DISTRIBUTED COORDINATION AND ESTIMATION OF MULTI-AGENT SYSTEMS

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Science and Engineering

2020

Ishak Hilton Pujantoro Tnunay
Department of Electrical and Electronic Engineering

Contents

List of Figures

Abbreviations

CRB	Cramér-Rao Bound
CKF	Curbature Kalman Filter
DRKF	Distributed Riemannian Kalman Filter
DUKF	Distributed Unscented Kalman Filter
EKF	Extended Kalman Filter
FIM	Fisher Information Matrix
ICRB	Intrinsic Cramér-Rao Bound
MAS	Multi-agent System
MSE	Mean-squared Error
PDF	Probability Density Function
RSN	Robotic Sensor Network
UKF	Unscented Kalman Filter
WSN	Wireless Sensor Network

The University of Manchester

Ishak Hilton Pujantoro Tnunay
Doctor of Philosophy
Distributed Coordination and Estimation of Multi-agent Systems
August 5, 2020

This thesis addresses problems arising in the coordination and estimation of a connected multi-agent system. The distributed coordination of agents consists of cooperative control and optimisation. Cooperative control aims at guiding a team of agents moving in a formation, maintaining a required pattern or velocity. An optimisation problem in a group of agents aims at driving them towards specific locations according to a performance index. Furthermore, the distributed estimation problem when states in a dynamical process are unknown is also considered; the agents are then assigned to estimate the unknown states cooperatively.

A coordination problem investigated in this thesis is the coverage control problem in robot sensor networks whose objective is to find the optimal locations of the sensors leading to the best measurement. In this work, the objective function of locational optimisation is solved by interchanging the positions of the neighbouring agents. The formulated Lagrangian function augments the objective function and the consensus constraint to accommodate this mechanism. Accordingly, a coverage controller with cooperative constraint is proposed via the gradient-descent protocol. Furthermore, a controller is designed to drive the robots to the optimal locations in a finite time regardless of their initial positions. However, since an environment is not generally convex due to the presence of obstacles, a coverage controller with potential-field based obstacle avoidance is also considered to drive the robot to the optimal location safely.

Motivated by the unavailability of information distribution in an environment before sensor deployment, a class of distributed nonlinear filter is designed to estimate the states of a dynamical process using the shared information among the agents. The proposed algorithm extends the distributed unscented Kalman filter to accommodate any communication topology. It utilises not only the measurement from an agent's sensor for estimating the process but also the shared information. The proposed filter is then employed to estimate the information distribution in the optimal coverage problem with unknown information distribution. Different from the existing field-estimation algorithms, the proposed filter also optimises the estimator gains in every iteration to avoid instability of the system caused by failure to choose the appropriate parameters.

Thereafter, the question regarding how good is a distributed recursive Bayesian-based estimator when the information belongs to more general manifolds other than Euclidean, i.e., Riemannian manifolds, are also addressed. In this thesis, the formulated intrinsic Cramér-Rao bounds (ICRBs) demonstrates that the non-zero curvature term of the manifold also affects the performance of the Bayesian estimator. Lastly, assuming that the probability density functions (PDF) of the noises are Gaussian, a distributed nonlinear Kalman filter for Riemannian systems is also proposed. The simulations verify that the mean-squared error of the designed filter will not be lower than its ICRBs.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's Policy on Presentation of Theses.

Publications

1. Published Journal Papers:

(a) H. Tnunay, Z. Li, and Z. Ding, "Distributed nonlinear Kalman filter with communication protocol," Information Sciences, vol. 513, pp. 270-288, Mar 2020.

(b) C. Wang, H. Tnunay, Z. Zuo, B. Lennox, and Z. Ding, "Fixed-Time Formation Control of Multirobot Systems: Design and Experiments," IEEE Transactions on Industrial Electronics, vol. 66, no. 8, 2019.

(c) O. Onuoha, H. Tnunay, Z. Li, and Z. Ding, "Affine Formation Algorithms and Implementation Based on Triple-Integrator Dynamics," Unmanned Systems, vol. 7, no. 1, 2019.

(d) Z. Li, H. Tnunay, S. Zhao, W. Meng, S. Xie, and Z. Ding, "Bearing-Only Formation Control with Pre-Specified Convergence Time," IEEE Transactions on Cybernetics, to be published.

2. Published Conference Papers:

(a) H. Tnunay, Z. Li, C. Wang, and Z. Ding, "Distributed collision-free coverage control of mobile robots with consensus-based approach," in 2017 13th IEEE International Conference on Control Automation (ICCA), Jul 2017, pp. 678-683.

(b) O. Onuoha, H. Tnunay, and Z. Ding, "Affine Formation Maneuver Control of Multi-Agent Systems with Triple-Integrator Dynamics," in 2019 American Control Conference (ACC), vol. 2019-July. IEEE, Jul 2019, pp. 5334-5339.

(c) O. Onuoha, H. Tnunay, and Z. Ding, "Optimal Affine Formation Control of Linear Multi-agent System," in 2019 IEEE 15th International Conference on Control and Automation (ICCA), vol. 2019-July. IEEE, Jul 2019, pp. 851-856.

(d) J. Zhang, H. Tnunay, C. Wang, X. Lyu, and Z. Ding, "Distributed Coverage Optimization and Control with Applications to Precision Agriculture," in Chinese Control Conference, CCC, vol. 2018-July, 2018.

Acknowledgements

This thesis indeed concludes my research at the Control Systems Centre of The University of Manchester. The completion of this work is certainly due to tremendous support of people around me.

First of all, I would like to thank Professor Zhengtao Ding for his continuous guidance during my research. It has been an honour to have such a rare opportunity to have insights and suggestions from him to develop ideas in tackling challenging problems in my research. I will be always indebted throughout my life for his valuable supervision.

I would also thank Indonesia Endowment Funds of Education (LPDP) for the financial support during my research at The University of Manchester. Their help is gratefully acknowledged.

My sincere appreciations go to Mr Okechi Onuoha, Dr Zhenghong Li and Dr Chunyan Wang, for their constructive comments, suggestions and other technical supports related to my field. The discussion and sharing with them helped me find paths leading to the solutions of my research problems. I am also in a great privilege to work with my colleagues in the Control Systems group; I thank them for the enjoyable research environment we have shared.

I would thank my parents for the never-ending prayer, support and encouragement during my research. Afterall, my special appreciation goes to my fiancée, Titi Iswari, for the valuable sharing, constant encouragement and continuous patience throughout these few years.

In memory of my father

Chapter 1

Introduction

1.1 Motivation

Aggregate motion of a flock of birds, school of fish and coordination of ants have inspired the development of the distributed multi-agent systems (MASs) [1]. Mainly in the area of robotics, wireless sensor networks, and distributed machine learning, the development of MASs has received attention in recent decades as it offers beneficial mechanism in the implementation side. Mimicking the phenomenons in nature, agents in a MAS do not necessarily receive and utilise information from the whole network; instead, they communicate with nearby agents in the coordination process, namely, distributed cooperative control. Due to limited resources in implementations, this mechanism may alleviate the burden of a centralised computer by allowing each agent to evaluate a decision according to information shared with neighbouring agents. By utilising shared resources, the distributed scheme might significantly improve operational effectiveness, enhances scalability and reduces computational costs [2].

In this thesis, distributed coordination of a team of agents could into cooperative control and optimisation. Cooperative control aims at guiding a team of agents moving in a formation maintaining certain patern or velocity. An optimisation problem in a group of agents aims at driving them towards certain locations according to the specified performance index. In many cases, since some parameters of the systems might be unknown, the agents can also be employed to approximate the unknown values cooperatively. This scenario could be referred

to as distributed estimation of multi-agent systems.

In a networked system, distributed coordination has widely been investigated in recent decades. Simple cooperative control using consensus algorithm has received considerable attention, such as [3, 4, 5, 6, 7]. The simple consensus protocol is also applicable to the optimisation problem in multi-agent systems whenever a consensus of the optimising parameters is required. Such an optimisation problem is then referred to as the distributed optimisation. Previous activities related to distributed optimisation has been investigated in [8, 9, 10, 11]. Furthermore, if some optimising parameters are unknown before, a distributed estimation algorithm needs to be involved. Several studies related to this area have been conducted by [12, 13, 14, 15, 16, 17, 18] demonstrating the performance of distributed estimation algorithm in connected multi-agent systems. Another distributed estimation algorithm combined with application to robotic sensor network for approximating an informatin distribution in an unknown area can be found in [19, 20, 21].

1.2 Literature Review

1.2.1 Consensus Protocol for Coordination

In coordination of a MAS, consensus protocol is a fundamental technique utilised in many applications including formation control [22], distributed optimisation [23] and estimation [24]. The agents in a consensus protocol exchanges information with neighbouring agents and subsequently updates its state based on the received information such that the information of all agents converges to a consensus value. From the optimisation perspective, the consensus protocol aims to minimise the disagreement function of information throughout the network [5, 25, 26]. A crucial means to investigating the consensus algorithm is graph theory. Algebraically, the graph topology of a network corresponds to adjacency, degree and Laplacian matrices which could accordingly be employed to design many cooperative controllers [27, 28, 6].

The communication topology, represented by a graph, plays an essential role in the performance of the consensus algorithm. In the continuous-time consensus

with fixed time topology, the consensus value could be achieved if and only if the communication graph is connected for undirected graph, or strongly connected for directed graph [5, 6]. The Laplacian matrix whose graph is strongly connected has only a zero eigenvalue. Moreover, the average consensus is reached asymptotically if the directed graph is balanced. In discrete-time consensus protocol, the communication topology is often represented via Perron stochastic matrix instead of Laplacian matrix [29, 13]. Unlike the continuous-time consensus, a strongly connected graph is a necessary condition, not sufficient, to guarantee the convergence to a consensus value. Also, a balanced graph is just a necessary condition to guarantee the convergence to average consensus. In some cases, Perron matrix could be assembled utilising the Laplacian matrix with some additional parameters [5]. Consensus protocol with fixed communication topology has been established in [30, 31]. However, in implementation, switching topology might happen due to, for example, limited communication range. The problem of consensus with switching topologies has been studied in [32, 25, 33]. In term of communication topology, these results about switching graphs indicated that as long as the graph is strongly connected, the consensus value of the network can still be achieved.

Timeliness has become a crucial requirement in many applications, such as in a post-disaster evacuation and nuclear decommissioning, to prevent worsening situations [34]. In control theory, timeliness is related to the settling time or the convergence of an autonomous system to the origin from a set of initial values. The study reported in [35] has initiated the finite-time stability analysis in the control system by showing that the convergence time depends on the initial states. By utilising the proposed finite-time strategy, a finite-time consensus of a team of agents with single-integrator dynamics, double-integrator dynamics and with nonholonomic dynamics were presented in [36, 37, 38]. The finite-time based controller has also found application in the pose synchronisation of spacecrafts whose poses are represented by dual-quaternion as reported in [39]. However, since those results depend on the initial values, if the agents are initially separated with large distance, the system requires more time to converge to a formation. To overcome this problem, results in [40] proposed a

finite-time consensus controller that guarantees the convergence within a specified boundary of settling time regardless of the initial positions. Subsequently, work in [41] has extended the result in [40] to consensus of multi-agent systems with double-integrator. Furthermore, this approach has also been applied to design a finite-time consensus controller with the presence of time delays in a networked system in [42].

1.2.2 Coverage Control in Robotic Sensor Network

Wireless sensor network (WSN), defined as a group of wirelessly-connected sensors deployed in an environment to monitor its physical condition and accordingly to provide the required information, has received attention in recent decades. The physical conditions might include, but not limited to, temperature, pollution levels, humidity, wind, sound, and nuclear radiation. In a remote area, a WSN could be equipped with actuators enabling them to move according to particular command dynamically. WSN with dynamic movement capability could be referred to as a robotic sensor network (RSN). There are two possible architectures of the network: centralised and decentralised. However, to improve the modularity and scalability of the network as well as to comply with the limited resources, including power supply, processor, and memory, a decentralised communication scheme is desirable in implementation.

From the optimisation perspective, the main task of a WSN is to maximise the coverage of the deployed sensors, which leads to the best measurement data of the corresponding environment. To address this problem, the locational optimisation, originated from the field of operation research, has been proposed to find the best locations of agents given an interest function [43, 44]. However, this optimisation problem is solved only for specific information distribution. This optimisation requires manual recalculation whenever the information distribution changes. To overcome this problem, centroidal Voronoi Tessellation has become a recognised tool to solve this locational problem [45, 46]. Several strategies to solve this problem by iteratively minimises the objective function of the locational optimisation has been reported in [46] and [47].

The distributed coverage control implementation has attracted attention from

the robotics community. By adopting the locational optimisation problem [48], a simple proportional controller was initially developed in [46]. However, this algorithm requires several assumptions to be satisfied: the sensing range of all agents needs to be unlimited, isotropic and homogenous, the environment where the robots are deployed needs to be convex, and the communication graph has been known a priori. A coverage strategy to overcome the problem with limited sensing range has been addressed in [49] by assuming that the sensing range is still isotropic but only within a certain range. To relax the homogeneity assumption about the sensor, the protocol in [50] has enabled implementation of coverage control to a non-convex environment using a group of sensors with heterogeneous sensing range. Studies in [51] and [52] have investigated coverage control with non-convex areas by a group of sensors with limited and anisotropic sensing range. Another result related to the non-convex environment coverage control has been conducted in [53] by considering obstacles inside the environment. Regarding the communication topology, result in [54] has included a dynamically-routing communication algorithm while optimising the coverage control problem. However, among the existing mechanisms in literature, coverage algorithm from the control perspective that guarantees the timely convergence in a finite time has not been investigated.

