p(x(k+1)|x(k)) \sum_{i=1}^{N_s} \alpha_i^k \delta(x(k) - x_i^+(k)) \mathbf{d}x(k) \\
&= \sum_{i=1}^{N_s} \alpha_i^k \int p(x(k+1)|x(k)) \delta(x(k) - x_i^+(k)) \mathbf{d}x(k) \\
&= \sum_{i=1}^{N_s} \alpha_i^k p(x(k+1)|x(k) = x_i^+(k)) \\
&= \sum_{i=1}^{N_s} \alpha_i^k \delta(x(k+1) - F(x_i^+(k), u^*, \eta_i(k))) \\
&:= \sum_{i=1}^{N_s} \alpha_i^k \delta(x(k+1) - x_i^-(k+1))
\end{aligned} \tag{4.28}$$

where $\eta_i(k)$ is a sampled noise according to its distribution

3. Measurement Incorporation (Posterior Estimation):

Given the prior estimation, we construct the posterior estimation using the Bayesian rule by substituting (4.28) into (4.27)

$$\begin{aligned}
p(x(k+1)|Z(k+1)) &= \frac{p(z(k+1)|x(k+1))p(x(k+1)|Z(k))}{\int p(z(k+1)|x(k+1))p(x(k+1)|Z(k))\mathbf{d}x(k+1)} \\
&= \frac{p(z(k+1)|x(k+1)) \sum_{i=1}^{N_s} \alpha_i^k \delta(x(k+1) - x_i^-(k+1))}{\int p(z(k+1)|x(k+1)) \sum_{i=1}^{N_s} \alpha_i^k \delta(x(k+1) - x_i^-(k+1)) \mathbf{d}x(k+1)} \\
&= \frac{\sum_{i=1}^{N_s} \alpha_i^k p(z(k+1)|x_i^-(k+1)) \delta(x(k+1) - x_i^-(k+1))}{\sum_{i=1}^{N_s} \alpha_i^k \int p(z(k+1)|x(k+1)) \delta(x(k+1) - x_i^-(k+1)) \mathbf{d}x(k+1)} \\
&= \frac{\sum_{i=1}^{N_s} \alpha_i^k p(z(k+1)|x_i^-(k+1)) \delta(x(k+1) - x_i^-(k+1))}{\sum_{i=1}^{N_s} \alpha_i^k p(z(k+1)|x_i^-(k+1))} \\
&= \sum_{i=1}^{N_s} \frac{\alpha_i^k p(z(k+1)|x_i^-(k+1))}{\sum_{j=1}^{N_s} \alpha_j^k p(z(k+1)|x_j^-(k+1))} \delta(x(k+1) - x_i^-(k+1))
\end{aligned} \tag{4.29}$$

where we further define $\frac{\alpha_i^k p(z(k+1)|x_i^-(k+1))}{\sum_{i=1}^{N_s} \alpha_i^k p(z(k+1)|x_i^-(k+1))}$ as α_i^{k+1} which is nothing but the updated weights. The resulting posterior estimation is now using $x_i^-(k+1)$ as particles but their weights is adjusted with the new measurements.

An unresolved important issue involved in PF is how to (re)sample particles given an approximated distribution instead of the real continuous one. Sampling techniques are not just useful for initialization but also for avoiding degeneracy problems [52]. We here present several efficient sampling algorithms that are commonly used in engineering applications [36]

Sampling Algorithms [63]

All the sampling methods explained below do not impose constraints on the number of particles (i.e., the number of particles before and after sampling can be different) and they all rely on cumulative sum of the weights α_i^k . First, the desired number of particles at step k is denoted as $N_s(k)$. Besides, we define the cumulative weights $\{Q_i^k\}_{i=1}^{N_s(k)}$ with $Q_i^k = \sum_{i'=1}^i \alpha_{i'}^k$.

- **Multinomial Sampling:**

Generate $N_s(k+1)$ points $u_i \sim U[0,1]$. Then for each u_i , search for index j such that $Q_{j-1}^k \leq u_i < Q_j^k$. The posterior sampled points are $x_i^+(k+1) = x_j^-(k+1)$.

- **Stratified Sampling:**

Generate $N_s(k+1)$ points u_i obeys the rule $u_i = (i-1)/N_s(k+1) + \tilde{u}_i$, where $\tilde{u}_i \sim U[0,1/N_s(k+1)]$. Then do Multinomial sampling.

- **Systematic Sampling:**

Generate $N_s(k+1)$ points u_i obeys the rule $u_i = (i-1)/N_s(k+1) + \tilde{u}^*$, where $\tilde{u}^* \sim U[0,1/N_s(k+1)]$. Then do Multinomial sampling.

Multinomial sampling directly generates points that obey the uniform distribution, whereas stratified sampling and systematic sampling both first divide the $[0,1]$ into $N_s(k+1)$ intervals and pick points in those intervals. Further, stratified sampling picks those points in a stochastic way for each interval, while systematic sampling generates a uniform *shift* for all intervals. Regardless of the methods, after sampling, all the weights are reset as $\alpha_i^{k+1} = 1/N_s(k+1)$

It is clear to see that systematic sampling is the most computational efficient since it only requires generating 1 points that obey the uniform distribution. Another strategy is to apply sampling in an *event-triggered* manner and only perform sampling when $x_i^-(k)$ is not informational enough. There are various criteria to determine the number of effective filters using the weights α_i^k [101].

Determining weights using measurements and maximum likelihood (ML) is a commonly used method. Properly weighing particles is the most crucial factor that influences the performance of the PF. There are other alternatives to determine weights, among which a big family combines the EKF or the UKF with the PF where each particle is regarded as a refined estimated state using the EKF or the UKF [146]. Nonetheless, these approaches have great computational complexity and thus are not suitable for distributed state estimation for MAVs.

Still, combining the KF and the PF to exploit their advantages each other is a favorable choice. The most famous hybrid solution is the Rao-blackwellized particle filter (RBPF) which handles the *curse of dimensionality* of the PF.

RBPF algorithm (simplified)

- **State Decomposition:**

Decompose states into linear ones x^L and nonlinear ones x^R , where the dynamics of nonlinear states only relates to the nonlinear states themselves and after fixing the nonlinear states, the remaining dynamics of the linear states is linear.

- **PF for Nonlinear States:**

Perform PF on nonlinear states given the posterior estimation of both linear and nonlinear states

- **KF for linear States:**

Perform standard KF on linear states given the estimated nonlinear states obtained from PF

Notably, the traditional RBPF works better for systems where the measurement function explicitly depends on both the linear and nonlinear states. For the 3-D peer-to-peer RL problem with distance measurement, the naive measurement model only depends on the relative distance without touching on the relative heading (yaw). Besides, Gaussian measurement noise is a valid assumption but unsuitable for UWB measurements. To apply RBPF to our interested problem, we present some potential solutions listed below.

Insights on Applying RBPF for 3-D peer-to-peer RL

- **Modifying the Measurement Model:**

The UWB unit is not placed at the geometric center of the MAV, thus making the distance measurement between two UWB units also a function of the headings. This modification will expand the observability space.

- **Integrating with EKF:**

Regardless of modifying the measurement model or not, the measurement model is a nonlinear mapping to the states. Thus, RBPF needs to combine EKF. The satisfactory trait of using RBPF is now the nonlinearity only occurs in measurement function and it is quadratic which has at most second-order nonlinearity.

- **Incorporating other forms of KF:**

It is of great interest to combine particle filter with MEE-KF or CEE-KF to improve performance against non-Gaussian noise.

4.2.2. Relative Localization Using Particle Filter

After introducing the theoretical foundations of PF, we now present some applications for relative localization using PF. To see different steps of implementing PF in practice, we first give a succinct summary of the entire PF algorithm.

PF algorithm (Complete Version)

Initialization:

Initialize the particles based on the distribution of the initial state $p(x^+(0))$.

For each step $k = 0, 1, 2, \dots$

1. Particle Propagation:

Generate the $N_s(k)$ particles $x_i^-(k+1)$ according to the dynamics with randomly chosen noise particles $\eta_i(k)$

$$x_i^-(k+1) = F(x_i^+(k), u^*, \eta_i(k)) \quad (4.30)$$

2. Weight Update:

Given the new measurements $z(k+1)$, update the weights for each particles using the likelihood

$$\alpha_i^{k+1} = \frac{\alpha_i^k p(z(k+1)|x_i^-(k+1))}{\sum_{j=1}^{N_s} \alpha_j^k p(z(k+1)|x_j^-(k+1))} \quad (4.31)$$

3. Particle Resampling:

Determine whether resampling is required based on the number of effective particles $N_{\text{eff}} = [\sum_{i=1}^{N_s(k)} (\alpha_i^k)^2]^{-1}$ (or alternative criteria). Then resample $N_s(k+1)$ particles $x_i^+(k+1)$ based on the updated weights if needed. If resampling is not necessary, simply keep $x_i^+(k+1) = x_i^-(k+1)$ for the next iteration.

4. Implication:

The mean ($\hat{x}(k+1)$) and variance ($\hat{\sigma}_x^2(k+1)$) of the state is given as

$$\begin{aligned} \hat{x}(k+1) &= \sum_{i=1}^{N_s(k+1)} \alpha_i^{k+1} x_i^-(k+1) \\ \hat{\sigma}_x^2(k+1) &= \sum_{i=1}^{N_s(k+1)} \alpha_i^{k+1} (x_i^-(k+1) - \hat{x}(k+1))(x_i^-(k+1) - \hat{x}(k+1))^T \end{aligned} \quad (4.32)$$

One early work using particle filter for distance-based localization aims for exact localization [95], and their localization is computed in a centralized framework with state augmentation. Specifically, at each iteration, a large group of particles is created for the augmented states which contain the 2-D pose information of all the agents. Consequently, the number of particles is $(3N)^{m_p}$ where N is the number of agents and m_p is the number of particles assigned to each state, rendering a huge computational cost. Besides, though they are motivated to favor the PF over the KF for dealing with non-Gaussian noise, their noise model is still Gaussian.

An improved decentralized 2-D RL solution using a particle filter has been developed recently [93]. In this work, each agent receives multiple distances from all its neighbors and constructs relative pose estimation in the local frame for all the neighboring agents in a peer-to-peer manner. Notably, they initialize the particle filter using the relative pose estimation from the geometric kinematics of any pair

of agents with a sequence of distance measurements between them, which is solved by weighted least square (WLS) optimization [144]. Besides, in the particle filtering process, an average consensus auxiliary variable [160] is introduced to assist the particle filter. Still, this work assumes too simple dynamics of the agents and Gaussian distributed noise.

Other PF-based localization solutions [48, 64, 46] do not fit our framework in that they either use more than one UWB unit, fixed beacons, or assume relative pose is directly measured. Therefore, we do not elaborate more details on them at this point.

Unlike most of the existing solutions [64, 93, 95, 48] where the emphasis is put on applying PF theory to the system models, there are some attempts on improving the PF algorithm for localization. Since the basic principle of utilizing particles to approximate the posterior distribution itself is computationally inefficient, it is more practical to modify the baseline of the PF without introducing more complex tools to PF. An interesting modification is swapping the sequence of applying new measurement information and propagating through dynamics, which leads to the dual PF (DPF) and the mixture PF (MPF) [135].

4.3. Moving Horizon Estimation Based Cooperative Relative Localization

Bayesian filter, owing to its generic theoretical framework, leads to many different variations. Specifically, when assuming linearity of system dynamics and Gaussian distributed noise, the Bayesian filter becomes the KF; while if approximating prior and posterior distributions using particles, the Bayesian filter becomes the PF. In this section, we introduce the MHE which is another popular variation of the Bayesian filter. Same as in previous sections, we will first introduce the basic theory of the MHE, recent advances in the MHE, and practical issues involved when applying the MHE. Subsequently, we present several examples built on the MHE framework.

4.3.1. Introduction to Moving Horizon Estimation

We start from some of the intermediate results of the Bayesian filter algorithm as in (4.26) and (4.27). First, substituting (4.26) into (4.27), we obtain

$$p(x(k+1)|Z(k+1)) \propto [p(z(k+1)|x(k+1))p(x(k+1)|x(k))]p(x(k)|Z(k)). \quad (4.33)$$

From 4.33, by mathematical induction, we have the following result

$$p(x(k+1)|Z(k+1)) \propto p(x(0)|z(0)) \prod_{i=0}^k p(x(i+1)|x(i)) \prod_{i=0}^k p(z(i+1)|x(i+1)). \quad (4.34)$$

Then from the probabilistic perspective, the best posterior estimate of state $x(k+1)$ would be

$$\begin{aligned} \hat{x}(k+1) &= \arg \max_{x(k+1)} p(x(k+1)|Z(k+1)) \\ &= \arg \max_{x(k+1)} p(x(0)|z(0)) \prod_{i=0}^k p(x(i+1)|x(i)) \prod_{i=0}^k p(z(i+1)|x(i+1)) \\ &= \arg \max_{x(k+1)} \log p(x(0)) + \sum_{i=0}^k \log p(x(i+1)|x(i)) + \sum_{i=0}^k \log p(z(i+1)|x(i+1)). \end{aligned} \quad (4.35)$$

Intuitively but not rigorously, $p(x(i+1)|x(i))$ and $p(z(i+1)|x(i+1))$ are characterized by distribution of the augmented process noise ($\eta(i)$) and measurement noise ($v(i)$), respectively. Consequently, we are interested in maximizing an objective function of the following form

$$J(x(0), \{\eta(i)\}_{i=0}^k) = L(\{\eta(i)\}_{i=0}^k, \{v(i)\}_{i=0}^{k+1}) + \Gamma_0(x(0)), \quad (4.36)$$

where $v(i)$ is explicitly shown in the objective but can be implicitly expressed as a function of the measurement $z(i)$ and states $x(i)$. Further, the states $x(i)$ can be implicitly expressed using $x(0)$ and $\{\eta(i)\}_{i=0}^{i-1}$. Thus, the final decision variables of the objective function is $\{x(0), \{\eta(i)\}_{i=0}^k\}$.

The formulated objective function takes all past measurements up until step $k+1$ as parameters, rendering the optimization problem more difficult to solve as more measurements are available. To

resolve this issue, the dynamic programming principle is borrowed, and the resulting objective function with truncated state trajectory is given as

$$J(x(k-W+1), \{\eta(\iota)\}_{\iota=k-W+1}^k) = L\left(\{\eta(\iota)\}_{\iota=k-W+1}^k, \{v(\iota)\}_{\iota=k-W+2}^{k+1}\right) + \Gamma_{k-W}\left(x(k-W+1), \hat{x}(k-W+1)\right), \quad (4.37)$$

which can also be deduced from applying induction only W times backward according to (4.33). Correspondingly, the best estimate $\hat{x}(k)$ is obtained from the latest W measurements (i.e., $z(k-W+1)$ to $z(k)$) and an estimated state $\hat{x}(k-W)$ by solving an optimization problem. We call $L(\cdot)$ the *measurement cost* and $\Gamma_{k-W}(\cdot)$ the *arrival cost*.

Solving the optimization problem returns an updated estimate for $x(k-W+1)$ and a *best* noise sequence $\{\eta(\iota)\}_{\iota=k-W}^{k-1}$, with which an optimal state trajectory $\hat{x}(k), \hat{x}(k-1), \dots, \hat{x}(k-W+1)$ can be inferred. At the next iteration, $\hat{x}(k-W+1)$ will be used as the parameter for the new arrival cost, and the new measurement $z(k+1)$ will serve as one of the parameters for the measurement cost. In summary, the state estimation is performed with a moving horizon of size W , which is the reason why we call this estimation strategy moving horizon estimation.

The MHE intrinsically performs state estimation using optimization, which makes incorporating constraints in the optimization formulation very convenient. In addition, it also shows better performance against EKF when dealing with nonlinear systems since it theoretically aims to find the optimal estimation instead of a sub-optimal one. The major curse is that the computational cost will significantly increase if the system is nonlinear. Moreover, if the function describing the system dynamics and/or the measurement model is non-convex, the optimization may fall into local optimum and fails to return a satisfactory estimation.

Remark 1 (Relation between MHE and KF) *The MHE not only provides an estimate of the current state, but it also serves as a smoother which improves the estimation for previous states. Under the conditions that the system is linear with Gaussian noise disturbance, and the window size is 1, the MHE degenerates to the KF.*

Based on (4.37), we move one step further and present the commonly used MHE optimization formulation [102, 43] with quadratic norm.

MHE Optimization Formulation (Quadratic Norm)

Assume that the measurement noise is *additive* and the augmented process noise is only consisted of actuator noise, obtaining the latest state estimation $\hat{x}(k)$ without smoothing can be achieved by solving the following optimization problem

- **Objective Function:**

$$J(\{x(\iota)\}_{\iota=k-W}^k, \{u(\iota)\}_{\iota=k-W}^{k-1}) = \|x(k-W) - \hat{x}(k-W)\|_{P_x^{-1}(k-W)} + \sum_{\iota=k-W}^k \|z^*(\iota) - z(\iota)\|_{R^{-1}(\iota)} + \sum_{\iota=k-W}^{k-1} \|u^*(\iota) - u(\iota)\|_{Q^{-1}(\iota)} \quad (4.38)$$

- **Constraints:**

- **Dynamic Constraints:**

$$x(\iota+1) - F(x(\iota), u(\iota)) = 0, \forall \iota \in \{k-W, k-W+1, \dots, k-1\}$$

- **Measurement Constraints:**

$$z(\iota) - h(x(\iota)) = 0, \forall \iota \in \{k-W, k-W+1, \dots, k\}$$

- **Input Constraints:**

$$\underline{u} \leq u(\iota) \leq \bar{u}, \forall \iota \in \{k-W, k-W+1, \dots, k-1\}$$

- **State Constraints:**

$$\underline{x} \leq x(\iota) \leq \bar{x}, \forall \iota \in \{k-W, k-W+1, \dots, k\}$$

Remark 2 (MHE with non-Gaussian noise) *Most of the existing literature assumes that the cost function of the MHE is of the quadratic form (i.e., using quadratic norm to characterize the distance), which is generally NOT true if the system is perturbed by non-Gaussian noise. More importantly, they also assume that the noise is single-modal, while current MHE cannot handle multi-modal noise distributions. Nonetheless, MHE still exhibits convincing results even in systems with non-Gaussian noise since quadratic norm can capture the tendency of the true single-modal (non-Gaussian) distribution.*

In general, moving horizon theory for state estimation of nonlinear systems has three major challenges: 1) how to formulate a statistically rational arrival cost without requiring too much extra computation, 2) how to solve the general nonlinear optimization online 3) how to prove the stability of the estimator. For swarm applications or multi-agent systems, designing distributed MHE algorithms with consensus guarantees becomes an additional challenge.

The first challenge on arrival cost is trivial if neglecting process noise [113]. For general systems with actuator noise and/or parametric uncertainty, determining the uncertain level (i.e., the covariance matrix $\hat{P}_x(k-W)$) is difficult since this information is not provided by the MHE itself. Most of the existing solutions are using auxiliary nonlinear filters which can provide covariance information. Representative examples either utilize the EKF [140], the UKF [114], or the PF [145, 98].

Solving online optimization and achieving estimator stability are closely related in that the stability of the MHE is not determined by the formulated optimization problem solely. In general, this problem boils down to nonlinear programming, thus leading to gradient-based methods [1, 105]. There are also solutions that apply a feed-forward neural network [2] to help accelerate the computation.

Distributed MHE theory is an appealing extension for the localization of multi-agent systems. However, many current algorithms focus on estimation for linear systems [41, 39, 10, 54]. For nonlinear systems, either additional consensus estimator [40] or auxiliary observer [164] are designed for each agent, suggesting the MHE itself cannot handle nonlinear multi-agent systems with satisfactory results.

4.3.2. Relative Localization Using Moving Horizon Estimation

One early work that applies the MHE for RL deals with mobile robots [102]. Their proposed solution typically relies on the ACADO toolbox [43] designed for MHE, and they provided no theoretical contribution regarding the localization problem. More importantly, their framework relies on both distance measurements and bearing measurements, which fails to fit our requirements of only permitting distance measurements. Another more suitable application for localization of mobile robots also applies average consensus to assist the MHE [133].

For UAV applications, an MHE-EKF algorithm is proposed for CL of multiple UAVs [153]. Still, their framework contains a ground station, and each UAV is also equipped with an altimeter to retrieve height information, thus making it also unsuitable for our problem where both ground station and altimeter are unavailable. Another similar work for the AUV application also relies on a depth measurement device [152] which is the counterpart of the altimeter for aerial vehicles.

4.4. Comparison of Different Filters

As a summary of previous sections where details of filters are examined, we compare different filters in this section from various aspects. We finally present possible directions for developing nonlinear filters for improved localization performance.

We first present a concise table where all filters are compared from different evaluation criteria. In Table 4.3, for example, nonlinearity means *the ability to handle nonlinearity*, the same interpretation also works for non-Gaussian noise and multimodal noise. The ability level is generally categorized as *good*, *poor*, and *limited*, where *limited* typically means a method requires case-by-case analysis (i.e., the performance differs per application).

Insights on Developing Nonlinear Filtering Theory for CRL

- Combine PF and EKF, apply PF to the relative heading and apply EKF for the position states.
- Leverage other error norms (e.g., CEE, MEE) and combine it with EKF
- Transform the non-convex dynamics and convex ones and apply KKT condition to explicitly

	EKF	UKF	PF	MHE
Nonlinearity	Poor	Good	Good	Good
Non-Gaussian Noise	Poor	Poor	Good	Limited
Multimodal Noise	Poor	Poor	Good	Poor
Source of Computational Curse	None	Sigma Points	Particles	Arrival Cost & Nonlinear Optimization
Convergence	Limited	Limited	Good	Limited
Sensitivity to Initialization	High	Medium	Low	Limited
Smoothing	No	No	No	Yes
Incorporating Constraints	No	No	No	Yes

Table 4.3: Comparison of Different Filters

solve the optimization problem for MHE.

- Apply MHE in an event-triggered manner.
- Apply consensus algorithm in an event-triggered manner.

5

Graph Rigidity Theory Based Cooperative Relative Localization

This chapter will introduce another big family of localization solutions where geometry is of our prior interest. Due to our hardware setting, inter-agent distance is the only available exteroceptive information, thus we focus on geometric properties relating to *lengths* and neglect angles. Based on the graph formulation of the swarm where agents are *nodes(joints)* and the communication links are *edges(bars)*, the resulting problem is the well-known *geometric rigidity problem* with an old history dated back to 1776 when the famous mathematician Euler made conjectures about the rigidity of polyhedrons. During past decades, many promising results have been presented in terms of rigidity theory for *bar-and-joint* frameworks, which further benefits several research areas ranging from swarm robotics to structural biology [132]. This chapter starts with a rather mathematical section where essential definitions, propositions, and theorems of rigidity theory are provided. Equipped with the mathematical foundations, we then analyze some of the RL solutions using rigidity theory. Subsequently, we delve into the details of the optimization tools used in solving rigidity problems. The final section of the chapter will discuss miscellaneous topics in rigidity theory such as clock rigidity for UWB localization, rigidity-assisted filtering techniques, and rigidity in barycentric coordinate.