Furthermore, the aforementioned results assumed that the information distribution of an area had been known by all agents, which is restrictive in implementation. This situation would require an additional field estimation algorithm to enable deployment of mobile sensors in an unknown area. The most common strategy to find the information distribution is by assuming that the information distribution is a linear combination of kernel functions whose weights are adjusted during observations [55, 56, 21]. A decentralised and adaptive coverage control to estimate the appropriate weights has been investigated in [55]. The convergence rate of the algorithm was then improved via ladybug exploration scenario in [57] and consensus learning [58]. In [56], a distributed interpolation scheme has also been investigated to estimate the weight of the kernel functions recursively. It should be noted that noises in the system and measurement have not been considered explicitly. Different from the coverage control perspective,

a gradient-based source-finding mechanism has been proposed in [59] where the noise in the measurement process has been considered.

Most related to our work, studies about coverage problems with field estimation of unknown information distribution can be found in [60, 61, 21]. These field estimation algorithms have not considered the noises in the measurement and communication network. Moreover, these algorithms require estimator gains need tuning before execution, but the boundary of the gains has not been given. Failure to choose suitable gains prior to the execution might lead to an unstable system.

1.2.3 Distributed Unscented Kalman Filter

The estimation problem of a process is mainly characterised by the presence of, at least, an unknown parameter in the dynamics. This problem naturally emerges whenever a system provides no direct measurement of the required parameter, or the process and sensor noises disrupt the information. An estimator is required to provide an estimate value closest to the true value. Since measurements and processes, in reality, are generally noisy, many areas in signal processing require solving the estimation problem, such as the attitude estimation of an aircraft [62], visual odometry [63], localisation and mapping [64], etc.

In the area of estimation problem, Kalman filter has been a notable technique utilised to deal with sensor filtering, data fusion, state estimation and system identification. It includes uncertainties to the model of a system, and then iteratively analyzes the statistics from sensing information in order to achieve a minimum mean-squared error (MMSE) between the obtained information and the desired value. Generally, there are two main classifications of Kalman filter: linear Kalman filter and nonlinear Kalman filter. Linear Kalman filter is designed to estimate a system with linear dynamics. Example of the previous development in this linear Kalman filter can be seen in [65]. As an extension of the linear Kalman filter, Extended Kalman Filter (EKF) was designed to solve the nonlinear estimation problem by linearizing the nonlinear dynamics and then treated the linearised system as the linear Kalman filter. Another technique to treat the nonlinear estimation is Unscented Kalman Filter (UKF) by utilising unscented transformation in the prediction update. There have been numerous studies

investigating this nonlinear estimation, the example of the developments related to these nonlinear filtering methods can be found in [66, 67, 68, 69, 16]. In addition to ordinary estimation problem, Kalman filter has also found application in the reinforcement learning algorithm to estimate the optimal policy, value function and Q-function via temporal difference method [70, 71].

The field of Kalman filters in a connected sensor network has become an interesting area to investigate. It incorporates the concept of distributed control systems and the Kalman filtering method. In the case of centralised Kalman filters, for both the linear and nonlinear systems, intensive investigations have been carried out, for example, by [65, 67, 68, 69, 16, 72] and [73]. While in the case of networked linear filter, the study reported by [74] added updating weight on the measurement dynamics to fuse information from a number of sensors, while [75] includes the low, high- and band-pass consensus filters. The stability of a networked filter, namely, the Kalman-Consensus filter, has been analysed in [12], while various scenarios of diffusive Kalman filtering for the linear system have been designed in [15]. Generalising the concept of distributed Kalman filtering, the networked nonlinear Kalman filtering has also attracted numerous developments in distributed Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Cubature Kalman Filter (CKF) [14, 76, 77, 78, 13].

In application, the distributed version of Kalman filter-based estimator has also attracted attentions due to the high scalability, robustness to failure, and flexibility [13, 79]. The distributed protocol is able to reduce the computation burden of a central processor while maintaining the performance of the global result. In practice, communication medium has noises which can affect the quality of shared information among agents. Although there have been some results on the distributed nonlinear Kalman filter algorithms in literature, such as in [75, 13, 80], the communication noise has not been considered and analysed. Both in the linear and nonlinear cases, the previous research on distributed Kalman filter have considered the implementation of consensus protocol to minimise the disagreement of the estimate results among the sensors. However, the optimal gain of the consensus term has not been analysed in order to guarantee the performance. If one fails to pick the consensus gain, the result may oscilate and

become unstable. Thus, the optimal Kalman and consensus gains need to be designed such that the optimality and boundedness can be guaranteed.

1.2.4 Distributed Bayesian Estimation on Manifolds

Literature indicates that information can evolve not only in Euclidean space but also in other manifolds, such as Riemannian manifolds. Estimator for, specifically, Euclidean information can be designed by minimising the norm of the mean-squared Euclidean distance between the true and the estimate values. This scheme applies to the estimation problem of information in a locally-flat manifold and having a sufficiently-large signal-to-noise ratio(SNR). However, since the direct implementation of this type of estimator to Riemannian systems is not generally applicable, the estimator needs modification to accommodate the non-zero curvature of the manifold.

An important measure of the performance of an estimator is the Cramér-Rao bound (CRB). The bound describes the theoretical limitation of an estimator, which can accordingly be utilised as a means to assess the actual performance quality. In the Euclidean estimation problem, the bound corresponds to matrix inequality where the covariance matrix of an estimator is at least as high as the inverse of the Fisher information matrix (FIM) [81, 82]. This bound indeed applies to Riemannian manifold if the manifold is locally Euclidean and the information has a sufficiently-large signal-to-noise ratio (SNR). Otherwise, the lower bound of the covariance matrix will comprise the inverse of the FIM together with the curvature terms [83, 84, 85].

The development of recursive CRBs in Euclidean information space have been conducted in the recent years, as reported in [81, 86, 87, 82, 88, 89]. The recursive version of the posterior CRB in [81] was elegantly derived by firstly constructing a Fisher information matrix from the first to the last posterior probability distributions; followed by extracting the inverse of the right-lower block of the large Fisher information matrix. Following the identical procedure, one improved CRBs to explain the performance of filtering, prediction, and smoothing steps of an estimator [86]. In [87], conditional CRB of recursive Bayesian estimation was developed. However, another conditional CRB proposed in [82] removed the

auxiliary FIM of the recursive update for the conditional bound proposed in [87], and provided alternative approximation which was shown to be more compact and computationally efficient. By assuming the distribution of the noises to being Gaussian, posterior CRB for dual Kalman estimation was derived in [88]. In [89], the bound for a class of distributed recursive Bayesian estimator, i.e., unscented particle filter, with application to multiple source tracking using acoustic sensor network was derived.

Different from the Euclidean estimation problem, which has been intensively studied, Riemannian based Bayesian estimator still requires more attention. The recursive Bayesian estimators for information in manifolds with Gaussian distribution might extend the Euclidean approach; hence, Kalman filter is the optimal solution of the estimation problem. However, as information is evolving in a manifold, the curved geodesics of the manifold necessitates additional treatment to tackle the estimation problem. Directly implementing Euclidean filtering strategies on curved Riemannian manifold might leads to incorrect estimate value as they do not consider the curvature of the manifold utilising, for example, exponential or logarithm mapping. Accordingly, results in previous estimator designs, as reported in [90] and [91], have utilised the exponential and logarithmic mapping of manifolds. The latter explicitly considers the noises in the process and measurement dynamics and provides more rigorous formulation. Instead of designing filter in generic Riemannian manifold, there have been several developments that consider the Lie group or quaternion property in the filter design and apply it to intrinsic filter on Lie group [92, 93], particle filter on Lie group [94], pose estimation using dual-quaternion [95], attitude estimation in SE(3) with gyroscope bias [96], localisation problem [97], and visual inertial odometry [98, 99]. However, based on those existing results, a distributed implementation of the filtering algorithm for Riemannian system has not been studied.

Although several studies have addressed the Riemannian estimator, those results have not provided the bounds for the distributed recursive estimator, especially the Bayesian recursive estimator. Because the Euclidean CRBs do not have curvature terms, those CRBs are not applicable to a Riemannian estimator as the bounds will inaccurately describe the performance of the estimator. The

ICRBs for general Riemannian manifolds were presented in [83], wherein the curvature terms of the information manifold has been included. Later, ICRBs for Riemannian submanifold and quotient manifold were formulated in [84], showing that the higher-order terms of the curvature in the ICRBs for general Riemannian manifolds vanishes for signal with sufficiently large SNR. Considering the special case of ICRB for Riemannian manifolds, the bound for Lie groups was developed in [85], where the bound was derived in terms of Lie bracket and proved to be identical with the one in Riemannian manifolds. Nevertheless, to the best of our knowledge, ICRB for recursive estimation has not been addressed.

1.3 Contributions

The contributions of the thesis could primarily be classified into distributed coordination problem in a robotic sensor network to solve the coverage control problem and distributed estimation problem.

The coordination problem addressed in this work is mainly focused on guaranteeing a group of sensors to achieve the best locations to measure the information of an environment while maintaining the cohesiveness of the network. Different from the existing results in literature, the first scenario of the coverage controller is developed via distributed optimisation perspective to guarantee the ability to track the optimal points of the coverage problem while maintaining the consensus of the agents' formation. The second scenario considers the finite-time based coverage control protocol to guarantee the convergence of the positions of the robots within a limited amount of time. Furthermore, the obstacle avoidance controller is also appended to the controller of each robot prevent the collision between an agent and the other objects while moving towards the optimal location and maintaining the formation.

In the estimation problem, the distributed nonlinear Kalman filter is firstly developed to accommodate various communication processes. Our algorithm has explicitly considered the communication noise in a communication module which could disrupt the information exchanged within a network. Secondly, as reflected in literature review, there have been considerable works developing the

CRBs for recursive Bayesian estimator for Euclidean systems. This thesis relaxes the Euclidean-space restriction by formulating the ICRBs for Bayesian estimator accomodating the non-zero curvature of Riemannian manifolds. Moreover, after presenting the lower bounds, this thesis also provides an extension of distributed unscented Kalman filter for systems whose states and measurements belong to Riemannian manifolds, namely, distributed Riemannian Kalman filter.

1.4 Thesis Organisation

The remainder of this thesis is constructed of five chapters, and those are organised as follows.

In Chapter 2, several relevant theories required in this study, including the algebraic graph theory, Riemannian manifolds, intrinsic statistic on manifolds, and stability theory, are briefly reviewed.

In Chapter 3, the distributed coverage control of multiple connected sensors is presented. The objective function of the locational optimisation is modified using the notion of distributed optimisation to accommodate the consensus constraint of the parameters. Thereafter, a finite-time coverage controller is proposed to guarantee the convergence to the optimal setting before a specified time. After designing the coverage control algorithms, an obstacle avoidance based controller is proposed to prevent collision between robots and their surrounding objects, including other robots and static obstacles. Please note that some results presented in this chapter have been published in the paper entitled "Distributed Collision-free Coverage Control of Mobile Robots with Consensus-based Approach". The other results will be published in the paper entitled "Distributed Coverage Control with Finite-Time Convergence."

Chapter 4 subsequently addresses the estimation problem which naturally emerges in the coverage control of mobile sensors in an unknown environment. An estimator is required to estimate the information distribution of the area to feed the estimate value to the coverage controller. To accommodate the noise in a communication process among the agents, the distributed nonlinear Kalman filter with communication protocol is developed to extend the unscented transformation

originated in Unscented Kalman Filter (UKF). The proposed estimator is then applied to the coverage control problem in an unknown environment along with the proposed coverage controller. Please note that the results presented in this chapter have been published in the paper entitled "Distributed Nonlinear Kalman Filter with Communication Protocol."

Chapter 5 elaborates our attempt to answer the question about the quality of a distributed Bayesian-based recursive estimator if information belongs to a Riemannian manifold. The ICRBs is formulated as the lowest covariance matrix, and consequently the mean-squared error, an estimator could achieve. A distributed unscented Kalman filter for Riemannian systems is also proposed as an example of a distributed Bayesian-based filter in manifolds. Please note that the results presented in this chapter will be published in the paper entitled "Intrinsic Cramér-Rao Bounds for Distributed Bayesian Estimator."

Finally, Chapter 6 provides the summary of this work and outlines the possible directions of the future works.

Chapter 2

Preliminaries

This chapter presents the preliminaries related to the work in this thesis. It briefly reviews the notions of graph theory, Riemannian manifolds, intrinsic statistics on manifolds, and stability theory of nonlinear systems.

2.1 Algebraic Graph Theory

To model the communication or sensing network among the robots, the notion of directed graph is utilised. Suppose that there are n robots. In graph theory, vertices represent the agents; while edges represent the connection links among the agents.