5.1. Introduction to Graph Rigidity Theory

We first revisit some of the concepts in graph theory that have been introduced in 3.4.2. Only consider the distances between agents, a group of agents can be described by an *undirected graph* $\bar{\mathcal{G}} = \{\mathcal{V}, \bar{\mathcal{E}}\}$ where $\mathcal{V} = \{1, 2, \dots, N\}$ is the set of nodes representing the agents, and $\bar{\mathcal{E}} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges that specify all measured distances. In the sequel, we will drop the upper bar notation (*cdot*) and simply use $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ to denote the undirected graph. The distance of edge $e_{ij} \in \mathcal{E}$ is denoted as l_{ij} , ($l_{ij} > 0$).

Definition 7 (Bar-and-joint Configuration) *Given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with the distance l_{ij} of all edges $e_{ij} \in \mathcal{E}$ specified, a d -dimensional bar-and-joint configuration is a mapping $p : \mathcal{V} \mapsto \mathbb{R}^{Nd}$ such that $\forall i, j \in \mathcal{V}, (i, j) \in \mathcal{E}, p(i) \neq p(j)$.*

Definition 8 (Bar-and-joint Framework) *Given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, a d -dimensional bar-and-joint framework is a pair $P := (\mathcal{G}, p)$ where p is a configuration.*

Definition 9 Bar-and-joint Linkage *Given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with the distance l_{ij} of all edges $e_{ij} \in \mathcal{E}$ specified, a d -dimensional bar-and-joint linkage is a pair (\mathcal{G}, δ) where δ is a mapping $\delta : \mathcal{V} \mapsto \mathbb{R}^{Nd}$ such that $\forall i, j \in \mathcal{V}, (i, j) \in \mathcal{E}, \|\delta(i) - \delta(j)\| = l_{ij}$.*

Given a graph \mathcal{G} and a fixed dimension d , there are infinitely many frameworks. To clarify the relation among the frameworks, we need the following definitions.

Definition 10 (Equivalent Frameworks (Equivalence)) *Two frameworks $P_1 = (\mathcal{G}, p_1)$ and $P_2 = (\mathcal{G}, p_2)$ are equivalent if $\forall (i, j) \in \mathcal{E}, \|p_1(i) - p_1(j)\| = \|p_2(i) - p_2(j)\|$.*

Definition 11 (Congruent Frameworks (Congruence)) Two frameworks $P_1 = (\mathcal{G}, p_1)$ and $P_2 = (\mathcal{G}, p_2)$ are congruent if $\forall i, j \in \mathcal{V}, \|p_1(i) - p_1(j)\| = \|p_2(i) - p_2(j)\|$.

Remark 3 (Equivalence & Congruence) Equivalence between two frameworks is characterized only by the length of the bars in that framework, while congruence is a stronger condition that requires distance equivalence to be held for all pairs of joints in the framework. Apparently, any congruent frameworks are also equivalent but not vice versa.

Another important aspect of a framework is how it changes in \mathbb{R}^d and then results in a new framework. Therefore, we define the concept of motion of Euclidean spaces as well as the motion of frameworks.

Definition 12 (Motion (Continuous Flex)) A motion of \mathbb{R}^d is a continuous family $\Phi(t)$ of isometries of \mathbb{R}^d such that $\Phi(0)$ is the identity isometry. Without loss of generality, we can restrict the starting time and ending time of the motion to be 0 and 1. Correspondingly, motion is a mapping $[0, 1] \mapsto \text{Iso}(\mathbb{R}^d)$.

Definition 13 (Motion (Continuous Flex) of Frameworks) A motion of any framework $P = (\mathcal{G}, p)$ is a continuous family of frameworks $P(t) := (\mathcal{G}, p(t))$ that are all equivalent to $(\mathcal{G}, p), \forall t \in [0, 1]$ such that $P(0) = P$. In addition, we call a motion of a framework to be trivial if it is induced by a motion of \mathbb{R}^d (i.e., $P(t) = \Phi(t) \circ P, \forall t \in [0, 1]$)

Definition 14 (Framework Space) Given a Linkage (\mathcal{G}, δ) , all the corresponding congruent frameworks that permit up to trivial motions and satisfy the distance constraints imposed by δ comprise a class. All classes further comprise a framework space $\mathcal{F}(\mathcal{G}, \delta)$.

With the above definitions, we are able to concisely define rigidity for bar-and-joint frameworks.

Definition 15 ((Generic) Rigidity) A framework is rigid if one of the following condition holds

- All the admissible motions of the framework are trivial.
- Implicitly introducing the distance mapping δ and the linkage (\mathcal{G}, δ) , the corresponding framework space $\mathcal{F}(\mathcal{G}, \delta)$ is of 0-dimension.

otherwise, it is called flexible.

The above-defined rigidity is commonly referred to as the *generic rigidity* (i.e., rigidity of *generic frameworks*) since it is desired to characterize rigidity as a property of the graph itself instead of the configuration. That also motivates why always explicitly keep the notation \mathcal{G} in the definition of framework and why favor framework space over configuration space. The term *generic* means that there does not exist a non-zero polynomial $h : \mathbb{R}^{Nd} \mapsto \mathbb{R}$ such that $h(p(1), p(2), \dots, p(N)) = 0$ (e.g., colinear points). From a probability perspective, generic is equivalent to having no polynomial algebraic relation in the framework space is *almost surely*. A stronger definition of rigidity attempts to eliminate the differences between equivalent classes of frameworks, which is given as follows.

Definition 16 (Global rigidity) A framework is globally rigid if one of the following condition holds

- All equivalent frameworks are congruent.
- Implicitly introducing the distance mapping δ and the linkage (\mathcal{G}, δ) , the corresponding framework space $\mathcal{F}(\mathcal{G}, \delta)$ has a single point if taking all frameworks in a congruent class to be the same).

otherwise, it is only rigid or even flexible.

Remark 4 ((Global) Rigidity of Linkages) Given a linkage (\mathcal{G}, δ) , it directly inherits the (global) rigidity from its corresponding frameworks or framework space.

Definition 17 (Local Rigidity) We define local rigidity as the rigidity of the frameworks of the sub-graphs.

Remark 5 Some textbooks or papers use local rigidity as the definition for the (generic) rigidity and simply use rigidity as the definition for the global rigidity.

Definition 18 (Minimally Rigid & Redundant Rigid) A rigid framework is *minimally rigid* if it becomes flexible after removing any edge from its original graph. On the other hand, *redundant rigidity* is defined as a rigid framework maintains rigidity upon the removal of any edge.

In general, it is co-NP hard to check whether a given framework is rigid or not for $d \geq 2$ [7], though it is only proved for $d \geq 1$ [125]. Luckily, the *tangent space* of $\mathcal{F}(\mathcal{G}, \delta)$ provides tractable and algebraic properties of rigidity, which will be introduced next.

Definition 19 (Tangent Space of Framework Space) The tangent space $\partial\mathcal{F}(\mathcal{G}, \delta)$ is formally given as

$$\partial\mathcal{F}(\mathcal{G}, \delta) = \{(\mathcal{G}, p) | (p(i) - p(j))(v(i) - v(j))^\top = 0, \forall (i, j) \in \mathcal{E}\} \quad (5.1)$$

where $v(i)$ is defined as $v(i) := \partial p(i)$, which is the derivative of the mapped position with respect to time, or rather, the velocity of the mapped position.

Definition 20 (Infinitesimal Motion) The infinitesimal motion of a framework is a vector of instantaneous velocity $\vec{v} = [v(1), v(2), \dots, v(N)]$ of all joints that is admissible of the tangent space $\partial\mathcal{F}(\mathcal{G}, \delta)$. The infinitesimal motion is called *trivial* if either $v(i) = 0, \forall i \in \mathcal{V}$ or the velocities together induce an trivial motion.

Definition 21 (Infinitesimal Rigidity) A framework is *infinitesimal rigid* if the tangent space of its corresponding framework space does not permit any non-trivial infinitesimal motion.

Theorem 1 (Infinitesimal Rigidity & Generic Rigidity [7]) Given a generic framework (\mathcal{G}, p) , then it is rigid if and only if it is infinitesimally rigid.

Based on the formal definition of the tangent space and the definition of infinitesimal motion, it can be observed that characterizing infinitesimal motion is closely related to linear algebra. First, $[p(i) - p(j), p(j) - p(i)][v(i), v(j)]^\top = 0$ can be expressed in a compact form of *group level*. Specifically, we have $r_{ij}\vec{v}^\top = 0, r_{ij} \in \mathbb{R}^d N$, where the $d(i-1) + 1$ -th to di -th entries of r_{ij} equal to $p(i) - p(j)$ and the $d(j-1) + 1$ -th to dj -th entries equal to $p(j) - p(i)$. Further, if append all row vectors $r_{ij} ((i, j) \in \mathcal{E})$ to form a matrix $M(\mathcal{G}, p)$, then a valid infinitesimal motion belongs to the *null space* of $M(\mathcal{G}, p)$. Subsequently, it is convenient to perform a rank check on the matrix $M(\mathcal{G}, p)$ to determine infinitesimal rigidity. Therefore, we call $M(\mathcal{G}, p)$ the *rigidity matrix* of a framework, and the details of the rank check is provided in the following theorem.

Theorem 2 (Rank Condition on Infinitesimal Rigidity[7]) Given a generic framework (\mathcal{G}, p) with at least 2 joints, then it is infinitesimal rigid if and only if $\text{rank}(M(\mathcal{G}, p)) = s(N, d)$, where

$$s(N, d) = \begin{cases} Nd - \binom{d+1}{2}, & N \geq d + 2; \\ \binom{N}{2}, & N \leq d + 1 \end{cases} \quad (5.2)$$

Moreover, if we restrict the dimension to be $d = 2$, the above rank theorem degenerates to part of the famous Laman's theorem. In planer 2-dimension scenarios, global rigidity can also be clearly characterized [12].

Still, the rank check can provide concrete help in determining generic rigidity in 3-dimension. Specifically, we are interested to have the rank of the rigidity matrix equal to $3N - 6$ in localization in 3-D. Regarding RL, where the translation is not important, we still have $3(N - 1) - 3 = 3N - 6$ rank constraints on the reduced rigidity matrix which is defined as follows

Definition 22 (Reduced Rigidity Matrix) For a framework with N joints, assume the i^* -th joint is the reference joint without loss of generality (i.e., fix $p(i^*) = 0$). The edge set is still \mathcal{E} . Expressing all distance constraints l_{ij} with respect to the i^* -th joint leads to

$$l_{ij} = \begin{cases} (p(i) - p(j))(p(i) - p(j))^\top, & i, j \neq i^* \\ p(j)p(j)^\top, & i = i^* \end{cases} \quad (5.3)$$

where the case $j = i^*$ is neglected since it is equivalent to $i = i^*$. Next, append all l_{ij} into a column vector \vec{l} , the reduced rigidity matrix is defined as

$$M_{re}(\mathcal{G}, \delta, i^*) = \frac{\partial \vec{l}}{\partial \vec{p}}, \quad M_{re}(\mathcal{G}, \delta, i^*) \in \mathbb{R}^{|\mathcal{E}| \times (N-1)} \quad (5.4)$$

where $|\mathcal{E}|$ is the cardinality of \mathcal{E} , and $\vec{p} = [p(1)^\top, p(2)^\top, \dots, p(i^* - 1)^\top, 0, p(i^* + 1)^\top, \dots, p(N)^\top]^\top$ is the concatenated relative position vector.

On checking global rigidity on higher dimensional spaces (i.e., $d \geq 2$), the most recent results use the shared stress matrix of a graph and the Gauss map from differential geometry [47]. Nonetheless, there do not exist any combinatorial conditions for checking global rigidity in \mathbb{R}^3 [47]. Therefore, taking into account that computing (relative) positions of a rigid framework are more paramount than checking rigidity, a more viable solution for 3-D localization is constructing a global rigid framework for sub-graphs. The most recent results is bonding tetrahedrons (i.e., the simplex in \mathbb{R}^3 that satisfy the rank condition of reduced rigidity matrix) to form a *2-star-4-vertex-connected* framework in \mathbb{R}^3 [28].

Focusing on computing the (relative) positions, it is a question that whether knowing the distances suffices to uniquely localize the agents in a network. To this end, the concept of *universal rigidity* has been proposed to characterize the localizability of a network [168]. The definition is given as follows.

Definition 23 ((Generic) Universal Rigidity) *A framework (\mathcal{G}, p) is (generically) universally rigid if and only if it is (generically) globally rigid for any dimension $d \geq 1$.*

Universal rigidity further eliminates the dependence of the dimension for rigidity since global rigidity, being its weaker condition, still implicitly relies on the dimension of the embedded space. At this point, we eventually reach the strongest definition of rigidity and thus end this section on the mathematical foundations of rigidity theory.

5.2. Rigidity Theory for Cooperative Relative Localization

Applying geometric properties to solve localization problems starts from *trilateration* and representative early solutions [8, 37, 168], though benefit from rigidity theory, still heavily relies on trilateration by proposing *trilateration graphs* with pertinent constructions and solutions.

Later improvements regarding localization using rigidity theory put emphasis on noisy distance measurements [104, 163]. Considering that UWB also produces noisy distance measurements, it is desired to elaborate more on this aspect. As already identified in the previous section, tetrahedrons are concrete and reliable frameworks in \mathbb{R}^3 , and it is also treated as a *robust quadrilateral* to bound the localization error [104]. However, the algorithm based on their proposed robust quadrilateral only provides the lowest probability of correctly computing the positions of joints in a graph. A completely different approach focusing on individual distance measurements and outlier inaccuracy has been developed [163] where the concept of *verifiable edge* was proposed to determine whether any outlier exists for given distance measurement with the rigidity of the rest of the graph. Still, checking the verifiability of the edges lacks probability analysis of the measurements themselves and it greatly increases the computational complexity by introducing a bunch of rigidity checks.

Notably, most existing solutions using rigidity [104, 163, 8, 37, 168] assume that part of the agents in the network has access to their global position information while the major task is determining the relative positions to these fixed beacons. At first glance, a slight modification can be made to serve the RL problem by assigning a reference node as what is defined in the reduced rigidity matrix. However, it is still an unanswered question that how many fixed beacons are necessary to uniquely compute the relative positions in existing solutions using rigidity theory.

Therefore, the most recent example on 3-D RL using geometric graph properties abandons the rigidity theory and instead exploits sequential distance measurements between a pair of agents [72], which was first proposed for 3-D relative pose estimation with quaternion formulation and WLS solver [144]. Indeed, simply applying rigidity theory to retrieve accurate pose estimation strongly relies on the accuracy of the distance measurements and the synchronization of the distances obtained at different time instants, both of which are far too strong requirements for UWB-based swarms. Correspondingly, it may be helpful to combine sequential distance measurements and make predictions based on the dynamics to guide the rigidity computation.

5.3. Optimization for Rigidity Theoretic Approaches

In this section, we delve further into the computational perspective of rigidity theory, and we will focus on SDP optimization which is a commonly used formulation of rigidity problems. Besides, SDP also demonstrates good performance for the problem formulation using sequential distance measurements [72]. As a supplement, the relaxation issue of SDP will also be briefly discussed.

5.3.1. Introduction to Semidefinite Programming

The most well-known formulation of SDP is given in the following block.

SDP Primal Formulation

Let $X \in \mathbb{R}_s^{n \times n}$ be the decision variable, and $C, A_i (i = 1, 2, \dots, m) \in \mathbb{R}_s^{n \times n}$ be constant symmetric matrices. Besides, let $b = [b_1, b_2, \dots, b_m]^T \in \mathbb{R}^m$ be a constant vector. The primal SDP is given as the following optimization problem

$$\begin{aligned} X^* &= \arg \min_X C \bullet X \\ \text{s.t. } & A_i \bullet X = b_i, \forall i = 1, 2, \dots, m, \\ & X \succeq 0, \end{aligned} \quad \begin{aligned} (5.5a) \\ (5.5b) \end{aligned}$$

where the bullet dot \bullet is the notation for matrix inner product, i.e., $A \bullet B = \sum_{i=1}^n \sum_{j=1}^n A_{ij} B_{ij} = \text{tr}(A^T B)$. C contains all parameters of the objective function, and $A_i \bullet X = b_i (i = 1, 2, \dots, m)$ form m linear equations. In addition, $X \succeq 0$ is equivalent to $X \in \mathbb{R}_{\text{spds}}^{n \times n}$, which states that each of the n eigenvalues of X is non-negative.

Remark 6 (Matrix Inner Product Formulation of Quadratic Form) Given a column vector $a \in \mathbb{R}^n$ and a matrix $X \in \mathbb{R}^{n \times n}$, we have the following equality

$$a^T X a = (a a^T) \bullet X \quad (5.6)$$

which suggests that a quadratic form can be converted to a matrix inner product. Thus, many optimization problems with quadratic equality constraints can also be posed as SDP.

SDP is an extension of linear programming (LP) [148]. Same as LP, SDP also has its dual formulation, and there are strong connections between the solution of the primal problem and the dual problem. Unlike LP, SDP intrinsically is a special type of convex optimization and thus *duality gap* exists in general. The dual problem sometimes can be solved more efficiently [148], thus we also present the dual formulation.

SDP Dual Formulation

Let $y = [y_1, y_2, \dots, y_m]^T \in \mathbb{R}^m$ be the decision variable, and $C, A_i (i = 1, 2, \dots, m) \in \mathbb{R}_s^{n \times n}$ be constant symmetric matrices. Besides, let $b = [b_1, b_2, \dots, b_m]^T \in \mathbb{R}^m$ be a constant vector. The dual formulation of the primal SDP is given as the following optimization problem.

$$\begin{aligned} X^* &= \arg \max_y b^T y \\ \text{s.t. } & C - \sum_{i=1}^m y_i A_i \succeq 0 \end{aligned} \quad (5.7a)$$

Any feasible value of the primal and the dual problem maintains the following relation in general

$$C \bullet X - b^T y = \epsilon \geq 0 \quad (5.8)$$

where ϵ is the duality gap. (5.8) also refers to as the *weak duality relation*. Moreover, having $\epsilon = 0$ is the *strong duality relation*. Similar to the duality theorem for general convex programming problems [11], if the primal problem or the dual problem is *strictly feasible*, then strong duality holds.

The most efficient algorithm for solving SDP is the interior-point method [148]. For computer-based simulation, SEDUMI [138] is a very popular on-the-shelf solver, and it is also embedded in YALMIP [97]. For real experiments of embedded system testing, extra coding for the optimization solver is necessary.

5.3.2. Semidefinite Programming for Cooperative Relative Localization

After introducing the basics of SDP, we now formulate the *intermediate* CRL problem from a rigidity perspective. Given a group of N agents described using a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with its framework realized in \mathbb{R}^d , assume that a sub-group of agents, called *virtual ground beacons*, have already performed relative pose estimation within themselves. Thereupon, the coming task is performing a state estimation to get the relative pose of the remaining agents with respect to the virtual ground beacons using only distance measurements. For the set of virtual ground beacons denoted as $\mathcal{B} \subset \mathcal{V}$, we can assign a reference position $p(k^*) = 0$ to an arbitrarily chosen agent $k^* \in \mathcal{B}$. Correspondingly, the absolute positions of other beacons $k \in \mathcal{B} \setminus k^*$, in this reference, become the known relative positions.

To diffuse future ambiguities, we denote the known position of agents $k \in \mathcal{B}$ as β_k , $k = 1, 2, \dots, |\mathcal{B}|$, and k will be the index for agents in \mathcal{B} . Regarding the *unknown agents* in $\mathcal{A} = \mathcal{V} \setminus \mathcal{B}$, we use $i(j) = 1, 2, \dots, |\mathcal{A}|$ as the index, and their positions (i.e, the relative position) is denoted as $x_i(x_j)$. Neglecting the edges within the virtual beacons, we further divide the remaining edges into two sets \mathcal{E}_{aa} and \mathcal{E}_{ab} representing the edges within the undecided agents and the edges between virtual beacons and unknown agents, respectively. Based on the above formulation, the distance constraints within the group is given as

$$\|\beta_k - x_j\|^2 = l_{jk}^2, \forall (j, k) \in \mathcal{E}_{ab}, \quad (5.9a)$$

$$\|x_j - x_i\|^2 = l_{ij}^2, \forall (i, j) \in \mathcal{E}_{aa}. \quad (5.9b)$$

For compactness, we build a concatenated matrix $X_p = [x_1, x_2, \dots, |\mathcal{A}|]$ to represent all the unknown relative positions, with which the distance constraints in (5.9) can be rewritten as

$$[\beta_k; -e_j]^\top [I \ X]^\top [I \ X] [\beta_k; -e_j] = l_{jk}^2, \forall (j, k) \in \mathcal{E}_{ab}, \quad (5.10a)$$

$$(e_i - e_j)^\top X^\top X (e_i - e_j) = l_{ij}^2, \forall (i, j) \in \mathcal{E}_{aa}. \quad (5.10b)$$

Therefore, we can formulate the *intermediate* CRL problem into SDP as follows.

SDP Formulation of Intermediate CRL

Computing the unknown relative positions is equivalent to solving the following optimization problem

$$Z^* = \arg \min_X C \bullet Z$$

$$\text{s.t. } (e_i e_i^\top) \bullet Z = 1, \forall i = 1, 2, \dots, d \quad (5.11a)$$

$$([\beta_k; -e_j][\beta_k; -e_j]^\top) \bullet Z = l_{jk}^2, \forall (j, k) \in \mathcal{E}_{ab} \quad (5.11b)$$

$$([0; e_i - e_j][0; e_i - e_j]^\top) \bullet Z = l_{ij}^2, \forall (i, j) \in \mathcal{E}_{aa} \quad (5.11c)$$

$$Z = \begin{bmatrix} I & X \\ X^\top & Y \end{bmatrix} \succeq 0 \quad (5.11d)$$

where $Y = X^\top X \in \mathbb{R}_s^{d \times d}$, and $Z \in \mathbb{R}_s^{(d+|\mathcal{A}|) \times (d+|\mathcal{A}|)}$ are both symmetric matrices. Besides, $C = 0$ is the naive setting since we are looking for feasible solutions. Upon obtaining Z^* , X^* can be trivially retrieved thereafter.

This intermediate CRL formulation is not the whole story, as it assumes part of the agents has already obtained relative position information. Fortunately, this formulation is quite general such that the assumption itself can also be solved using the same SDP formulation, which leads to an *iterative* CRL. Again, it is compatible with peer-to-peer RL using nonlinear filtering techniques introduced in the previous chapter 4. To enlighten future research, we summarize the insights as follows.

Insights on Rigidity Guided CRL using SDP

- **Iterative SDP:**

Pick a candidate agent as the starting agent and perform a rigidity-based CRL for a small universally rigid group (i.e., four agents forming a tetrahedron) locally. Subsequently, treat

this small rigid group as the group of virtual ground beacons and perform estimation for the other agents that are linked to them but relatively more spatially far away from the starting agent. Repeat the above procedure either locally or globally until all interested agents are visited at least once. To draw an analogy, this approach is like the spread of public opinions among people.