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a collection of vertices $\mathcal{V}$, $\mathcal{V} = \{1, 2, \ldots, n\}$, connected by a collection of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. If there exists an edge $e_i = (i, j) \in \mathcal{E}$, agent i is able to receive information from agent j. If, for all $(i, j) \in \mathcal{E}$, there exists $(j, i) \in \mathcal{E}$, the graph is called undirected; it implies that this communication link enable those two agents to exchange their information. Vertices $j \in \mathcal{N}_i \subset \mathcal{V}$ are the neighbours of agent i, for all $j \neq i$ if there is communication link between agent i and j, i.e., $(i, j) \in \mathcal{E}$.

To algebraically express the connectivity of a graph, we introduce the adjacency, in-degree, Laplacian and Perron matrices.

The adjacency matrix of a graph $\mathcal{G}$, denoted as $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, is a square

matrix whose elements are given by

$$a_{ij} = \begin{cases} 1 & \forall (j,i) \in \mathcal{E}, \\ 0 & \text{otherwise}, \end{cases} \tag{2.1}$$

for all $i, j \in \mathcal{V}$. In the case of undirected graph, it can clearly be seen that A is a symmetric matrix, i.e., $a_{ij} = a_{ji}$. Correspondingly, summing up the i-th row of A yields a in-degree matrix, i.e., $D = \text{diag}(d_1, d_2, \ldots, d_N) \in \mathbb{R}^{n \times n}$ such that $d_i = \sum_{j=1, j \neq i}^{n} a_{ij}$, $\forall i \in \{1, 2, \ldots, n\}$. The associated Laplacian matrix can be then defined as $\mathcal{L} = [\mathcal{L}_{ij}] \in \mathbb{R}^{n \times n}$, for $\mathcal{L}_{ii} = d_i$ and $\mathcal{L}_{ij} = -a_{ij}$. In a matrix operation, a Laplacian matrix can be obtained from the subtraction of in-degree by adjacency matrix, $\mathcal{L} = D - A$. For a more compact expression, we use $\mathcal{L}$, D, and A to refer to $\mathcal{L}$, D, and A, respectively. Laplacian matrix has eigenvalues whose properties might be obtained using the Gershgorin's disc theorem [6]. They can sequentially be expressed as

$$0 = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n \leqslant 2\Delta \tag{2.2}$$

where 0 is the trivial eigenvalue of $\mathcal{L}$ with corresponding eigenvector $\mathbf{1}_n \in \mathbb{R}^n$ and $\Delta = \max_i d_i = \max_i \sum_{j \in N_i} a_{ij}$. For directed graph $\mathcal{G}$, zero is a simple eigenvalue of $\mathcal{L}$ if and only if $\mathcal{G}$ has spanning tree, i.e., a tree starting from a vertex i to all other nodes with a minimum possible number of edges. For undirected graph, zero is a simple eigenvalue of $\mathcal{L}$ if and only if $\mathcal{G}$ is connected [100, 6]. For a strongly connected directed graph, there is only one zero eigenvalue of $\mathcal{L}$, i.e., $\lambda_1 = 0$. Moreover, according to Courant-Fisher theorem, the smallest and largest non-zero eigenvalues of $\mathcal{L}$ satisfy $\lambda_2 \|z\|^2 \leqslant z^\top \mathcal{L} z$ for $\mathbf{1}_n^\top z = 0$ and $\lambda_n \|z\|^2 \geqslant z^\top \mathcal{L} z$ for $z \in \mathbb{R}^n$, respectively [101].

Before explaining the Perron matrix, we say a matrix H is row, or column, stochastic if the sum of each row, or column, elements equals to one. Perron matrix of the graph denoted by $\Pi = [\pi_{ij}] \in \mathbb{R}^{n \times n}$ is a stochastic nonnegative matrix where its eigenvalues are inside a unit circle. If graph $\mathcal{G}$ is balanced, then Π is a doubly-stochastic matrix. Perron matrix Π is said to be primitive if there exists a positive m-th power of Π for some integer m. In this case, strongly connected graph is a necessary condition for Π to be primitive. In Euclidean consensus, the relationship between Laplacian and Perron matrices could be

expressed as $P = I - \epsilon L$ for some scalar ϵ. If the graph is strongly connected and $\epsilon \in (0, 1/\Delta]$, then the Perron matrix Π is primitive [5, 29].

In the coverage control problem, the notions of Voronoi tessellation and Delaunay graph are required. Let n robots be located in a convex set $\mathcal{Q} \in \mathbb{R}^d$ at a set of distinct positions $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, $p_i \in \mathcal{Q}$. These points are then assigned as Voronoi generators. The Voronoi tessellation of $\mathcal{Q}$ generated by $\mathcal{P}$ is defined as $V(\mathcal{P}) = \bigcup_{i=1}^{n} \mathcal{V}_i(p_i)$, where

$$V_i(p_i) = \{q \in \mathcal{Q} : \|q - p_i\| \leqslant \|q - p_j\|, \forall p_j \in \mathcal{P}, j \neq i\}. \tag{2.3}$$

We use V_i conveniently to refer to $V_i(p_i)$.

To exemplify the property of Voronoi partition, suppose that the Voronoi partition is a two-dimensional Euclidean space. We can see that the boundary of the Voronoi partition is a one-dimensional Euclidean space, or a line. In a general case, we can say that if $\mathcal{Q}$ is a convex set in a d-dimensional Euclidean space, then the boundary of $\mathcal{V}_i$ is a $(d-1)$-dimensional convex set.

To build Voronoi partition in a distributed manner, an agent need to know the position of its neighbouring agents. The connection among the agents can then be constructed as a networked topology called proximity graph – a graph that is constructed based on particular geometric requirements. There are a number of proximity graph, such as Delaunay graph, r-disk graph, r-Delaunay graph and r-limited Delaunay graph. The details related to these graphs can be found in [49]. In the case of Voronoi tessellation systems, a proximity graph $\mathcal{G}_{\text{prox}} = (\mathcal{P}, \mathcal{E}(\mathcal{P}))$ is constructed using the Voronoi generator $\mathcal{P} \in \mathcal{Q}$ and its edges $\mathcal{E}(\mathcal{P})$, where $\mathcal{E} \subseteq \mathcal{Q} \times \mathcal{Q}$.

In the discussion about coverage control, we assume that the agents are connected by Delaunay graph which the definition is given as follows. Let $\mathcal{B}_r(p) = \{q \in \mathbb{R}^d : \|q - p\| \leqslant r\}$ be a closed ball in $\mathbb{R}^d$ at $p \in \mathbb{R}^d$ with radius $r \in \mathbb{R}_+$. We then have Delaunay graph denoted by $\mathcal{G}_D = (\mathcal{P}, \mathcal{E}_D(\mathcal{P}))$ whose edge set is given by

$$\mathcal{E}_D(\mathcal{P}) = \{(p_i, p_j) \in \mathcal{P}^2 \setminus \text{diag}(\mathcal{P}^2) | \mathcal{V}_i \cap \mathcal{V}_j \neq \emptyset\}.$$

Since the Delaunay triangulation is applied to generate Voronoi partition without limited communication range, we always have an connected Delaunay graph of

the network. Thus, there always exists the smallest non-zero eigenvalue of graph Laplacian $\mathcal{L}_D$ satisfying $\lambda_2\|z\|^2 \leqslant z^\top \mathcal{L}_D z$ for $\mathbf{1}_n^\top z = 0$.

2.2 Riemannian Manifolds

In this section, we briefly review of some concepts in Riemannian geometry relevant to the formulation of the ICRBs and the Kalman-based nonlinear filter. Please refer to [102] for more detailed elaboration.

Let a pair $(\mathcal{M}, g)$ denote a set of real smooth manifold $\mathcal{M}$ equipped with a metric g. Also, let the set of all tangent vectors of $\mathcal{M}$ at $p \in \mathcal{M}$ be defined as tangent space at p and denoted by $T_p\mathcal{M}$. If metric g at each point $p \in \mathcal{M}$ is a bilinear, symmetric and positive-definite map $g_p : T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$ and induces an inner product $\langle X_p, Y_p\rangle_p \doteq g_p(X_p, Y_p)$ for $X_p, Y_p \in T_p\mathcal{M}$, then g is a Riemannian metric. Hence, a manifold $\mathcal{M}$ with a Riemannian metric g is Riemannian manifold.

Suppose that all vector fields of differential manifold $\mathcal{M}$ is denoted by $\mathcal{X}(\mathcal{M})$. Then, for smooth real-valued functions on manifold $\mathcal{M}$, $f, g : \mathcal{M} \to \mathbb{R}$ and $X, Y, Z \in \mathcal{X}(\mathcal{M})$, affine connection ∇ of a differential manifold $\mathcal{M}$ is a mapping $\nabla : \mathcal{X}(\mathcal{M}) \times \mathcal{X}(\mathcal{M}) \to \mathcal{X}(\mathcal{M}) : (X, Y) \mapsto \nabla_X Y$ satisfying:

1. $\nabla_{fX+gY} Z = f\nabla_X Z + g\nabla_Y Z$,
2. $\nabla_X(Y + Z) = \nabla_X Y + \nabla_X Z$,
3. $\nabla_X(fY) = f\nabla_X Y + df(X)Z$.

Moreover, if the affine connection satisfies the following properties:

1. preserving the metric: $X\langle Y, Z\rangle = \langle \nabla_X Y, Z\rangle + \langle Y, \nabla_X Z\rangle$, and
2. torsion-free or symmetric, i.e., $\nabla_X Y - \nabla_Y X = [X, Y]$,

where $[X, Y]$ is the Lie bracket of X and Y, then ∇ is the Riemannian connection of manifold $\mathcal{M}$, or widely-known as Levi-Civita connection. In fact, the differential manifold equipped with this Riemannian connection is a Riemannian manifold.

Consider $\mathcal{I} \subseteq \mathbb{R}$ as an open interval subset of $\mathbb{R}$, and a differentiable function $\gamma : \mathcal{I} \to \mathcal{M}$ as a differentiable curve in $\mathcal{M}$. Using the Riemannian metric g along the interval $\mathcal{I}$, the arc length of γ is

$$L_a^b(\gamma) \doteq \int_a^b \|\dot{\gamma}(t)\| dt = \int_a^b \langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle_{\gamma(t)}^{1/2} dt, \tag{2.4}$$

where $\dot{\gamma} \in T_\gamma \mathcal{M}$ is the curve derivative or instantaneous speed vector. If, for two points $p, q \in \mathcal{M}$, there exists a curve γ along $\mathcal{I} = (a, b)$, such that $\gamma(a) = p$ and $\gamma(b) = q$, the distance between p and q is

$$\text{dist}(x, y) \doteq \inf_\gamma L_a^b(\gamma). \tag{2.5}$$

Accordingly, a geodesic is defined as the differentiable curve γ minimising the arc length between those two points. In fact, a geodesic is a generalisation paradigm of straight line in a Euclidean space. In term of Levi-Civita connection, a geodesics can also be defined as a curve γ along the interval $\mathcal{I}$ satisfying

$$\nabla_{\dot{\gamma}} \dot{\gamma} = 0. \tag{2.6}$$

If the interval $\mathcal{I}$ can be relaxed to $\mathbb{R}$, then $\mathcal{M}$ is called a geodesically-complete manifold. In this case, based on the Hopf-Rinow theorem, there exists at least a geodesic joining each pair $(p, q) \in \mathcal{M}$. Moreover, a vector field $X \in \mathcal{X}(\mathcal{M})$ is parallel along the defined curve γ if and only if it satisfies the following condition:

$$\nabla_{\dot{\gamma}} X(t) = 0, \ \forall t \in \mathcal{I}. \tag{2.7}$$

Let the domain of γ be defined as $\mathcal{I} = [a, b]$. Then, there exists a unique isomorphism $P_\gamma : T_{\gamma(a)}\mathcal{M} \to T_{\gamma(t)}\mathcal{M}$, for $t \in \mathcal{I}$, such that a vector field $X(t) \in \mathcal{X}(\mathcal{M})$ is parallel along the curve γ with $P_\gamma(X(a)) = X(t)$. This isomorphism is defined as a parallel transport along γ.

Consider two points in manifold denoted by $p, q \in \mathcal{M}$. Let $\gamma : [0, 1] \to \mathcal{M}$ be a geodesic such that, at $v \in T_p\mathcal{M}$, $\gamma_v(0) = p$, $\gamma_v(1) = q$ and $\dot{\gamma}_v(0) = v$. A Riemannian exponential map at p associating the tangent vector $T_p\mathcal{M}$ to manifold $\mathcal{M}$ can be expressed as

$$\text{Exp}_p : T_p\mathcal{M} \to \mathcal{M}, \ v \mapsto \text{Exp}_p(v) = q. \tag{2.8}$$

Within the maximal domain where the exponential map is a diffeomorphism, the inverse of the mapping exists and is defined by

$$\mathrm{Log}_p : \mathcal{M} \to T_p\mathcal{M},\ q \mapsto \mathrm{Log}_p(q) = v. \tag{2.9}$$

The exponential mapping can geometrically be interpreted as a geodesic curve $\gamma(t)$ starting from point p at $t = 0$ with unit velocity $v/\|v\|$ and length $\|v\|$. In a sufficiently small set, the geodesic is unique. It is worth noticing that these exponential and logarithm mappings work as a local diffeomorphism from a sufficiently small neighbourhood of $\gamma(0)$ in the tangent space $T_{\gamma(0)}\mathcal{M}$ into the neighbourhood of $\gamma(1)$ in the manifold $\mathcal{M}$.

Consider a tangent vector $v \in T_p\mathcal{M}$ at $p \in \mathcal{M}$ and a geodesic $\gamma(t) = \exp_p(vt)$ from $t = 0$ to $t = t_v < \infty$. If, for $t > t_v$, the curve γ is no longer a minimising arc length, then the point $\gamma(t_v)$ is a cut point. A collection of all cut points is called a cut locus of p, denoted by $\mathcal{C}_p \subset \mathcal{M}$. If the exponential mapping is still diffeomorphism, the tangential cut locus can easily obtained taking the inverse of the exponential map, $\tilde{\mathcal{C}}_p = \mathrm{Exp}_p^{-1}(\mathcal{C}_p) = \mathrm{Log}_p(\mathcal{C}_p) \subset T_p\mathcal{M}$. The cut locus becomes the boundary of the maximal domain where the exponential mapping is still a diffeomorphism. The maximal domain in the tangent space is denoted $\tilde{\mathcal{U}}_p \subset T_p\mathcal{M}$; while the corresponding set on manifold is the maximal intrinsic domain and denoted by $\mathcal{U}_p = \mathcal{M} - \mathcal{C}_p \subseteq \mathcal{M}$. Based on this explanation, the exponential and logarithm map at point $p \in \mathcal{M}$ can more specifically be expressed as

$$\tilde{\mathcal{U}}_p \longleftrightarrow \mathcal{U}_p \tag{2.10}$$

$$v = \mathrm{Log}_p(q) \longleftrightarrow q = \mathrm{Exp}_p(v). \tag{2.11}$$

Consider a point $p \in \mathcal{M}$. As a generalisation of the definition of ball in Euclidean space, one defines the geodesic ball with centre point at p and radius r as

$$\mathcal{B}(p, r) \doteq \{q \in \mathcal{M} : \mathrm{dist}(p, q) < r\}. \tag{2.12}$$

If all points within the geodesic ball centered at p with radius r are inside the maximal intrinsic domain, then r is the injectivity radius of the maximal geodesic ball, denoted by $\mathrm{inj}_p\mathcal{M}$. Accordingly, the infimum of the injectivity radius over

$p \in \mathcal{M}$ is denoted by $\text{inj}\mathcal{M}$. If the radius of the ball satisfies $2r\sqrt{\kappa} < \pi$ with κ denoting the upper-bound of the Riemannian curvature of the manifold, then the ball is regular. Furthermore, for a connected and complete Riemannian manifold with non-positive curvature, the geodesic ball covers the manifold due to diffeomorphism to $\mathbb{R}^d$ according to Cartan-Hadamard theorem [103].

Consider vector fields $X, Y, Z \in \mathcal{X}(\mathcal{M})$ on a Riemannian manifold $\mathcal{M}$ equipped with Levi-Civita connection ∇. Then, the Riemannian curvature tensor $\mathcal{R} : \mathcal{X}(\mathcal{M}) \times \mathcal{X}(\mathcal{M}) \times \mathcal{X}(\mathcal{M}) \to \mathcal{X}(\mathcal{M})$ is defined as follows:

$$\mathcal{R}(X,Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X,Y]} Z. \tag{2.13}$$

This notion is related to the sectional curvature of the manifold $\mathcal{M}$. The real-number sectional curvature at point $p \in \mathcal{M}$ could be defined as

$$K(X_p, Y_p) = \frac{\langle \mathcal{R}(X,Y)Z, X \rangle_p}{\|X_p\|^2 \|Y_p\|^2 - \langle X_p, Y_p \rangle}, \tag{2.14}$$

where $X, Y, Z \in \mathcal{X}(\mathcal{M})$ and $X_p, Y_p \in T_p\mathcal{M}$.

2.3 Intrinsic Statistics on Manifolds

In this section, we provide some important notions related to intrinsic statistics on Riemannian manifolds based on [103, 104, 105, 106]. Intrinsic statistics of Riemannian manifold also includes the properties of statistics in Euclidean space.

Related to intrinsic statistics, define a discrete point $x = \{x_i \in \mathcal{M} : i = 1, 2, \ldots, n\}$ as a set of random variable in a manifold $\mathcal{M}$. The variance of the random point x with a fixed point y is given by

$$\sigma_x^2(y) = \mathbb{E}\{\text{dist}(y,x)^2\} = \frac{1}{n}\left(\sum_{i=1}^{n} \text{dist}(y, x_i)^2\right).$$

Accordingly, mean point of x is defined as a point minimising this variance. The set containing these points is given by

$$\mathbf{E}\{x\} = \arg\min_{y \in \mathcal{M}} \left(\mathbb{E}\{\text{dist}(y,x)^2\}\right).$$

By assuming that the mean value $\bar{x} \in \mathbf{E}\{x\}$ is unique, the covariance matrix can be defined as

$$\Sigma_{xx} = \text{cov}_x(\bar{x}) = \mathbb{E}\{\text{Log}_{\bar{x}} x (\text{Log}_{\bar{x}} x)^\top\} = \frac{1}{n}\sum_{i=1}^{n} \text{Log}_{\bar{x}} x_i (\text{Log}_{\bar{x}} x_i)^\top.$$

From the covariance matrix, the variance of x can equivalently be obtained via $\sigma_x^2(\bar{x}) = \text{tr}(\Sigma_x x)$. If the probability density function (PDF) of the data Gaussian, we use notation $\mathcal{N}(\bar{x}, \Sigma_{xx})$.

To discuss the properties of Riemannian center of mass, we provide the following definitions and lemmas from [106].

Definition 2.1 ([106])**.** A convex mean function on manifold $\mathcal{M}$ of order $N \geqslant 2$, $\mu : \mathcal{M}^N \to \mathcal{M}$, is a function that maps N points $x = (x_1, x_2, \ldots, x_N)$ to a mean point $\mu(x_1, x_2, \ldots, x_N)$ such that

1. $\mu(x_1, x_2, \ldots, x_N)$ is continuous in its arguments,
2. $\mu(x_1, x_2, \ldots, x_N)$ lies in the closure of the convex hull of $\{x_i\}_{i=1}^N$, and
3. $\mu(x_1, x_2, \ldots, x_N)$ lies in to a closed strongly convex ball containing $\{x_i\}_{i=1}^N$. In particular, $\mu(x_1, x_2, \ldots, x_N)$ is strictly convex mean function if it belongs to the interior of a closed strongly convex ball containing $\{x_i\}_{i=1}^N$ unless $x_1 = x_2 = \ldots = x_n = \bar{x}$ and $\mu(\bar{x}, \bar{x}, \ldots, \bar{x}) = \bar{x}$.

Definition 2.2 ([106])**.** A (strictly) convex vectorial mean function on manifold $\mathcal{M}$ of order $N \geqslant 2$, $\bar{\mu} : \mathcal{M}^N \to \mathcal{M}^N$, is defined as

$$\bar{\mu}(x) = (\mu_1(x), \ldots, \mu_N(x)) \in \mathcal{M}^N, \text{ for } x = (x_1, x_2, \ldots, x_N), \tag{2.15}$$

whose entry μ_i is a (strictly) convex mean function of order N. The composition of $l \geqslant 2$ (strictly) convex vectorial mean functions is given by $\bar{\mu}^l(x) = (\bar{\mu}^1 \circ \bar{\mu}^2 \circ \ldots \circ \bar{\mu}^l)(x)$.

Definition 2.3 ([106])**.** A primitive vectorial mean function is a convex vectorial mean function $\bar{\mu}$ with the primitivity index n_p such that $\bar{\mu}^{n_p}$ is strictly convex mean function.

From the given definitions, it could be observed that the radius of the minimal ball containing the codomain of a convex vectorial mean function is less than or equal to the radius of minimal ball containing the domain. Moreover, if the mean function is strictly convex, the radius of the minimal ball containing the codomain is less than the radius of the minimal ball containing the domain.

The following lemma states the convergence of a primitive vectorial mean function.

Lemma 2.4 ([106]). *The sequence of primitive vectorial mean functions, $\{\bar{\mu}^k(x)\}_{k=1}^{\infty}$, converges pointwise to a strictly convex vectorial function whose entries are equal i.e., $\bar{\mu}^* = (\mu^*, \ldots, \mu^*)$.*

For a Riemannian manifold $\mathcal{M}$ with upper-bounded sectional curvature by κ, consider N points in $\mathcal{M}$, $\{x_i \in \mathcal{M}\}_{i=1}^{N}$. Accordingly, let us define an optimisation of manifold-valued function $f : \mathcal{M} \to \mathbb{R}$:

$$f(x) = \frac{1}{2}\sum_{i=1}^{N} w_i \mathrm{dist}^2(x, x_i), \tag{2.16}$$

with $\sum_{i=1}^{N} w_i = 1$. The Riemannian center of mass or Fréchet mean of these points, denoted by $x^* \in \mathcal{M}$, can be defined as the minimiser of $f(x)$. The following technical lemmas adapted from [106] and [105] state the convexity of the Fréchet mean minimising the objective function (2.16).

Lemma 2.5. *If $\{x_i\}_{i=1}^{N}$ belong to the closure of the convex ball $\bar{\mathcal{B}}(y, r) \subseteq \mathcal{M}$ with $r < r_\kappa = \frac{1}{2}\min\{\mathrm{inj}\mathcal{M}, \frac{\pi}{\sqrt{\kappa}}\}$, then Fréchet mean, x^*, is unique and lies in $\mathcal{B}(y, r)$.*

Lemma 2.6. *If $\{x_i\}_{i=1}^{N}$ belong to the closure of the convex ball $\bar{\mathcal{B}}(y, r) \subseteq \mathcal{M}$ with $r < r_\kappa = \frac{1}{2}\min\{\mathrm{inj}\mathcal{M}, \frac{\pi}{\sqrt{\kappa}}\}$, then Fréchet mean, x^*, lies in the interior of the convex hull of $\{x_i\}_{i=1}^{N}$.*

According to these lemmas, if a set of points are located inside the convex ball $\bar{\mathcal{B}}(y, r) \subseteq \mathcal{M}$, the Fréchet mean of (2.16) is a strictly convex mean function by definition. This fact also follows from the theorem about Riemannian center of mass in [107].

For example, a 2-sphere with radius one and constant curvature one, i.e., $\kappa = 1$, is illustrated in Fig. 2.1. In this illustration, the 2-sphere has injectivity radius $\mathrm{inj}_p\mathcal{S}^2 = \pi$ and convex-ball radius $r_\kappa = \frac{\pi}{2}$. It follows that the diffeomorphism of the exponential mapping exists for points inside the maximal domain whose boundary is given by the injectivity radius. Furthermore, by using Lemmas 2.5 and 2.6, some points located inside the convex ball with radius $\pi/2$ will always have unique Fréchet mean inside the convex hull of those points.

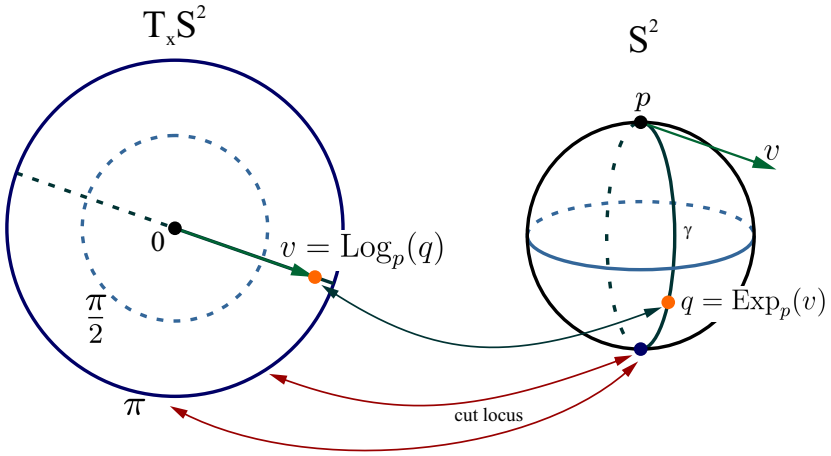


Figure 2.1: Relationship between 2-sphere and its tangent space.

2.4 Stability Theory

In this section, we briefly review some important definitions and properties in stability analysis based on [108] and [109].