- **Guided Filtering with SDP:**

Perform peer-to-peer RL using nonlinear filtering in a parallel manner for different pairs of agents. Correspondingly, many different groups of virtual ground beacons are available, with which a bunch of SDP computations directly follow and have desired decentralized property. An analogous example is burning a thick and big canvas by igniting multiple scattered points simultaneously.

- **Accelerated Convergence by Alternating SDP and Filtering:**

Since peer-to-peer RL takes time to converge and initialization also influences the speed of convergence, it is beneficial to apply SDP and filtering in an *overlapped* manner. Viable strategies are as follows.

1. Apply SDP to provide initialization.
2. At the intermediate point before nonlinear filtering converges, use that information as the initialization of the SDP and perform SDP to get a progressed solution which is then feedback to the nonlinear filtering in its next iteration. The solution obtained by the filter can then be feedback to the SDP in the next iteration of SDP.

- **Event-triggered SDP:**

This idea directly follows the guided filtering with the SDP approach. Since the computation of SDP is more demanding than filtering, it is reasonable to apply SDP in an *event-triggered* manner such that it interferes with the filter at *selected* iterations instead of every step.

5.3.3. Relaxation in Semidefinite Programming

The SDP formulation presented for intermediate CRL in the previous subsection is a relaxed version since the hard equality relation $Y = X^T X$ is relaxed as $Y \succeq X^T X$. In the original formulation without relaxation, there is a rank constraint $rank(Z) = d$ on the variable Z , which makes the resulting formulation *non-convex* and intractable to solve. To fix this problem, an alternating rank minimization algorithm (ARMA) [149] has been recently proposed where the rank condition is reformulated as a sub optimization problem.

An old attempt to handle the relaxation for quadratic constrained quadratic programming (QCQP) problems uses reformulation-linearization-technique (RLT) to retain the substantial feasible region removed by semidefiniteness constraints.

Without relying on multiple distance measurements of the big group, peer-to-peer RL using sequential distance measurements can also be posed as SDP optimization. This approach also uses a relaxation of the original problem to fit the SDP framework, and it applies singular-value-decomposition (SVD) to tighten the relaxed problem.

5.4. Advances in Rigidity Theory

This final section discusses several miscellaneous topics in rigidity theory, which are all recent advances dealing with different aspects in RL for multi-agent systems. The first topic originates from the UWB sensor where rigidity is considered directly for the time measurements. We then focus on some recent solutions where rigidity is combined with filtering techniques. The last part in this section introduces some basics on applying rigidity theory to Barycentric coordinates, which is an alternative to tackle the non-convex optimization of the formulated rigidity problem without relaxation.

5.4.1. Rigidity Theory Meets Ultra-wideband

The mechanism of measuring distances using UWB has been introduced in chapter 2. The transmitted signal has a timestamp recording the time of transmission using the local time of the transmitter. On the receiver's end, the recorded arrival time of the signal obtained by the matched filter is also in the

local time of the receiver. For agent $i \in \mathcal{V}$, denote its local time as t_i , then the first-order relation to the global reference clock t is as

$$t = \alpha_i t_i + \phi_i, \quad (5.12)$$

where $\alpha_i \in \mathbb{R}_+$ and $\phi_i \in \mathbb{R}$. Further, the recorded local transmission time of agent i to agent j is $T_{i,j}^i$, where the superscript shows the local time belongs to which agent and the subscript clarifies the direction of the transmission. The two *directed* measurements can be easily computed as

$$c(\alpha_j T_{(i,j)}^j + \phi_j - \alpha_i T_{(i,j)}^i - \phi_i) = l_{ij}, \quad (5.13a)$$

$$c(\alpha_i T_{(j,i)}^i + \phi_i - \alpha_j T_{(j,i)}^j - \phi_j) = l_{ji}. \quad (5.13b)$$

Assuming the distance measurements are identical (i.e., $l_{ij} = l_{ji}$), combine (5.13a) and (5.13b) further leads to

$$f_{ij}^c = \alpha_i \bar{T}_{ij}^i + \phi_i - \alpha_j \bar{T}_{ij}^j - \phi_j = 0, \quad (5.14)$$

where $\bar{T}_{ij}^i = 0.5(T_{(i,j)}^i + T_{(j,i)}^i)$ and $\bar{T}_{ij}^j = 0.5(T_{(i,j)}^j + T_{(j,i)}^j)$.

Treating (α_i, ϕ_i) as the variable that need to be determined, a similar *clock framework* (\mathcal{G}, φ) can be defined with φ being the set of all clock parameter pairs (α_i, ϕ_i) . The number of The equality in (5.14) equals the cardinality of the edge set $|\mathcal{E}|$. Appending all f_{ij} generates a vector function of the clock parameters, and taking the partial derivative of this vector function with respect to the clock parameters leads to the *infinitesimal clock rigidity matrix*. Subsequently, trivial clock motion and rank check on the clock rigidity matrix can be further characterized [156, 157].

In addition, when working with one of the directed distance measurements as well as the position embedding information $p(i)$, a *biased* equality directly follows as

$$f_{ij}^{c,p} = \|p(i) - p(j)\|^2 - c(\alpha_j T_{(i,j)}^j + \phi_j - \alpha_i T_{(i,j)}^i - \phi_i) = 0. \quad (5.15)$$

Following the same procedure as in constructing clock rigidity matrix, taking the derivative of the vectorized $f_{ij}^{c,p}$ function with respect to both positions and clock parameters results in the *infinitesimal joint clock-position rigidity matrix* [156, 157].

This pioneering work that combines UWB and rigidity theory still suffers from several flaws. First, it does not discuss how to eliminate the bias in the joint clock-position rigidity theory. Besides, their approach for clock parameter estimation and joint estimation both use gradient-descent approaches on a predefined Lyapunov energy function while did not provide details about hyper-parameter tuning and initialization. A rather severe problem is they applied incorrectly the Barbashin's theorem on positive invariant set.

5.4.2. Rigidity Theory Meets Filtering Theory

Apart from the novel insights we presented in subsection 5.3.2, there is already an attempt trying to leverage rigidity theory and filtering theory [3]. This work aims for beacon-free localization of gun projectiles using radio-frequency (RF) communication, and it uses Kalman Filter to partly eliminate the ambiguities brought by simply relying on rigidity theory which fails to handle reflection. Indeed, reflection is a tricky problem even in cases where global rigidity is guaranteed, and the resulting position estimation using rigidity produces multiple hypotheses, which are further tested using KF. Therefore, this also inspires future research to design multi-hypothesis KF with rigidity theory.

Other irrelevant papers where (E)KF is also presented simply use the filter to verify the results in the simulation [126, 61]. Nonetheless, they still provide some interesting results. For instance, inspired by the observability analysis using Lie derivatives, a higher-order dynamic analysis is performed for the bearing rigidity matrix, which sheds light on a possible direction for analyzing higher-order dynamics also for the distance rigidity matrix.

5.4.3. Rigidity in Barycentric Coordinates

As identified in subsection 5.3.3, the SDP formulation does not exactly correspond to the true problem due the relaxation (c.f., (5.11d)). However, without the relaxation, the optimization problem is not convex and thus becomes difficult to solve. Barycentric coordinate system, being a useful tool in affine geometry, can describe the geometric constraints of a given framework embedded in Euclidean space using *linear* equations [55, 139].

Let s_1, s_2, \dots, s_{d+1} be $n + 1$ *affinely independent* points in a general *affine space* \mathbb{S} of dimension n . Then for any point s^* and an arbitrary point o , there exists a unique set of scalar parameters $\lambda_1, \lambda_2, \dots, \lambda_{n+1}$ independent of o such that the following equations hold:

$$\overrightarrow{os^*} = \sum_{i=1}^{n+1} \lambda_i \overrightarrow{os_i}, \quad \sum_{i=1}^{n+1} \lambda_i = 1, \quad (5.16)$$

and the scalars $\lambda_i (i = 1, 2, \dots, n + 1)$, being not identically zero, are the *barycentric coordinates* of s^* with respect to s_i . In real life where the affine space degenerates to the 3-D Euclidean space, we need 4 points which can form a tetrahedron to serve as the *basis points*.

Another prominent characteristic of the barycentric coordinate system is its invariance concerning the isometry of a Euclidean space (e.g., translation, rotation, and reflection). Consequently, it is not necessary to know the exact true position of the basis points for computing the barycentric coordinates. An effective approach is to first construct a congruent framework based on the distance measurements, then compute the barycentric coordinates based on the Euclidean position of the vertices of the congruent framework [55].

It should be emphasized that localization using barycentric coordinates requires at least 4 nodes to be the *anchors*. By the same token, it is necessary to have at least 4 neighbors that are communicating with the agent that needs to be localized, which is a rather stringent requirement in dynamic networks. Imagine a scenario where a new agent is joining a group and it needs to be localized (i.e., the other agents in the group need to determine the relative pose of this newcomer), this new agent may fly within the communication range of only 1 or 2 agents, which renders applying barycentric coordinates infeasible.

6

Conclusion

We conclude this review by summarizing important remarks on different solutions we analyzed, formulating the research questions as well as the research objective, and finally providing future research directions.

6.1. Concluding Remarks of Different Techniques

Thus far, we covered three types of nonlinear filtering techniques (i.e., the KF, the PF, and the MHE). Due to the nonlinear nature of the MAV dynamics and the UWB measurement model, the EKF is the baseline filter that can solve the state estimation problem in localization. However, though enjoying simple applicability, the EKF fails to capture higher nonlinearities and cannot handle general non-Gaussian noise. Moreover, there does not exist rigorous proof to guarantee the convergence of the EKF, and it is indeed sensitive to initialization. Other variations of the Bayesian filter provide prominent advantages compared to the KF, typical examples are PF which approximates posterior distribution, and the MHE which exploits optimization. Notwithstanding, improved estimation performance of the PF and the MHE comes with a price of increased computational complexity.

Nonlinear filtering is an effective solution for peer-to-peer localization but benefiting from the big group to achieve a cooperative and consistent solution requires extra tools. For filtering-based solutions, covariance intersection has demonstrated its potential to fuse direct and indirect measurements with unknown correlations. An alternative solution is to exploit graph rigidity theory and its SDP formulation for position estimation. The computation of covariance intersection can be distributed but its performance greatly depends on that of the underlying filter. The rigidity theoretical approach provides position estimation with group information, but it requires solving an online optimization whose computation is hard to be distributed. Besides, it needs prior information and even fails to take effect when some of the agents only have a sparse local connection.

6.2. Research Questions & Research Objectives

This section formulates the research questions, research objective, and sub-goals of the research objective, motivated by the analysis provided in the literature review.

6.2.1. Research Questions

The main research question is *how to design a well-suited CRL algorithm that can consistently estimate 3-D relative pose (i.e., relative position in local coordinate) for swarm MAVs in infrastructure-free and GPS denied environments using only onboard IMU and inter-agent distance measurements from UWB*, where the meaning of *well-suited* is three-fold: 1) loosely coupled decentralized 2) computational efficient for online implementation 3) centimeter-level localization accuracy. According to different requirements, this main research question can be divided into several sub-questions which are listed as follows.

1. **(Reconstructability Problem)** How to achieve peer-to-peer 3-D relative localization using only inter-agent distance measurements and onboard IMU?

2. (**Consistency Problem**) How to benefit from the cooperation of the group to achieve consistent estimation (i.e., the estimated variables of different agents match) as well as more accurate estimation (i.e., centimeter-level accuracy) given the loosely coupled communication constraints?
3. (**Decentralization Problem**) How to decentralize the computation of the estimation algorithm?
4. (**Computational Problem**) How to reduce the computational complexity such that it can be implemented online while satisfying centimeter-level accuracy?

These four sub-questions are closely related. The baseline would be designing an effective estimation algorithm that can reconstruct the 3-D relative pose state variables using only IMU and UWB distance measurements for a pair of MAV agents. Then for a swarm of large group size, it is desired to fuse direct and indirect measurements. Further, virtual constraints provided by physical truth can serve as additional measurements to enhance estimation performance and ensure consistency. When cooperation of the group is utilized to provide consistent and more accurate localization, we need to investigate the computation requirements of this integrated algorithm. This integrated cooperative algorithm is a good design if it mathematically does not require a centralized computation, otherwise, it needs to be decentralized. It is possible that the cooperative design is not decentralizable, and if this is the case, we need to redesign the cooperative algorithm. The last question is to evaluate the algorithm's computational complexity for each MAV agent and try to reduce the computational cost if possible.

6.2.2. Research Objective

The main research objective is to achieve consistent and accurate (i.e., centimeter-level accuracy) 3-D relative pose estimation for swarm MAVs in infrastructure-free and GPS-denied environments using only IMU and UWB distance measurements using designing the next generation CRL algorithm that has decentralized online computational capability and suits loosely coupled communication constraints.

The first sub-goal is to derive and design a CRL algorithm that fits our problem framework by leveraging stochastic systems theory, graph rigidity theory, and optimization techniques. The algorithm should have strong mathematical foundations in terms of stochastic observability check, convergence proof, error bound derivation, and computational complexity analysis. After successfully designing the CRL algorithm, the second sub-goal is to implement the algorithm in computer programs and to perform simulations thereafter. The final sub-goal is applying the algorithm to real MAVs to test the effectiveness of the algorithm in real experiments.

Achieving the first sub-goal and the second sub-goal is interleaved since the development of the algorithm has different phases and it is beneficial to construct modular simulations between phases and check intermediate performance at the end of each phase. At the same time, the algorithm should be modified based on the result of the simulations. Programming for real embedded controllers of MAVs and subsequent online flight tests is only permitted after the simulations produce convincing results.

6.3. Inspiring Future Research Instruction

The future requirements of RL for swarm MAVs as four levels. Firstly, it should be more accurate. Then, it should take less time to produce accurate enough results, or rather, the algorithm should converge faster. Further, it should be robust not only to noise but also to network configuration, which means it should be able to handle different kinds of noise and dynamic network topology. To fulfill these requirements, several insights have been proposed in previous chapters. We here present a final uniform instruction for future research as follows.

1. Leverage PF and EKF with MEE or CEE.
2. Apply covariance intersection on the peer-to-peer filter in step 1 to fuse direct and indirect estimation but not measurements.
3. Interleave the filtering-based method developed in 2 and the rigidity theoretic method and apply rigidity theory in an event-triggered manner.
4. Provide proof of convergence, derived error bound, and computational complexity analysis.

The above instruction may fail to produce good results in the beginning or even may not be feasible. Changes will be made when necessary and more literature research will also be carried out in parallel.

References

- [1] Angelo Alessandri and Mauro Gaggero. “Fast moving horizon state estimation for discrete-time systems using single and multi iteration descent methods”. In: *IEEE Transactions on Automatic Control* 62.9 (2017), pp. 4499–4511.
- [2] Angelo Alessandri et al. “Moving-horizon state estimation for nonlinear systems using neural networks”. In: *IEEE Transactions on Neural Networks* 22.5 (2011), pp. 768–780.
- [3] Bethany L Allik et al. “Kalman filter aiding MDS for projectile localization”. In: *AIAA Scitech 2019 Forum*. 2019, p. 1159.
- [4] Brian DO Anderson et al. “Rigid graph control architectures for autonomous formations”. In: *IEEE Control Systems Magazine* 28.6 (2008), pp. 48–63.
- [5] Ienkaran Arasaratnam and Simon Haykin. “Cubature kalman filters”. In: *IEEE Transactions on automatic control* 54.6 (2009), pp. 1254–1269.
- [6] Filippo Arrichiello et al. “Observability metric for the relative localization of AUVs based on range and depth measurements: theory and experiments”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2011, pp. 3166–3171.
- [7] Leonard Asimow and Ben Roth. “The rigidity of graphs”. In: *Transactions of the American Mathematical Society* 245 (1978), pp. 279–289.
- [8] James Aspnes et al. “A theory of network localization”. In: *IEEE Transactions on Mobile Computing* 5.12 (2006), pp. 1663–1678.
- [9] Alexander Bahr, Matthew R Walter, and John J Leonard. “Consistent cooperative localization”. In: *2009 IEEE International Conference on Robotics and Automation*. IEEE. 2009, pp. 3415–3422.
- [10] Giorgio Battistelli. “Distributed moving-horizon estimation with arrival-cost consensus”. In: *IEEE Transactions on Automatic Control* 64.8 (2018), pp. 3316–3323.
- [11] Dimitri Bertsekas. *Convex optimization theory*. Vol. 1. Athena Scientific, 2009.
- [12] Béla Bollobás and Bela Bollobas. *Modern graph theory*. Vol. 184. Springer Science & Business Media, 1998.
- [13] Richard S Bucy and Peter D Joseph. *Filtering for stochastic processes with applications to guidance*. Vol. 326. American Mathematical Soc., 2005.
- [14] Giuseppe C Calafiore, Luca Carlone, and Mingzhu Wei. “Distributed optimization techniques for range localization in networked systems”. In: *49th IEEE Conference on Decision and Control (CDC)*. IEEE. 2010, pp. 2221–2226.
- [15] Yanjun Cao et al. “Distributed TDMA for mobile UWB network localization”. In: *IEEE Internet of Things Journal* 8.17 (2021), pp. 13449–13464.
- [16] Zhiqiang Cao et al. “Relative Localization of Mobile Robots with Multiple Ultra-WideBand Ranging Measurements”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 5857–5863.
- [17] Luis C Carrillo-Arce et al. “Decentralized multi-robot cooperative localization using covariance intersection”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 1412–1417.
- [18] Guofei Chai et al. “Consensus-based cooperative source localization of multi-agent systems with sampled range measurements”. In: *Unmanned Systems* 2.03 (2014), pp. 231–241.
- [19] Anusna Chakraborty, Kevin M Brink, and Rajnikant Sharma. “Cooperative relative localization using range measurements without a priori information”. In: *IEEE Access* 8 (2020), pp. 205669–205684.

- [20] Tsang-Kai Chang, Kenny Chen, and Ankur Mehta. "Resilient and consistent multirobot cooperative localization with covariance intersection". In: *IEEE Transactions on Robotics* (2021).
- [21] Badong Chen et al. "Minimum error entropy Kalman filter". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51.9 (2019), pp. 5819–5829.
- [22] Yufeng Chen et al. "Multi-agent based surveillance". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2006, pp. 2810–2815.
- [23] Jan Chudoba et al. "Localization and stabilization of micro aerial vehicles based on visual features tracking". In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2014, pp. 611–616.
- [24] Leonardo Colombo et al. "On the observability of relative positions in left-invariant multi-agent control systems and its application to formation control". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 7333–7338.
- [25] Mario Coppola et al. "On-board communication-based relative localization for collision avoidance in micro air vehicle teams". In: *Autonomous robots* 42.8 (2018), pp. 1787–1805.
- [26] Alejandro Cornejo and Radhika Nagpal. "Distributed range-based relative localization of robot swarms". In: *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 91–107.
- [27] Charles Champagne Cossette et al. "Relative position estimation between two uwb devices with imus". In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 4313–4320.
- [28] Mihai Cucuringu. "Graph realization and low-rank matrix completion". PhD thesis. Princeton University, 2012.
- [29] Lujuan Dang et al. "Cubature Kalman Filter Under Minimum Error Entropy With Fiducial Points for INS/GPS Integration". In: *IEEE/CAA Journal of Automatica Sinica* 9.3 (2021), pp. 450–465.
- [30] Daniela De Palma, Giovanni Indiveri, and Gianfranco Parlangeli. "Multi-vehicle relative localization based on single range measurements". In: *IFAC-PapersOnLine* 48.5 (2015), pp. 17–22.
- [31] Decawave Ltd. *V. DW1000 User Manual, "2.11.(decawave, 2017)," 2017.*
- [32] Xiwang Dong et al. "Time-varying formation tracking for second-order multi-agent systems subjected to switching topologies with application to quadrotor formation flying". In: *IEEE Transactions on Industrial Electronics* 64.6 (2016), pp. 5014–5024.
- [33] Nabil M Drawil, Haitham M Amar, and Otman A Basir. "GPS localization accuracy classification: A context-based approach". In: *IEEE Transactions on Intelligent Transportation Systems* 14.1 (2012), pp. 262–273.
- [34] Sebastian Dudzik. "Application of the Motion Capture System to Estimate the Accuracy of a Wheeled Mobile Robot Localization". In: *Energies* 13.23 (2020), p. 6437.
- [35] Bardienus P Duisterhof et al. "Sniffy bug: A fully autonomous swarm of gas-seeking nano quadcopters in cluttered environments". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 9099–9106.
- [36] Jos Elfring, Elena Torta, and René van de Molengraft. "Particle filters: A hands-on tutorial". In: *Sensors* 21.2 (2021), p. 438.
- [37] Tolga Eren et al. "Rigidity, computation, and randomization in network localization". In: *IEEE INFOCOM 2004*. Vol. 4. IEEE. 2004, pp. 2673–2684.
- [38] Xu Fang et al. "Graph optimization approach to range-based localization". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51.11 (2020), pp. 6830–6841.
- [39] Marcello Farina, Giancarlo Ferrari-Trecate, and Riccardo Scattolini. "Distributed moving horizon estimation for linear constrained systems". In: *IEEE Transactions on Automatic Control* 55.11 (2010), pp. 2462–2475.
- [40] Marcello Farina, Giancarlo Ferrari-Trecate, and Riccardo Scattolini. "Distributed moving horizon estimation for nonlinear constrained systems". In: *International Journal of Robust and Nonlinear Control* 22.2 (2012), pp. 123–143.
- [41] Marcello Farina, Giancarlo Ferrari-Trecate, and Riccardo Scattolini. "Distributed moving horizon estimation for sensor networks". In: *IFAC Proceedings Volumes* 42.20 (2009), pp. 126–131.