Let $x \in U \subset \mathbb{R}^d$, respectively be the state of a system and a domain with $x = 0$ as an interior point. Define a nonlinear system as a continuous mapping $f : U \times [0, +\infty) \to \mathbb{R}^d$ such that

$$\dot{x} = f(x, t). \tag{2.17}$$

Definition 2.7 (Stability of Equilibrium Points)**.** For the system defined in (2.17):

1. the equilibrium point $x = 0$ is Lyapunov stable if, for any given $R \in \mathbb{R}_+$, there exists a constant r to ensure that that $\|x\| < R$, $\forall t > 0$ if $\|x\| < r$;

2. the equilibrium point $x = 0$ is asymptotically stable if it is Lyapunov stable and $\lim_{t\to\infty} x(t) = 0$;

3. the equilibrium point $x = 0$ is globally asymptotically stable if the origin is asymptotically stable for any initial values in $\mathbb{R}^d$;

4. the equilibrium point $x = 0$ is exponentially stable if there exist $\alpha, \beta > 0$ such that

$$\|x(t)\| < \alpha\|x(0)\| \exp(-\beta t), \text{ for } t > 0,$$

in some neighbourhood $U \subset \mathbb{R}^d$ containing the equilibrium point;

5. the equilibrium point $x = 0$ is globally exponential stable if the origin is exponentially stable for any initial values in $\mathbb{R}^d$.

Theorem 2.8 (Lyapunov Stability [108]). *If, for $U \subset \mathbb{R}^d$ containing the equilibrium point $x = 0$, the function $V(x)$ is positive definite and continuously differentiable such that its time derivative along any state trajectory of (2.17) is non-positive, i.e., $\dot{V}(x) \leqslant 0$, then the equilibrium point $x = 0$ of system (2.17) is Lyapunov stable. Moreover, if $\dot{V}(x) < 0$, then the origin is asymptotically stable.*

Theorem 2.9 (LaSalle Invariance Principle [108]). *Let $\Omega \in U$ be a compact set that is positively invariant with respect to (2.17). Let $V : U \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(x) \leqslant 0$ in U. Let D be the set of all points in U where $\dot{V} = 0$. Let M be the largest invariant set in D. Then, every solution starting in D approaches M as $t \to \infty$.*

Chapter 3

Distributed Coverage Control of Mobile Sensors

In this chapter, we address the coverage control problem of robotic sensor networks deployed in an environment. The investigation mainly aims to design algorithms that enable cooperative computation in a distributed fashion while guaranteeing the convergence to the optimal points of the corresponding global objective function. The objective function of the coverage control problem is originated from the locational optimisation problem, that is, a problem of finding the optimal location of sensors in an area. This problem could then be transformed into a local optimisation problem with a consensus constraint.

There are several novel algorithms proposed in this work. The first algorithm directly implements the simple gradient-based control derived from the Lagrangian of the distributed optimisation problem. However, due to some costly computational burdens, we design coverage control protocol that reduces the distributed-optimisation-based controller to a consensus-based tracking controller which could be guaranteed to converge in finite time. Lastly, we also design a coverage controller with collision avoidance mechanism to prevent collisions if the agents face obstacle while moving towards the optimal locations.

The remainder of this chapter is structured as follows. Section 3.1 contains some definitions related to the established locational optimisation problem. Section 3.2 formulates the distributed objective function of locational optimisation which will be solved in our work. Subsequently, in Section 3.3, we present our

proposed algorithms: simple gradient-descent method controller, finite-time coverage controller and coverage control with obstacle avoidance. After presenting the proposed algorithms, verification of the performance is performed via a set of numerical experiments in 3.4. Finally, Section 3.5 summarises our work in this coverage control problem.

3.1 Locational Optimisation

Consider n robots deployed in an environment denoted by a convex set $\mathcal{Q} \subset \mathbb{R}^d$. The set containing the position of all robots is denoted by $\mathcal{P} = \{p_i\}_{i=1}^n \subset \mathcal{Q}$ with p_i is the position of robot i.

Sensing unreliability function $g : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}_+ : (q, p_i) \mapsto g(q, p_i)$, is a function that provides the quantitative information of the sensing performance at point $q \in \mathcal{Q}$ measured by agent i at p_i. In our discussion, we assume the sensing unreliability function to be isotropic, increasing and convex. A function is isotropic if the value is independent on its direction. Hence, the function $g(q, p_i)$ can be re-parametrised to a norm-valued function $f : \mathbb{R} \to \mathbb{R}_+$ such that $g(q, p_i) = f(\|q - p_i\|)$, for $i \in \{1, 2, \ldots, n\}$. In the sense of unreliability function, the isotropic, increasing and convex sensing function means the sensing quality at point q measured by robot i decreases proportionally with the distance $\|q - p_i\|$, that is, the greater is the value of sensing unreliability function, the worse is the sensing quality.

The distribution of information in the environment is represented by a density function, or information distribution function, and denoted by $\phi : \mathcal{Q} \to \mathbb{R}_+ : q \mapsto \phi(q)$. This density function indicates the importance of a quantity to measure at particular point q. For instance, consider an environment with Gaussian information distribution function. In this case, the information is peaked at certain point and therefore the robots should intuitively be located around the peak of the density function.

After providing the definitions of the sensing unreliability function and density function, we introduce the locational optimization problem. Generated by the sensor positions at time t, $\mathcal{P}$, we are able to use the Voronoi tessellation of $\mathcal{Q}$

given by

$$V_i(p_i) = \{q \in \mathcal{Q} : \|q - p_i\| \leqslant \|q - p_j\|, \forall p_j \in \mathcal{P}, j \neq i\}. \tag{3.1}$$

In the following discussion, we use V_i conveniently to refer to $V_i(p_i)$. With this Voronoi partitions, the objective function of the locational optimisation is formulated as

$$H(\mathcal{P}) = \sum_{i=1}^{n} \int_{V_i} g(q, p_i)\phi(q)dq. \tag{3.2}$$

With the defined conditions of the sensing unreliability function and density functions, the following lemma states the convexity of the objective function of the locational optimization.

Lemma 3.1. *Assume that the sensing unreliability function is isotropic, increasing, and convex in $p_i \in \mathcal{P}$, for all $i \in \{1, 2, \ldots, n\}$. Then, for a positive density function, the cost function H in 3.2 is convex.*

Proof. Due to isotropic property of the sensing unreliability function, we have a norm-valued function $f : \mathbb{R} \to \mathbb{R}$ that re-parametrises $g(q, p_i)$ such that $g(q, p_i) = f(\|q - p_i\|)$. By letting $H^*(p_i)$ be the i-th term of $H(\mathcal{P})$ in (3.2), we have

$$H^*(p_i) = \int_{V_i} f(\|q - p_i\|)\phi(q)dq.$$

To prove that $H^*(p_i)$ is convex in p_i, applying the property of the convex function to $f(\|q - p_i\|)$ yields

$$f(\theta\|q - z_1\| + (1 - \theta)\|q - z_2\|) \leqslant \theta f(\|q - z_1\|) + (1 - \theta)f(\|q - z_2\|) \tag{3.3}$$

by taking $\theta \in [0, 1]$, and $\|q - z_1\|, \|q - z_2\| \in \text{dom}(f)$. By utilising the triangle inequality, the domain of left-hand side of (3.3) can be written as

$$\theta\|q - z_1\| + (1 - \theta)\|q - z_2\| \geq \|\theta(q - z_1) + (1 - \theta)(q - z_2)\|.$$

Since f is increasing, substituting the above domain inequality to (3.3) yields

$$f(\|q - (\theta z_1 + (1 - \theta)z_2)\|) \leqslant \theta f(\|q - z_1\|) + (1 - \theta)f(\|q - z_2\|).$$

With the assumption of the positive density function, that is $\phi(q) \geqslant 0$ for all q, multiplying both sides by $\phi(q)$ and integrating them along the i-th Voronoi

partition V_i leads to

$$\int_{V_i} f(\|q-(\theta z_1+(1-\theta)z_2)\|)\phi(q)dq \\ \leqslant \int_{V_i} \theta f(\|q-z_1\|)\phi(q)dq + \int_{V_i}(1-\theta)f(\|q-z_2\|)\phi(q)dq.$$

In a compact expression, we have

$$H^*(\theta z_1+(1-\theta)z_2) \leqslant \theta H^*(z_1)+(1-\theta)H^*(z_2),$$

which is convex. Therefore, since H is a summation of convex functions H^*, that is,

$$H(\mathcal{P}) = \sum_{i=1}^{n} H^*(p_i),$$

we conclude that H is also convex. □

In our study, we will utilise the quadratic sensing unreliability function as, $f(\|q-p_i\|) = \|q-p_i\|^2$. With the quadratic function, we may borrow some notions of rigid body motion. Consider the mass, moment of inertia and centroid of i-th Voronoi region expressed as

$$M_{V_i} = \int_{V_i} \phi(q)dq,\ \mathcal{I}_{V_i} = \int_{V_i} q\phi(q)dq,\ \text{and}\ C_{V_i} = \frac{\mathcal{I}_{V_i}}{M_{V_i}}, \tag{3.4}$$

respectively. Therefore, applying the parallel-axis theorem of rigid-body motion [110] to the cost function (3.2) leads to an equivalent expression given by

$$H(p) = \sum_{i=1}^{n} \mathcal{I}_{V_i} + \sum_{i=1}^{n} M_{V_i}\|p_i - C_{V_i}\|^2, \tag{3.5}$$

where $p = [p_1^\top, \ldots, p_n^\top]^\top \in \mathbb{R}^{nd}$ denotes the vectorised positions of the robots. The coverage control problem could be considered as a problem that aims to designing control inputs of robots that are capable of driving them towards the optimal positions such that the objective function of the locational optimisation is minimised.

3.2 Problem Formulation

Consider n robots deployed in a convex space $\mathcal{Q} \subseteq \mathbb{R}^d$, and their connection topology represented by a connected undirected graph $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$. In our work,

we are employing the Delaunay graph resulted from Delaunay triangulation and used to generate the Voronoi tessellation in (3.1), referred to as Delaunay graph. The corresponding Laplacian of this graph is denoted by $\mathcal{L}_n \in \mathbb{R}^{n\times n}$. For agent $i \in \mathcal{V}_n$, the position of the agent is denoted by $p_i \in \mathcal{Q}$. The continuous-time dynamics of robot $i \in \mathcal{V}_n$ is given by

$$\dot{p}_i = u_i, \tag{3.6}$$

with $u_i \in \mathbb{R}^d$ denoting the control input of agent i.

Inspired by the optimisation method in [10], to solve the locational optimisation problem in a distributed fashion, we can equivalently transform (3.2) into

$$\min_{p_i} \quad \tilde{H}(p) = \sum_{i=1}^{n} \int_{V_i} f(\|q - p_i\|)\phi(q)dq, \tag{3.7a}$$

$$\text{s.t.} \quad \hat{\mathcal{L}}_n(p - C_V) = \mathbf{0}_{nd}, \tag{3.7b}$$

with $f(\|q - p_i\|) = \|q - p_i\|^2$, $\hat{\mathcal{L}}_n = \mathcal{L}_n \otimes I_d \in \mathbb{R}^{nd\times nd}$, $p = [p_1^\top, \ldots, p_n^\top]^\top \in \mathbb{R}^{nd}$ and $C_V = [C_{V_1}^\top, \ldots, C_{V_n}^\top]^\top \in \mathbb{R}^{nd}$. Furthermore, we could say that the above constrained optimisation problem is feasible because the objective function is convex and the constrain is linear.

With the transformed constrained optimisation problem and the defined agent dynamics in (3.6), the objectives of this work are:

1. to design distributed gradient-based coverage controller that minimises (3.7),

2. to design finite-time coverage controller that guarantees the convergence to the optimiser of (3.7) in finite time, and

3. to design obstacle-avoidance based coverage controller that prevent collision among agents and collission between agents and obstacles while guiding the robots towards the optimiser of (3.7).

3.3 Distributed Coverage Control Algorithms

3.3.1 Gradient-descent Technique

In our algorithm design, by adopting the distributed optimisation method in [10], the Lagrangian multiplier theory is employed to design the distributed gradient-descent based controller. The Lagrangian function of this problem, $F : \mathbb{R}^{nd} \times \mathbb{R}^{nd} \to \mathbb{R} : (p, v) \mapsto F(p, v)$, containing the distributed objective function (3.7) with the disagreement function of consensus protocol is denoted by

$$F(p, v) = \tilde{H}(p) + v^\top \hat{\mathcal{L}}_n (p - C_V) + \frac{1}{2}(p - C_V)^\top \hat{\mathcal{L}}_n (p - C_V), \tag{3.8}$$

where $v = [v_1^\top, \ldots, v_n^\top]^\top \in \mathbb{R}^{nd}$ denotes the Lagrangian multiplier of the equality constraint. The saddle point optimising $F(p, v)$, denoted by $(p^\star, v^\star)$, could be obtained by utilising the Karush-Kuhn-Tucker (KKT) conditions of the constrained optimisation in (3.7) [111]. For the optimal solution $p^\star \in \mathbb{R}^{nd}$, there exists a Lagrange multiplier $v^\star \in \mathbb{R}^{nd}$ with $\hat{\mathcal{L}}_n v^\star = \nabla_p \tilde{H}(p^\star)$ such that $(p^\star, v^\star)$ is the saddle point of $F(p, v)$.