- [42] Federal Communication Commission. *First report and order 02-48*. 2002.
- [43] Hans Joachim Ferreau et al. "High-speed moving horizon estimation based on automatic code generation". In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE. 2012, pp. 687–692.
- [44] Sinan Gezici and H Vincent Poor. "Position estimation via ultra-wide-band signals". In: *Proceedings of the IEEE 97.2* (2009), pp. 386–403.
- [45] Sinan Gezici et al. "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks". In: *IEEE signal processing magazine 22.4* (2005), pp. 70–84.
- [46] Javier González et al. "Mobile robot localization based on ultra-wide-band ranging: A particle filter approach". In: *Robotics and autonomous systems 57.5* (2009), pp. 496–507.
- [47] Steven J Gortler, Alexander D Healy, and Dylan P Thurston. "Characterizing generic global rigidity". In: *American Journal of Mathematics 132.4* (2010), pp. 897–939.
- [48] Samet Güler, Mohamed Abdelkader, and Jeff S Shamma. "Peer-to-peer relative localization of aerial robots with ultrawideband sensors". In: *IEEE Transactions on Control Systems Technology 29.5* (2020), pp. 1981–1996.
- [49] Kexin Guo, Duo Han, and Lihua Xie. "Range-based cooperative localization with single landmark". In: *2017 13th IEEE International Conference on Control & Automation (ICCA)*. IEEE. 2017, pp. 588–593.
- [50] Kexin Guo, Xiuxian Li, and Lihua Xie. "Ultra-wideband and odometry-based cooperative relative localization with application to multi-UAV formation control". In: *IEEE transactions on cybernetics 50.6* (2019), pp. 2590–2603.
- [51] Kexin Guo et al. "Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in GPS denied environments". In: *International Journal of Micro Air Vehicles 9.3* (2017), pp. 169–186.
- [52] Fredrik Gustafsson. "Particle filter theory and practice with positioning applications". In: *IEEE Aerospace and Electronic Systems Magazine 25.7* (2010), pp. 53–82.
- [53] Ismail Guvenc, Chia-Chin Chong, and Fujio Watanabe. "NLOS identification and mitigation for UWB localization systems". In: *2007 IEEE Wireless Communications and Networking Conference*. IEEE. 2007, pp. 1571–1576.
- [54] Aleksandar Haber and Michel Verhaegen. "Moving horizon estimation for large-scale interconnected systems". In: *IEEE Transactions on Automatic Control 58.11* (2013), pp. 2834–2847.
- [55] Tingrui Han et al. "A barycentric coordinate based approach to three-dimensional distributed localization for wireless sensor networks". In: *2017 13th IEEE International Conference on Control & Automation (ICCA)*. IEEE. 2017, pp. 600–605.
- [56] Yongqiang Han et al. "A novel cooperative localization method based on IMU and UWB". In: *Sensors 20.2* (2020), p. 467.
- [57] Zhimin Han et al. "Integrated relative localization and leader–follower formation control". In: *IEEE Transactions on Automatic Control 64.1* (2018), pp. 20–34.
- [58] Shinsuke Hara and Ramjee Prasad. "Overview of multicarrier CDMA". In: *IEEE communications Magazine 35.12* (1997), pp. 126–133.
- [59] Mohammad Heidari and Kaveh Pahlavan. "Identification of the absence of direct path in toa-based indoor localization systems". In: *International Journal of Wireless Information Networks 15.3* (2008), pp. 117–127.
- [60] Bas van der Heijden et al. "Iterative bias estimation for an ultra-wideband localization system". In: *IFAC-PapersOnLine 53.2* (2020), pp. 1391–1396.
- [61] Larkin Heintzman and Ryan K Williams. "Nonlinear observability of unicycle multi-robot teams subject to nonuniform environmental disturbances". In: *Autonomous Robots 44.7* (2020), pp. 1149–1166.
- [62] Robert Hermann and Arthur Krener. "Nonlinear controllability and observability". In: *IEEE Transactions on automatic control 22.5* (1977), pp. 728–740.

- [63] Jeroen D Hol, Thomas B Schon, and Fredrik Gustafsson. "On resampling algorithms for particle filters". In: *2006 IEEE nonlinear statistical signal processing workshop*. IEEE. 2006, pp. 79–82.
- [64] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme. *Cooperative relative localization for mobile robot teams: An ego-centric approach*. Tech. rep. UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES DEPT OF COMPUTER SCIENCE, 2003.
- [65] Guoquan P Huang, Anastasios I Mourikis, and Stergios I Roumeliotis. "A quadratic-complexity observability-constrained unscented Kalman filter for SLAM". In: *IEEE Transactions on Robotics* 29.5 (2013), pp. 1226–1243.
- [66] Guoquan P Huang et al. "Observability-based consistent EKF estimators for multi-robot cooperative localization". In: *Autonomous Robots* 30.1 (2011), pp. 99–122.
- [67] Alexander T Ihler et al. "Nonparametric belief propagation for self-localization of sensor networks". In: *IEEE Journal on Selected Areas in Communications* 23.4 (2005), pp. 809–819.
- [68] Fernando H Iost Filho et al. "Drones: innovative technology for use in precision pest management". In: *Journal of economic entomology* 113.1 (2020), pp. 1–25.
- [69] Alberto Isidori. *Nonlinear control systems: an introduction*. Springer.
- [70] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.
- [71] James S Jennings, Greg Whelan, and William F Evans. "Cooperative search and rescue with a team of mobile robots". In: *1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97*. IEEE. 1997, pp. 193–200.
- [72] Bomin Jiang, Brian DO Anderson, and Hatem Hmam. "3-D relative localization of mobile systems using distance-only measurements via semidefinite optimization". In: *IEEE Transactions on Aerospace and Electronic Systems* 56.3 (2019), pp. 1903–1916.
- [73] Yung-Hoon Jo et al. "Accuracy enhancement for UWB indoor positioning using ray tracing". In: *Proceedings of IEEE/ION PLANS 2006*. 2006, pp. 565–568.
- [74] Simon J Julier and Jeffrey K Uhlmann. "A non-divergent estimation algorithm in the presence of unknown correlations". In: *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*. Vol. 4. IEEE. 1997, pp. 2369–2373.
- [75] Rokas Jurevičius, Virginijus Marcinkevičius, and Justinas Šeibokas. "Robust GNSS-denied localization for UAV using particle filter and visual odometry". In: *Machine Vision and Applications* 30.7 (2019), pp. 1181–1190.
- [76] Thomas Kailath. *Linear systems*. Vol. 156. Prentice-Hall Englewood Cliffs, NJ, 1980.
- [77] Rudolph Emil Kalman. "A new approach to linear filtering and prediction problems". In: (1960).
- [78] Sung-Mo Kang, Myoung-Chul Park, and Hyo-Sung Ahn. "Distance-based cycle-free persistent formation: Global convergence and experimental test with a group of quadcopters". In: *IEEE Transactions on Industrial Electronics* 64.1 (2016), pp. 380–389.
- [79] Akimitsu Kanzaki et al. "Dynamic TDMA slot assignment in ad hoc networks". In: *17th International Conference on Advanced Information Networking and Applications, 2003. AINA 2003*. IEEE. 2003, pp. 330–335.
- [80] Robin Kellermann, Tobias Biehle, and Liliann Fischer. "Drones for parcel and passenger transportation: A literature review". In: *Transportation Research Interdisciplinary Perspectives* 4 (2020), p. 100088.
- [81] Solmaz S Kia, Stephen Rounds, and Sonia Martinez. "Cooperative localization for mobile agents: A recursive decentralized algorithm based on Kalman-filter decoupling". In: *IEEE Control Systems Magazine* 36.2 (2016), pp. 86–101.
- [82] Manon Kok, Jeroen D Hol, and Thomas B Schön. "Indoor positioning using ultrawideband and inertial measurements". In: *IEEE Transactions on Vehicular Technology* 64.4 (2015), pp. 1293–1303.
- [83] Vijay Kumar and Nathan Michael. "Opportunities and challenges with autonomous micro aerial vehicles". In: *The International Journal of Robotics Research* 31.11 (2012), pp. 1279–1291.

- [84] Ryo Kurazume, Shigemi Nagata, and Shigeo Hirose. "Cooperative positioning with multiple robots". In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE. 1994, pp. 1250–1257.
- [85] Anton Ledergerber and Raffaello D'Andrea. "Ultra-wideband range measurement model with Gaussian processes". In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2017, pp. 1929–1934.
- [86] Anton Ledergerber and Raffaello D'Andrea. "Calibrating away inaccuracies in ultra wideband range measurements: A maximum likelihood approach". In: *IEEE Access* 6 (2018), pp. 78719–78730.
- [87] Jin-Shyan Lee, Yu-Wei Su, and Chung-Chou Shen. "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi". In: *IECON 2007-33rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2007, pp. 46–51.
- [88] Joon-Yong Lee and Robert A Scholtz. "Ranging in a dense multipath environment using an UWB radio link". In: *IEEE journal on selected areas in communications* 20.9 (2002), pp. 1677–1683.
- [89] Keith YK Leung, Timothy D Barfoot, and Hugh HT Liu. "Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach". In: *IEEE Transactions on Robotics* 26.1 (2009), pp. 62–77.
- [90] Hao Li and Fawzi Nashashibi. "Cooperative multi-vehicle localization using split covariance intersection filter". In: *IEEE Intelligent transportation systems magazine* 5.2 (2013), pp. 33–44.
- [91] Jiaxin Li et al. "Accurate 3d localization for mav swarms by uwb and imu fusion". In: *2018 IEEE 14th International Conference on Control and Automation (ICCA)*. IEEE. 2018, pp. 100–105.
- [92] Liang Li and Ming Yang. "Joint localization based on split covariance intersection on the Lie group". In: *IEEE Transactions on Robotics* 37.5 (2021), pp. 1508–1524.
- [93] Ming Li et al. "Relative localization in multi-robot systems based on dead reckoning and uwb ranging". In: *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*. IEEE. 2020, pp. 1–7.
- [94] Shushuai Li et al. "An autonomous swarm of micro flying robots with range-based relative localization". In: *arXiv preprint arXiv:2003.05853* (2020).
- [95] Ran Liu et al. "Cooperative relative positioning of mobile users by fusing IMU inertial and UWB ranging information". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 5623–5629.
- [96] Yuanpeng Liu et al. "Distributed 3D relative localization of UAVs". In: *IEEE Transactions on Vehicular Technology* 69.10 (2020), pp. 11756–11770.
- [97] J. Löfberg. "YALMIP : A Toolbox for Modeling and Optimization in MATLAB". In: *CACSD*. Taipei, Taiwan, 2004. URL: <http://users.isy.liu.se/johanl/yalmip>.
- [98] Rodrigo López-Negrete, Sachin C Patwardhan, and Lorenz T Biegler. "Constrained particle filter approach to approximate the arrival cost in moving horizon estimation". In: *Journal of Process Control* 21.6 (2011), pp. 909–919.
- [99] Nicola Macoir et al. "MAC protocol for supporting multiple roaming users in mult-cell UWB localization networks". In: *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE. 2018, pp. 588–599.
- [100] Agostino Martinelli and Roland Siegwart. "Observability analysis for mobile robot localization". In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2005, pp. 1471–1476.
- [101] Luca Martino, Víctor Elvira, and Francisco Louzada. "Effective sample size for importance sampling based on discrepancy measures". In: *Signal Processing* 131 (2017), pp. 386–401.
- [102] Mohamed W Mehrez, George KI Mann, and Raymond G Gosine. "Nonlinear moving horizon state estimation for multi-robot relative localization". In: *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE. 2014, pp. 1–5.