By utilising the gradient-descent method, we can express the optimisation algorithm and the update of Lagrangian multiplier as

$$u = -\nabla_p F(p, v), \tag{3.9a}$$

$$\dot{v} = \nabla_v F(p, v), \tag{3.9b}$$

where $u = [u_1^\top, \ldots, u_n^\top]^\top \in \mathbb{R}^{nd}$ denotes the augmented control input of all agents. Accordingly, the proposed algorithm to optimise the constrained coverage control problem (3.7) is given by

$$u = -2k_p M_V (p - C_V) - k_c \hat{\mathcal{L}}_n (p - C_V) - k_v \hat{\mathcal{L}}_n v, \tag{3.10a}$$

$$\dot{v} = \hat{\mathcal{L}}_n (p - C_V), \tag{3.10b}$$

where $k_p, k_c, k_v \in \mathbb{R}$ are positive constants. In the proposed algorithm, we have used $M_V = \mathrm{diag}(M_{V_i}) \in \mathbb{R}^{nd \times nd}$ as a positive diagonal matrix containing the mass of the Voronoi regions. Element-wise expression of the control input of agent

$i \in \mathcal{V}_n$ is given by

$$u_i = -k_p M_{V_i} \tilde{p}_i - k_c \sum_{j=1}^{n} a_{ij}\big(\tilde{p}_i - \tilde{p}_j\big) - k_v \sum_{j=1}^{n} \int_{t_0}^{t} a_{ij}\big(v_i - v_j\big)dt, \tag{3.11a}$$

$$\dot{v}_i = \sum_{j=1}^{n} a_{ij}\big(\tilde{p}_i - \tilde{p}_j\big)dt, \tag{3.11b}$$

with $\tilde{p}_i = p_i - C_{V_i}$. This gradient-based method allows each agent to share the full information containing the current position, centroid, and Lagrange multiplier to the neighbouring agents. Accordingly, the integral term of the proposed control law could utilise them to reduce the steady-state error of the robot formation by driving the robots to satisfy the consensus constraint.

The following theorem states the convergence of our distributed optimisation based coverage control.

Theorem 3.2 (Convergence of Distributed Gradient-descent Coverage Control)**.** *Consider a group of n agents are connected via Delaunay graph $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$. Let the dynamics of the agents be given by (3.6), for $i \in \mathcal{V}_n$. Then, by utilising the control protocol (3.10), the positions of the robots converge to the centroid of the Voronoi tessellation, that is,*

$$\lim_{t\to\infty} p(t) = C_V. \tag{3.12}$$

Proof. Define the constant point of the Lagrangian function as $(p^\star, v^\star)$ with $p^\star, v^\star \in \mathbb{R}^{nd}$. Let $\tilde{p} = p - p^\star$ and $\tilde{v} = v - v^\star$ denote the error of the primal-dual pair. Accordingly, by employing the robot dynamics in (3.6) and the control input in (3.10), the error dynamics can be expressed as

$$\dot{\tilde{p}} = -(2k_p M_V + k_c \hat{\mathcal{L}}_n)p + (2k_p M_V + k_c \hat{\mathcal{L}}_n)p^\star - k_v \hat{\mathcal{L}}_n \tilde{v}, \tag{3.13a}$$

$$\dot{\tilde{v}} = \hat{\mathcal{L}}_n \tilde{p}, \tag{3.13b}$$

for some positive constants k_p, k_c, k_v.

Consider a Lyapunov function candidate

$$V_c = \frac{1}{2}\tilde{p}^\top \tilde{p} + \frac{1}{2}\tilde{v}^\top \tilde{v}. \tag{3.14}$$

The corresponding time derivative is given by

$$\begin{aligned}\dot{V}_c =& \tilde{p}^\top \dot{\tilde{p}} + \tilde{v}^\top \dot{\tilde{v}} \\ =& -\tilde{p}^\top (2k_p M_V + k_c \hat{\mathcal{L}}_n)\tilde{p} - k_v \tilde{p}^\top \hat{\mathcal{L}}_n \tilde{v} + \tilde{v}^\top \hat{\mathcal{L}}_n \tilde{p} \\ =& -\tilde{p}^\top (2k_p M_V + k_c \hat{\mathcal{L}}_n)\tilde{p} - (k_v - 1)\tilde{p}^\top \hat{\mathcal{L}}_n \tilde{v} \\ =& -\tilde{p}^\top (2k_p M_V + k_c \hat{\mathcal{L}}_n)\tilde{p}\end{aligned}$$

where the symmetrical property of the Laplacian matrix and $k_v = 1$ have been applied. Thus, by denoting $m_V = \min\{M_{V_1}, \ldots, M_{V_n}\} > 0$ and utilising the Courant-Fisher theorem of the Laplacian matrix, that is, $x^\top \mathcal{L}_n x \geqslant \lambda_2(\mathcal{L}_n)\|x\|^2$, we have

$$\dot{V}_c \leqslant -(2k_p m_V + k_c \lambda_2(\mathcal{L}_n))\|\tilde{p}\|^2 \leqslant 0,$$

for $p \neq p^\star$ and $\tilde{p} \neq 0$. By invoking LaSalle's invariance principle, we can say that for any $p(0)$, we have $\tilde{p} \to 0$ as $t \to \infty$ implying that $\lim_{t\to\infty} p(t) = p^\star$.

Thereafter, we need to prove that the solution $p^\star$ equals to the centroids of Voronoi regions C_V. Recall Karush-Kuhn-Tucker optimality condition of $\tilde{H}(p)$ and take the i-th term, we have

$$\nabla_{p_i} \tilde{H}_i(p_i) = \frac{\partial}{\partial p_i} \int_{V_i} \|q - p_i^\star\|^2 \phi(q) dq = \int_{V_i} 2(q - p_i^\star)\phi(q) dq = 0.$$

Rearranging the last equality yields

$$p_i^\star = \frac{\int_{V_i} q\phi(q) dq}{\int_{V_i} \phi(q) dq},$$

which equivalent to the centroid of Voronoi region, C_{V_i}, as defined in (3.4). Generalising similar arguments to any $i \in \mathcal{V}_n$ leads to $p^\star = C_V$ with $p^\star = [(p_1^\star)^\top, \ldots, (p_n^\star)^\top]^\top$. □

3.3.2 Finite-time Coverage Control

In the previous discussion, a distributed gradient-descent based coverage control protocol has been designed. However, we can observe that the algorithm requires each agent to have sufficiently high-performance communication and computation resources to exchange their position, centroid and the Lagrange multiplier, and

perform the integration to obtain the control input. In the following control design, we discard the Lagrange-multiplier term and design a coverage controller that still guarantees the convergence to the optimal positions in finite time independent of the initial positions.

Recall the Lagrangian of the constrained coverage problem. By setting the Lagrangian multiplier to zero, we have the following objective function:

$$G(p) = \tilde{H}(p) + \frac{1}{2}(p - C_V)^\top \hat{\mathcal{L}}_n (p - C_V). \tag{3.15}$$

The corresponding optimal point of this optimisation given by $p^\star = C_V - \frac{1}{2} M_V^{-1} \hat{\mathcal{L}}_n \tau_v$ for some vector $\tau_v \in \mathbb{R}^{nd}$. Due to the graph connectivity, it follows that the last term vanishes with eigenvector $\tau_v = (p_1 - C_{V_1})^\star = \ldots = (p_n - C_{V_n})^\star$ associated to the zero eigenvalue. In other words, we could say that the objective function $G(p)$ is optimal when the position of the robots converge to the optimal point $p^\star = C_V$ and the consensus is achieved.

For all agent $i \in \mathcal{V}_n$, consider the following errors: $\tilde{p}_i = p_i - C_{V_i}$, $\tilde{p}_{ij} = \text{sign}(\tilde{p}_i - \tilde{p}_j)|\tilde{p}_i - \tilde{p}_j|$. Since there are two terms to optimise in (3.15), by employing these errors, we design a controller consisting of centroid stabiliser and the consensus stabiliser. The proposed centroid stabiliser, which is responsible for driving the robots toward the centroids, is expressed as

$$u_i^g = -k_g \text{sign}(\tilde{p}_i)(|\tilde{p}_i|^{2-\frac{a}{b}} + |\tilde{p}_i|^{\frac{a}{b}}), \tag{3.16}$$

where a, b are positive odd integers satisfing $a < b$ and k_g is a positive gain. Similarly, the consensus stabiliser, assigned to maintain the consensus, is given by

$$u_i^c = -k_c \sum_{j=1}^{n} a_{ij} \text{sign}(\tilde{p}_{ij})(|\tilde{p}_{ij}|^{2-\frac{a}{b}} + |\tilde{p}_{ij}|^{\frac{a}{b}}), \tag{3.17}$$

with k_c a positive gain. Hence, the augmented controller reads

$$u_i = u_i^g + u_i^c. \tag{3.18}$$

Before stating our result of the designed control protocol, we require a set of technical lemmas defined as follows.

Lemma 3.3 ([40]). *The equilibrium point of the scalar system*

$$\dot{x} = -\alpha x^{2-\frac{m}{n}} - \beta x^{\frac{m}{n}},\ x(0) = x_0,$$

where $\alpha, \beta > 0$, and m, n are positive odd integers satisfying $m < n$, is finite-time stable with upper-bound of the settling time given by

$$T(x_0) \leqslant \frac{n\pi}{2\sqrt{\alpha\beta}(n-m)}.$$

Remark 3.4. This lemma guarantees the finite-time convergence independent to the initial value of the dynamics.

Lemma 3.5 ([40]). *Let $\zeta_1, \zeta_2, \ldots, \zeta_n \geqslant 0$. Then*

$$\sum_{i=1}^{n} \zeta_j^q \geqslant \left(\sum_{i=1}^{n} \zeta_j\right)^q,\ \text{for } q \in (0,1).$$

Lemma 3.6 ([40]). *Let $\zeta_1, \zeta_2, \ldots, \zeta_n \geqslant 0$. Then*

$$\sum_{i=1}^{n} \zeta_j^q \geqslant n^{1-q}\left(\sum_{i=1}^{n} \zeta_j\right)^q,\ \text{for } q > 1.$$

In the following theorem, we present our result about the finite-time convergence of the proposed control protocol.

Theorem 3.7 (Convergence of Finite-time Coverage Controller). *Consider a group of n agents connected via a connected Delaunay graph $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ with agent dynamics defined in (3.6). Then, there exist some constants $\kappa_1, \kappa_2 > 0$ such that the finite-time coverage problem can be solved by employing the coverage control protocol (3.18) with settling time given by*

$$T \leqslant T_{\max} = \frac{b\pi}{2\sqrt{\kappa_1\kappa_2}(b-a)},$$

where a, b are positive odd integers satisfying $a < b$.

Proof. The system dynamics of agent $i \in \mathcal{V}_n$ with the proposed control input (3.18 could be expressed as

$$\dot{p}_i = -k_g \text{sign}(\tilde{p}_i)(|\tilde{p}_i|^{2-\frac{a}{b}} + |\tilde{p}_i|^{\frac{a}{b}}) - k_c \sum_{j=1}^{n} a_{ij} \text{sign}(\tilde{p}_{ij})(|\tilde{p}_{ij}|^{2-\frac{a}{b}} + |\tilde{p}_{ij}|^{\frac{a}{b}}), \quad (3.19)$$

with $\tilde{p}_i = p_i - C_{V_i}$, $\tilde{p}_{ij} = \text{sign}(\tilde{p}_i - \tilde{p}_j)|\tilde{p}_i - \tilde{p}_j|$. The equilibrium point of the system is given by C_V with $(p_1 - C_{V_1}) = \ldots = (p_n - C_{V_n}) = 0$. By assuming this to be invariant, we have $\dot{\tilde{p}}_i = \dot{p}_i$.