- [103] Francisco Molina Martel et al. "Unique 4-DOF relative pose estimation with six distances for UWB/V-SLAM-based devices". In: *Sensors* 19.20 (2019), p. 4366.
- [104] David Moore et al. "Robust distributed network localization with noisy range measurements". In: *Proceedings of the 2nd international conference on Embedded networked sensor systems*. 2004, pp. 50–61.
- [105] Bruno Morabito et al. "Simple and efficient moving horizon estimation based on the fast gradient method". In: *IFAC-PapersOnLine* 48.23 (2015), pp. 428–433.
- [106] Mark W Mueller, Michael Hamer, and Raffaello D'Andrea. "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadcopter state estimation". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 1730–1736.
- [107] Hyung G Myung, Junsung Lim, and David J Goodman. "Single carrier FDMA for uplink wireless transmission". In: *IEEE vehicular technology magazine* 1.3 (2006), pp. 30–38.
- [108] Esha D Nerurkar, Stergios I Roumeliotis, and Agostino Martinelli. "Distributed maximum a posteriori estimation for multi-robot cooperative localization". In: *2009 IEEE International Conference on Robotics and Automation*. IEEE. 2009, pp. 1402–1409.
- [109] Thien Hoang Nguyen and Lihua Xie. "Relative Transformation Estimation Based on Fusion of Odometry and UWB Ranging Data". In: *arXiv preprint arXiv:2202.00279* (2022).
- [110] Thien Minh Nguyen et al. "An ultra-wideband-based multi-UAV localization system in GPS-denied environments". In: *2016 International Micro Air Vehicles Conference*. Vol. 6. 2016, pp. 1–15.
- [111] Thien-Minh Nguyen et al. "Distance-based cooperative relative localization for leader-following control of MAVs". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3641–3648.
- [112] Thien-Minh Nguyen et al. "Robust target-relative localization with ultra-wideband ranging and communication". In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 2312–2319.
- [113] Sven Pfeiffer, Christophe De Wagter, and Guido CHE De Croon. "A Computationally Efficient Moving Horizon Estimator for Ultra-Wideband Localization on Small Quadrotors". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6725–6732.
- [114] Cheryl C Qu and Juergen Hahn. "Computation of arrival cost for moving horizon estimation via unscented Kalman filtering". In: *Journal of Process Control* 19.2 (2009), pp. 358–363.
- [115] Jorge Pena Queraltá et al. "VIO-UWB-based collaborative localization and dense scene reconstruction within heterogeneous multi-robot systems". In: *arXiv preprint arXiv:2011.00830* (2020).
- [116] Ioannis Rekleitis, Gregory Dudek, and Evangelos Milios. "Multi-robot collaboration for robust exploration". In: *Annals of Mathematics and Artificial Intelligence* 31.1 (2001), pp. 7–40.
- [117] James F Roberts et al. "3-D relative positioning sensor for indoor flying robots". In: *Autonomous Robots* 33.1 (2012), pp. 5–20.
- [118] Stergios I Roumeliotis. *Robust Mobile Robot Localization: From single-robot uncertainties to multi-robot interdependencies*. University of Southern California, 2000.
- [119] Stergios I Roumeliotis and George A Bekey. "Distributed multirobot localization". In: *IEEE transactions on robotics and automation* 18.5 (2002), pp. 781–795.
- [120] Lang Ruan et al. "Cooperative Relative Localization for UAV Swarm in GNSS-Denied Environment: A Coalition Formation Game Approach". In: *IEEE Internet of Things Journal* (2021).
- [121] Antonio Ramón Jiménez Ruiz and Fernando Seco Granja. "Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis". In: *IEEE Transactions on instrumentation and Measurement* 66.8 (2017), pp. 2106–2117.
- [122] Zafer Sahinoglu, Sinan Gezici, and Ismail Guvenc. "Ultra-wideband positioning systems". In: *Cambridge, New York* (2008).

- [123] Zafer Sahinoglu and Ismail Guvenc. "Multiuser interference mitigation in noncoherent UWB ranging via nonlinear filtering". In: *EURASIP Journal on Wireless Communications and Networking* 2006 (2006), pp. 1–10.
- [124] Frederico Santos, Luís Almeida, and Luís Seabra Lopes. "Self-configuration of an adaptive TDMA wireless communication protocol for teams of mobile robots". In: *2008 IEEE International Conference on Emerging Technologies and Factory Automation*. IEEE. 2008, pp. 1197–1204.
- [125] James B Saxe. "Embeddability of weighted graphs in k-space is strongly NP-hard". In: *Proc. of 17th Allerton Conference in Communications, Control and Computing, Monticello, IL*. 1979, pp. 480–489.
- [126] Fabrizio Schiano and Roberto Tron. "The dynamic bearing observability matrix nonlinear observability and estimation for multi-agent systems". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 3669–3676.
- [127] René Schneider and Christos Georgakis. "How to not make the extended Kalman filter fail". In: *Industrial & Engineering Chemistry Research* 52.9 (2013), pp. 3354–3362.
- [128] Jens Schroeder et al. "NLOS detection algorithms for ultra-wideband localization". In: *2007 4th Workshop on Positioning, Navigation and Communication*. IEEE. 2007, pp. 159–166.
- [129] Kibeom Seong, Mehdi Mohseni, and John M Cioffi. "Optimal resource allocation for OFDMA downlink systems". In: *2006 IEEE international symposium on information theory*. IEEE. 2006, pp. 1394–1398.
- [130] Mohammed Shalaby et al. "Relative position estimation in multi-agent systems using attitude-coupled range measurements". In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 4955–4961.
- [131] Zhiguo Shi et al. "An acoustic-based surveillance system for amateur drones detection and localization". In: *IEEE transactions on vehicular technology* 69.3 (2020), pp. 2731–2739.
- [132] Jessica Sidman and Audrey St John. "The rigidity of frameworks: Theory and applications". In: *Notices of the AMS* 64.9 (2017).
- [133] Andrea Simonetto, Daniele Balzaretto, and Tamás Keviczky. "A distributed moving horizon estimator for mobile robot localization problems". In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 8902–8907.
- [134] Anthony Man-Cho So and Yinyu Ye. "Theory of semidefinite programming for sensor network localization". In: *Mathematical Programming* 109.2 (2007), pp. 367–384.
- [135] Enrique Stevens-Navarro, Vijayanth Vivekanandan, and Vincent WS Wong. "Dual and mixture Monte Carlo localization algorithms for mobile wireless sensor networks". In: *2007 IEEE Wireless Communications and Networking Conference*. IEEE. 2007, pp. 4024–4028.
- [136] David Stier et al. "A Test Platform for UWB-Based Localization of Dynamic Multi-Agent Systems". In: *IEEE Robotics and Automation Letters* (2022).
- [137] Jared Strader et al. "Cooperative relative localization for moving UAVs with single link range measurements". In: *Proceedings of IEEE/ION PLANS 2016*. 2016, pp. 336–343.
- [138] Jos F Sturm. "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones". In: *Optimization methods and software* 11.1-4 (1999), pp. 625–653.
- [139] Pedro PV Tecchio et al. "N-dimensional distributed network localization with noisy range measurements and arbitrary anchor placement". In: *2019 American Control Conference (ACC)*. IEEE. 2019, pp. 1195–1201.
- [140] Matthew J Tenny and James B Rawlings. "Efficient moving horizon estimation and nonlinear model predictive control". In: *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*. Vol. 6. IEEE. 2002, pp. 4475–4480.
- [141] E Tijs et al. "Hear-and-avoid for micro air vehicles". In: *Proceedings of the International Micro Air Vehicle Conference and Competitions (IMAV), Braunschweig, Germany*. Vol. 69. 2010.
- [142] Nikolas Trawny and Stergios I Roumeliotis. "On the global optimum of planar, range-based robot-to-robot relative pose estimation". In: *2010 IEEE International Conference on Robotics and Automation*. IEEE. 2010, pp. 3200–3206.

- [143] Nikolas Trawny, Stergios I Roulmeliotis, and Georgios B Giannakis. "Cooperative multi-robot localization under communication constraints". In: *2009 IEEE International Conference on Robotics and Automation*. IEEE. 2009, pp. 4394–4400.
- [144] Nikolas Trawny et al. "3D relative pose estimation from distance-only measurements". In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2007, pp. 1071–1078.
- [145] Sridhar Ungarala. "Computing arrival cost parameters in moving horizon estimation using sampling based filters". In: *Journal of Process Control* 19.9 (2009), pp. 1576–1588.
- [146] Rudolph Van Der Merwe et al. "The unscented particle filter". In: *Advances in neural information processing systems* 13 (2000).
- [147] Thang Van Nguyen et al. "Least square cooperative localization". In: *IEEE Transactions on Vehicular Technology* 64.4 (2015), pp. 1318–1330.
- [148] Lieven Vandenberghe and Stephen Boyd. "Semidefinite programming". In: *SIAM review* 38.1 (1996), pp. 49–95.
- [149] Changhuang Wan et al. "Sensor network localization via alternating rank minimization algorithms". In: *IEEE Transactions on Control of Network Systems* 7.2 (2019), pp. 1040–1051.
- [150] Eric A Wan and Rudolph Van Der Merwe. "The unscented Kalman filter for nonlinear estimation". In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. IEEE. 2000, pp. 153–158.
- [151] Thumeera R Wanasinghe, George KI Mann, and Raymond G Gosine. "Decentralized cooperative localization for heterogeneous multi-robot system using split covariance intersection filter". In: *2014 Canadian Conference on Computer and Robot Vision*. IEEE. 2014, pp. 167–174.
- [152] Sen Wang et al. "An optimization based moving horizon estimation with application to localization of autonomous underwater vehicles". In: *Robotics and Autonomous Systems* 62.10 (2014), pp. 1581–1596.
- [153] Sen Wang et al. "Single beacon based multi-robot cooperative localization using Moving Horizon Estimation". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 1625–1630.
- [154] Yan Wang and Xin Li. "The IMU/UWB fusion positioning algorithm based on a particle filter". In: *ISPRS International Journal of Geo-Information* 6.8 (2017), p. 235.
- [155] Yongcai Wang et al. "Formation tracking in sparse airborne networks". In: *IEEE Journal on Selected Areas in Communications* 36.9 (2018), pp. 2000–2014.
- [156] Ruixin Wen, Eric Schoof, and Airlie Chapman. "Clock rigidity and joint position-clock estimation in ultra-wideband sensor networks". In: *arXiv preprint arXiv:2106.02199* (2021).
- [157] Ruixin Wen, Eric Schoof, and Airlie Chapman. "Estimation for Ultra-Wideband Sensor Networks: a Rigidity Perspective". In: *2021 Australian & New Zealand Control Conference (ANZCC)*. IEEE. 2021, pp. 12–17.
- [158] Adam Wiktor and Stephen Rock. "Compartmentalized Covariance Intersection: A Novel Filter Architecture for Distributed Localization". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 5589–5595.
- [159] Henk Wymeersch et al. "A machine learning approach to ranging error mitigation for UWB localization". In: *IEEE transactions on communications* 60.6 (2012), pp. 1719–1728.
- [160] Lin Xiao, Stephen Boyd, and Sanjay Lall. "A scheme for robust distributed sensor fusion based on average consensus". In: *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005*. IEEE. 2005, pp. 63–70.
- [161] Baojian Yang et al. "A New Robust Centered Error Entropy Cubature Kalman Filter". In: *2021 IEEE 7th International Conference on Control Science and Systems Engineering (ICCSSE)*. IEEE. 2021, pp. 119–124.
- [162] Liuqing Yang and Georgios B Giannakis. "Ultra-wideband communications: an idea whose time has come". In: *IEEE signal processing magazine* 21.6 (2004), pp. 26–54.

- [163] Zheng Yang et al. "Detecting outlier measurements based on graph rigidity for wireless sensor network localization". In: *IEEE Transactions on Vehicular Technology* 62.1 (2012), pp. 374–383.
- [164] Jing Zhang and Jinfeng Liu. "Distributed moving horizon state estimation for nonlinear systems with bounded uncertainties". In: *Journal of Process Control* 23.9 (2013), pp. 1281–1295.
- [165] Lijun Zhang et al. "Cubature Kalman filtering for relative spacecraft attitude and position estimation". In: *Acta Astronautica* 105.1 (2014), pp. 254–264.
- [166] Wenda Zhao et al. "Learning-based bias correction for ultra-wideband localization of resource-constrained mobile robots". In: *arXiv preprint arXiv:2003.09371* (2020).
- [167] Jianan Zhu. *UWB-Based Infrastructure-Free Cooperative Localization for First-Responders*. University of California, Irvine, 2020.
- [168] Zhisu Zhu, Anthony Man-Cho So, and Yinyu Ye. "Universal rigidity: Towards accurate and efficient localization of wireless networks". In: *2010 Proceedings IEEE INFOCOM*. IEEE. 2010, pp. 1–9.