Define a Lyapunov candidate:

$$V^f(\tilde{p}(t)) = \frac{1}{2}\sum_{i=1}^{n}\tilde{p}_i^2(t).$$

With the system dynamics in (3.19), the time derivative of the candidate function is given by

$$\dot{V}^f(\tilde{p}) = \dot{V}^g(\tilde{p}) + \dot{V}^c(\tilde{p}),$$

where centroid and consensus stabiliser terms are, respectively, given by

$$\begin{aligned}
\dot{V}^g(\tilde{p}) &= -k_g\sum_{i=1}^{n}\tilde{p}_i\text{sign}(\tilde{p}_i)(|\tilde{p}_i|^{2-\frac{a}{b}} + |\tilde{p}_i|^{\frac{a}{b}}), \text{ and}\\
\dot{V}^c(\tilde{p}) &= -k_c\sum_{i=1}^{n}\tilde{p}_i\sum_{j=1}^{n}a_{ij}\text{sign}(\tilde{p}_{ij})(|\tilde{p}_{ij}|^{2-\frac{a}{b}} + |\tilde{p}_{ij}|^{\frac{a}{b}}).
\end{aligned}$$

Since $|\tilde{p}_i| = \tilde{p}_i\text{sign}(\tilde{p}_i)$, the centroid stabiliser term could be written as

$$\begin{aligned}
\dot{V}^g(\tilde{p}) &= -k_g\sum_{i=1}^{n}|\tilde{p}_i|(|\tilde{p}_i|^{2-\frac{a}{b}} + |\tilde{p}_i|^{\frac{a}{b}})\\
&= -k_g\sum_{i=1}^{n}((\tilde{p}_i^2)^{\frac{3b-a}{2b}} + (\tilde{p}_i^2)^{\frac{a+b}{2b}})\\
&= -k_g\Big(\sum_{i=1}^{n}(\tilde{p}_i^2)^{\frac{3b-a}{2b}} + \sum_{i=1}^{n}(\tilde{p}_i^2)^{\frac{a+b}{2b}}\Big)\\
&\leqslant -k_g\Big(n^{\frac{a-b}{2b}}\Big(\sum_{i=1}^{n}\tilde{p}_i^2\Big)^{\frac{3b-a}{2b}} + \Big(\sum_{i=1}^{n}\tilde{p}_i^2\Big)^{\frac{a+b}{2b}}\Big)\\
&= -k_g(n^{\frac{a-b}{2b}}(2V^f(\tilde{p}))^{\frac{3b-a}{2b}} + (2V^f(\tilde{p}))^{\frac{a+b}{2b}}),
\end{aligned} \tag{3.20}$$

in which Lemmas (3.5) and (3.6) have been utilised to obtain the inequality. By utilising the property of the adjacency matrix and also $|\tilde{p}_{ij}| = \tilde{p}_{ij}\text{sign}(\tilde{p}_{ij})$, the consensus stabiliser term could be written as

$$\begin{aligned}
\dot{V}^c(\tilde{p}) &= -\frac{k_c}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}\tilde{p}_{ij}\text{sign}(\tilde{p}_{ij})(|\tilde{p}_{ij}|^{2-\frac{a}{b}} + |\tilde{p}_{ij}|^{\frac{a}{b}})\\
&= -\frac{k_c}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}((\tilde{p}_{ij}^2)^{\frac{3b-a}{2b}} + (\tilde{p}_{ij}^2)^{\frac{a+b}{2b}})\\
&= -\frac{k_c}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}((a_{ij}^{\frac{2b}{3b-a}}\tilde{p}_{ij}^2)^{\frac{3b-a}{2b}} + (a_{ij}^{\frac{2b}{a+b}}\tilde{p}_{ij}^2)^{\frac{a+b}{2b}})\\
&\leqslant -\frac{k_c}{2}\Big(n^{\frac{a-b}{b}}\Big(\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}^{\frac{2b}{3b-a}}\tilde{p}_{ij}^2\Big)^{\frac{3b-a}{2b}} + \Big(\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}^{\frac{2b}{a+b}}\tilde{p}_{ij}^2\Big)^{\frac{a+b}{2b}}\Big)
\end{aligned}$$

where the last inequality is obtained by employing Lemmas 3.5 and 3.6.

Consider two adjacency matrices of connected undirected graphs $\mathcal{G}_\alpha$ and $\mathcal{G}_\beta$ denoted by $A_\alpha = [a_{ij}^{\frac{2b}{3b-a}}] \in \mathbb{R}^{n\times n}$ and $A_\beta = [a_{ij}^{\frac{2b}{a+b}}] \in \mathbb{R}^{n\times n}$, respectively. The corresponding Laplacians are given by $\mathcal{L}_\alpha$ and $\mathcal{L}_\beta$. It follows that the inequality of the consensus stabiliser can equivalently be expressed as

$$\dot{V}^c(\tilde{p}) \leqslant -\frac{k_c}{2}(n^{\frac{a-b}{b}}(2\tilde{p}^\top \mathcal{L}_\alpha \tilde{p})^{\frac{3b-a}{2b}} + (2\tilde{p}^\top \mathcal{L}_\beta \tilde{p})^{\frac{a+b}{2b}}),$$

with $\tilde{p} = [\tilde{p}_1^\top, \ldots, \tilde{p}_n^\top]^\top \in \mathbb{R}^{nd}$. Applying the Courant-Fischer theorem of the Laplacian matrices, $\tilde{p}^\top \mathcal{L}_\alpha \tilde{p} \geqslant \lambda_2(\mathcal{L}_\alpha)\|\tilde{p}\|^2$ and $\tilde{p}^\top \mathcal{L}_\beta \tilde{p} \geqslant \lambda_2(\mathcal{L}_\beta)\|\tilde{p}\|^2$ for $\mathbf{1}_{nd}^\top \tilde{p} = \mathbf{0}_{nd}$, leads to

$$\begin{aligned}
\dot{V}^c(\tilde{p}) &\leqslant -\frac{k_c}{2}(n^{\frac{a-b}{b}}(2\lambda_2(\mathcal{L}_\alpha)\|\tilde{p}\|^2)^{\frac{3b-a}{2b}} + (2\lambda_2(\mathcal{L}_\beta)\|\tilde{p}\|^2)^{\frac{a+b}{2b}}). \\
&= -\frac{k_c}{2}(n^{\frac{a-b}{b}}(4\lambda_2(\mathcal{L}_\alpha)V^f(\tilde{p}))^{\frac{3b-a}{2b}} + (4\lambda_2(\mathcal{L}_\beta)V^f(\tilde{p}))^{\frac{a+b}{2b}}) \\
&= -\frac{k_c}{2}(n^{\frac{a-b}{b}}(4\lambda_f V^f(\tilde{p}))^{\frac{3b-a}{2b}} + (4\lambda_f V^f(\tilde{p}))^{\frac{a+b}{2b}}), \qquad (3.21)
\end{aligned}$$

in which we have used $\lambda_f = \min\{\lambda_2(\mathcal{L}_\alpha), \lambda_2(\mathcal{L}_\beta)\}$.

By adding (3.20) and (3.21) followed by some re-arrangements, the time derivative of the Lyapunov candidate can be written as

$$\begin{aligned}
\dot{V}^f(\tilde{p}) \leqslant &-\frac{1}{2}(2k_g n^{\frac{a-b}{2b}} + k_c n^{\frac{a-b}{b}}(2\lambda_f)^{\frac{3b-a}{2b}})(2V^f(\tilde{p}))^{\frac{3b-a}{2b}} \\
&-\frac{1}{2}(2k_g + k_c(2\lambda_f)^{\frac{a+b}{2b}})(2V^f(\tilde{p}))^{\frac{a+b}{2b}}.
\end{aligned}$$

By denoting $\xi = 2V^f(\tilde{p})$ and $\dot{\xi} = 2\dot{V}^f(\tilde{p})/\sqrt{2V^f(\tilde{p})}$ for $V^f(\tilde{p}) \neq 0$, we have

$$\dot{\xi} \leqslant -(2k_g n^{\frac{a-b}{2b}} + k_c n^{\frac{a-b}{b}}(2\lambda_f)^{\frac{3b-a}{2b}})\xi^{2-\frac{a}{b}} - (2k_g + k_c(2\lambda_f)^{\frac{a+b}{2b}})\xi^{\frac{a}{b}}.$$

Employing Lemma 3.3 and the Comparison Principle [108] yield the boundary of the settling time expressed as

$$T \leqslant T_{\max} = \frac{b\pi}{2\sqrt{\kappa_1\kappa_2}(b-a)},$$

with

$$\begin{aligned}
\kappa_1 &= 2k_g n^{\frac{a-b}{2b}} + k_c n^{\frac{a-b}{b}}(2\lambda_f)^{\frac{3b-a}{2b}}, \text{ and} \\
\kappa_2 &= 2k_g + k_c(2\lambda_f)^{\frac{a+b}{2b}},
\end{aligned}$$

such that $\lim_{t\to T}\|\tilde{p}_i\| = 0$, for all $i \in \mathcal{V}_n$. This completes our proof. □

Remark 3.8. The computation of the boundary of the settling time is indeed dependant to some design parameters and the algebraic graph topology but independent to the initial positions. Therefore, the robots will reach the centroids regardless of the initial positions before $T_{\max}$ given by

$$T_{\max} = \frac{b\pi}{2\sqrt{\kappa_1\kappa_2}(b-a)}, \tag{3.22}$$

with

$$\kappa_1 = 2k_g n^{\frac{a-b}{2b}} + k_c n^{\frac{a-b}{b}} (2\lambda_f)^{\frac{3b-a}{2b}}, \text{ and}$$
$$\kappa_2 = 2k_g + k_c (2\lambda_f)^{\frac{a+b}{2b}},$$

3.3.3 Coverage Control with Obstacle Avoidance

In the following discussion, we aim to design obstacle-avoidance algorithms to prevent both inter-agent collision and agent-obstacle collision using the notion of potential field algorithm. The proposed avoidance controller can be implemented along with either the gradient-based controller or the finite-time controller.

Consider m obstacles inside the convex set $\mathcal{Q}$ located at $p_l^o \in \mathcal{Q}$ for $l \in \{1, 2, \ldots, m\}$. By utilising the potential field algorithm, define the repulsive functions produced by the robots as

$$U_r = \begin{cases} \sum\limits_{j=1}^{n} \frac{1}{2} k_r \left(\frac{1}{\|p_i - p_j\|} - \frac{1}{\Delta_r} \right)^2, & \text{if } \|p_i - p_j\| \leqslant \Delta_r \\ 0, & \text{otherwise,} \end{cases} \tag{3.23}$$

and by the obstacles as

$$U_o = \begin{cases} \sum\limits_{l=1}^{m} \frac{1}{2} k_o \left(\frac{1}{\|p_i - p_l^o\|} - \frac{1}{\Delta_o} \right)^2, & \text{if } \|p_i - p_l^o\| \leqslant \Delta_o \\ 0, & \text{otherwise,} \end{cases} \tag{3.24}$$

where $k_r, k_o, \Delta_r, \Delta_o$, respectively, are positive gains, and activation radii of the corresponding potential function. The ball centered at point p_i, for $i \in \mathcal{V}_n$, with radius Δ_r is referred to as safety region of agent i; while the ball centered at p_l^o, for $l = \{1, \ldots, m\}$, with radius Δ_o is the safety region of obstacle l. In the following discussion, the safety region, avoidance regions or repulsive regions are used interchangeably. To employ this method, the gains need to be sufficiently

large in order to prevent collisions. Furthermore, the safety regions of the objects need to be chosen such that the outer point of every object is still far inside the safety ball.

Based on the defined potential function, the proposed coverage control protocol with obstacle avoidance can be expressed as

$$u_i = u_i^g + u_i^r + u_i^o \tag{3.25}$$

where u_i^g is the centroid controller, which could be chosen from either (3.11a) or (3.18). Avoidance terms u_i^r and u_i^o are, respectively, obtained by taking the gradient of the potential functions U_r and U_o with respect to the position of agent i. The avoidance terms are detailed as

$$u_i^r = \sum_{j=1}^{n} k_r \mu_{ij}^r \left(\frac{1}{\|p_i - p_j\|} - \frac{1}{\Delta_r} \right) \frac{p_i - p_j}{\|p_i - p_j\|^2},$$
$$u_i^o = \sum_{l=1}^{m} k_o \mu_{il}^o \left(\frac{1}{\|p_i - p_l^o\|} - \frac{1}{\Delta_o} \right) \frac{p_i - p_l^o}{\|p_i - p_l^o\|^2}.$$

In the above avoidance terms, some activation functions have been included. The activation functions are defined as

$$\mu_{ij}^r = \begin{cases} 1 & \text{if } \|p_i - p_j\| \leqslant \Delta_r, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\mu_{il}^o = \begin{cases} 1 & \text{if } \|p_i - p_l^o\| \leqslant \Delta_o, \\ 0 & \text{otherwise.} \end{cases}$$

Based on the defined avoidance protocol, the following theorem is established to explain the avoidance behaviour of robot $i \in \mathcal{V}_n$.

Theorem 3.9 (Avoidance Behaviour)**.** *Consider a convex set $\mathcal{Q} \subset \mathbb{R}^d$ with m obstacles and a group of n. Let the dynamics of each robot be given by (3.6) with control input (3.25). By choosing sufficiently large k_r and k_o, if robot i enters the the safety region of robot j, for $j \neq i$, or of obstacle l, for $l \in \{1, 2, \ldots, m\}$, then robot $i \in \mathcal{V}_n$ will not collide robot j or obstacle l.*

Proof. Let us firstly analyse the case of inter-robot collision where we assume that $u_i^o = 0$. Consider distance vector $\hat{p}_{ij} = p_i - p_j$, for $i \neq j$ and define a function

$V_i^r = \frac{1}{2}\|\hat{p}_{ij}\|^2$ whose time-derivative is given by $\dot{V}_i^r = \hat{p}_{ij}^\top \dot{\hat{p}}_{ij}$. With dynamics (3.6) and control input (3.25), if the robot is not inside the avoidance region of robot j, then $\dot{V}_i^r \leqslant 0$ because $u_i^r = 0$. In this situation, robot i will not collide robot j and the position converges to its Voronoi centroid. If robot i enters the avoidance region of robot j, then $u_i^r > 0$ and

$$\dot{V}_i^r = \hat{p}_{ij}^\top u_i^g + \hat{p}_{ij}^\top u_i^r = \hat{p}_{ij}^\top u_i^g + \sum_{j=1}^{n} k_r \left(\frac{1}{\hat{p}_{ij}} - \frac{1}{\Delta_r} \right).$$

By choosing a sufficiently large gain k_r such that $\dot{V}_i^r > 0$, there exists a ball $\mathcal{B}(\Delta_r, \varepsilon_j)$ centered at Δ_r with radius ε_j such that $p_i \in \mathcal{B}(\Delta_r, \varepsilon_j)$ as $\hat{p}_{ij} \to \Delta_r$. Therefore, p_i will not reach to p_j, that is, there is no collision between robot i and j, for $i \neq j$.

By, following similar arguments in the previous case, let us analyse the case of inter-robot collision where we assume that $u_i^r = 0$. Consider distance vector $\bar{p}_{il} = p_i - p_l$, for $l \in \{1, 2, \ldots, m\}$ and define a function $V_i^o = \frac{1}{2}\|\bar{p}_{il}\|^2$ whose time-derivative is given by $\dot{V}_i^o = \bar{p}_{il}^\top \dot{\bar{p}}_{il}$. With dynamics (3.6) and control input (3.25), if the robot is outside the avoidance region of obstacle l, then $\dot{V}_i^o \leqslant 0$ because $u_i^o = 0$. In this situation, robot i will not collide obstacle l and the position converges to its Voronoi centroid. If robot i enters the avoidance region of obstacle l, then $u_i^o > 0$ and we have

$$\dot{V}_i^o = \bar{p}_{il}^\top u_i^g + \bar{p}_{il}^\top u_i^o = \bar{p}_{il}^\top u_i^g + \sum_{l=1}^{m} k_o \left(\frac{1}{\bar{p}_{il}} - \frac{1}{\Delta_o} \right).$$

By choosing a sufficiently large gain k_o such that $\dot{V}_i^o > 0$, there exists a ball $\mathcal{B}(\Delta_o, \varepsilon_l)$ centered at Δ_o with radius ε_l such that $p_i \in \mathcal{B}(\Delta_o, \varepsilon_l)$ as $\hat{p}_{il} \to \Delta_o$. Therefore, p_i will not reach to p_l^o, that is, there is no collision between robot i and obstacle l. □

Remark 3.10. By choosing large gains $k_r, k_o > 0$, the position of robot i converges to the position of robot i or to the position of obstacle l even if the centroid of Voronoi region is inside the safety region.

3.4 Numerical Experiments

In this section, three numerical simulations are provided for verification of the proposed coverage controllers. The scenarios include the distributed gradient-descent, finite-time and obstacle-avoidance based controllers. The simulations are executed using MATLAB on a computer with a Linux-based operating system, 2.5-GHz processor, and 4-GB RAM.

3.4.1 Nonholonomic Robots

In our simulation, we utilise a set of nonholonomic ground robots whose dynamics are given by

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix},$$

where $p_x, p_y, \theta, v, \omega$ denote the x and y-axis positions, orientation, linear velocity and angular velocity, respectively. To overcome the nonholonomic constraints, consider the x and y-axis head positions of robot denoted by p_x^h and p_y^h, respectively (Please refer to Fig. 3.1). Then, by taking the distance between the center and head position, $R = \|p - p^h\|$, where $p = [p_x, p_y]^\top$ and $p^h = [p_x^h, p_y^h]^\top$, the kinematics of the robot can be transformed into

$$\begin{bmatrix} \dot{p}_x^h \\ \dot{p}_y^h \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -R\sin\theta \\ \sin\theta & R\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}.$$

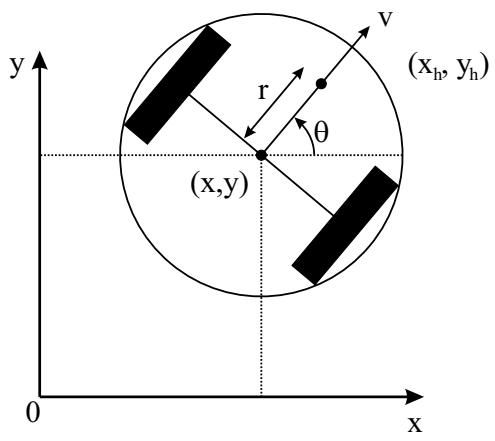


Figure 3.1: Coordinates of the differential drive robot.

Accordingly, the transformed dynamics could be controlled via

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\frac{1}{R}\sin\theta & \frac{1}{R}\cos\theta \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}. \tag{3.26}$$

Further, by denoting $u = [u_x, u_y]^\top$, we may express the dynamics of the robot as

$$\dot{p^h} = u, \tag{3.27}$$

with some boundary on the angular velocity. The boundary can be tuned in simulation, but the design of the analytical boundary is beyond our work.

3.4.2 Gradient-descent based Coverage Control

Given $n = 16$ robots randomly scattered in a square area $\mathcal{Q} \in \mathbb{R}^2$ whose boundaries are $\{(0,0),(0,2),(2,2),(2,0)\}$. The information distribution of this scenario is a double-peak exponential function given by

$$\phi(x,y) = \phi_1(x,y) + \phi_2(x,y), \tag{3.28}$$

where

$$\phi_1(x,y) = \omega_1 \exp\left(-k_1(a_1(x-x_1^c)^2 + b_1(y-y_1^c)^2 + (r_1^c)^2)\right), \tag{3.29}$$

$$\phi_2(x,y) = \omega_2 \exp\left(-k_2(a_2(x-x_2^c)^2 + b_2(y-y_2^c)^2 + (r_2^c)^2)\right), \tag{3.30}$$

with $\omega_1 = 10000,\ k_1 = 2.6,\ a_1 = 1,\ b_1 = 1,\ x_1^c = 1.5,\ y_1^c = 0.5,\ r_1^c = 1$ and $\omega_2 = 10000,\ k_2 = 2.5,\ a_2 = 1,\ b_2 = 1,\ x_2^c = 0.5,\ y_2^c = 1.5,\ r_2^c = 1$. This information distribution has two peaks, with one peak is higher than the other. The gains of the controller are $k_p = k_c = 1$ for all agents. In the simulation, the numerical integration is performed every $0.1s$ for 200 steps, equivalent to $20s$.

For instance, this scenario illustrates how a disastrous environment might possess two sources of fire with a limited communication network. A group of robots are assigned to find the sources and map the environment based on their temperature and light intensity sensors. Accordingly, because of the limited communication, some robots could be located close to the fire sources while the rest could also provide a communication relay for the entire flock.

By applying the gradient-descent based control protocol in (3.11a) to the robot dynamics (3.27), we obtain the resulting trajectory of robots and the Voronoi

partition illustrated in Fig. 3.2. We can also observe the trajectory of the Euclidean error $\|p_i - C_{V_i}\|$ and the objective function of this gradient-based controller case depicted in Figs. 3.3 and 3.4, respectively.

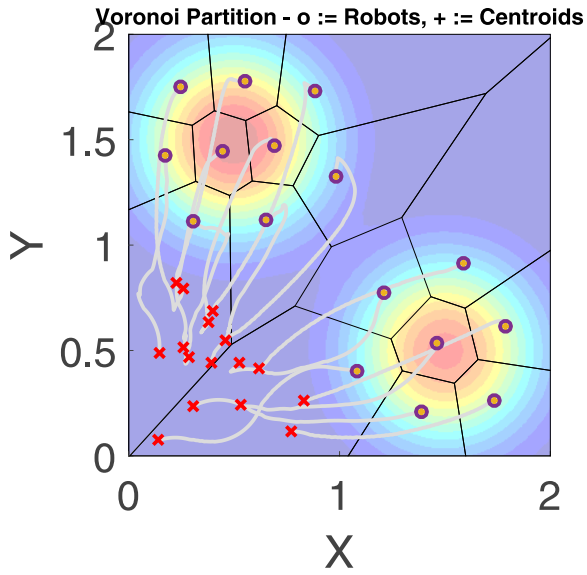


Figure 3.2: Trajectories and optimal centroidal Voronoi regions with double-peak distribution.

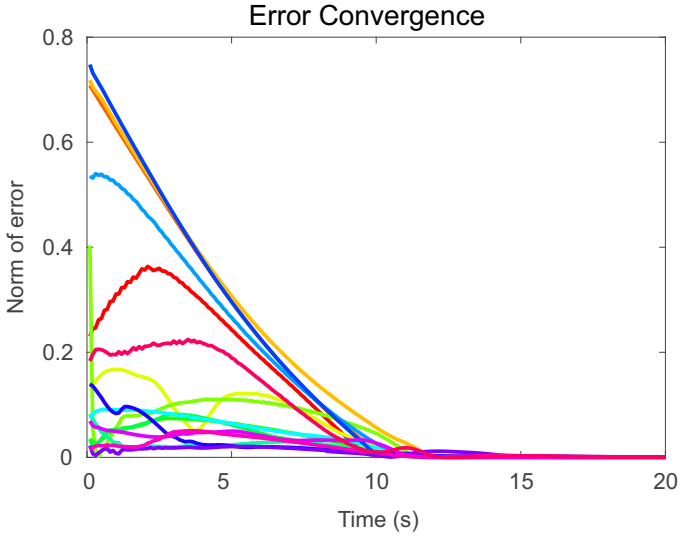


Figure 3.3: Convergence result of the centroid errors of coverage problem with double-peak distribution.

Fig. 3.2 demonstrates that the controller has successfully drive the robots such that their positions align with their centroids. Since the information distribution has two peaks with different heights, we observe that the number of robot deployed

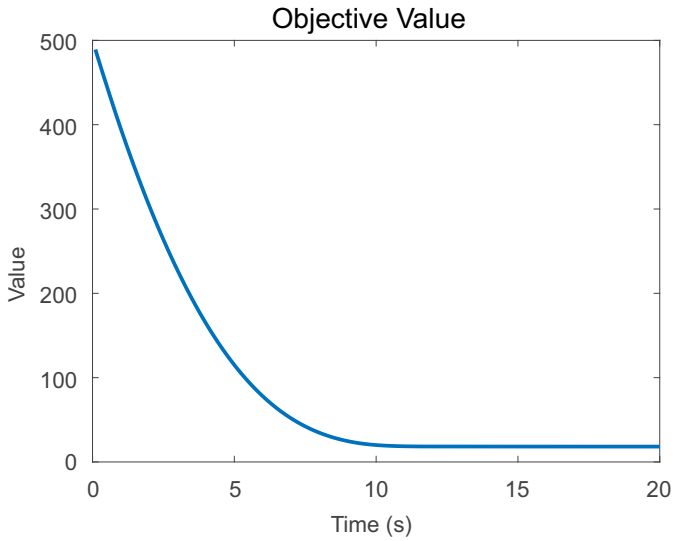


Figure 3.4: Convergence result of the objective function of coverage problem with double-peak distribution.

to the area with higher peak (upper-left peak) is higher than to area with lower peak (lower-right peak). In Fig. 3.4, we also see the convergence of the objective function to an optimal value when the centroids are reached. However, the figure shows that the objective value does not equal to zero. This phenomenon is caused by the nonholonomic constraints of the mobile robots preventing them to translate to the optimal location when they have been sufficiently close to the optimal location. Furthermore, the large value of the information distribution amplifies the small error caused by the nonholonomic constraints leading to non-zero objective value. These simulation results imply that the protocol (3.11a) can successfully solve the coverage control problem and drive the mobile robots with dynamics (3.27) close to the optimal positions, i.e., the Voronoi centroids.

3.4.3 Finite-time Coverage Control

Given $n = 16$ robots randomly deployed in a square area $\mathcal{Q} \in \mathbb{R}^2$ whose boundaries are $\{(0,0),(0,2),(2,2),(2,0)\}$. The information distribution of this scenario is uniform, that is,

$$\phi(x,y) = 1. \tag{3.31}$$

The parameters of the finite-time controller are $a = 21,\ b = 23,\ k_g = 0.4,\ k_c = 0.3$ and lower bound of the non-zero eigenvalue of graph Laplacian is $\lambda_f = 0.4615$.

By utilising (3.22), we may obtain the estimate maximum settling time is $T_{\max} = 18.0816s$. In the simulation, the numerical integration is performed every $0.1s$ for 200 steps, equivalent to $20s$.

By applying the gradient-descent based control protocol in (3.18) to the robot dynamics (3.27), we obtain the resulting trajectory of robots and the Voronoi partition illustrated in Fig. 3.5. We can also observe the trajectory of the Euclidean error $\|p_i - C_{V_i}\|$ and the objective function of this finite-time case depicted in Figs. 3.6 and 3.7, respectively.

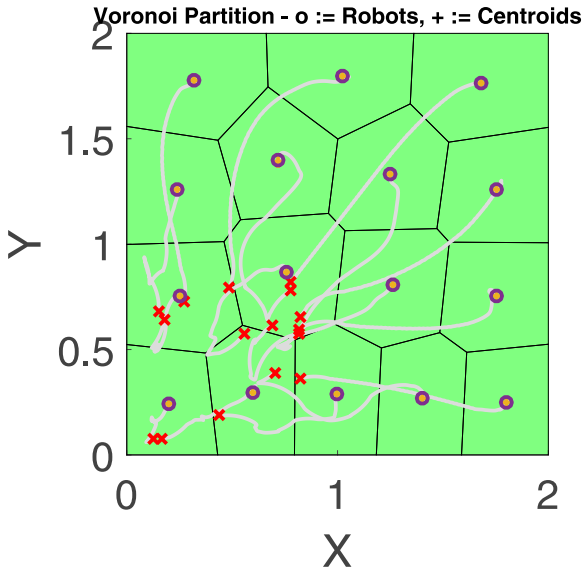


Figure 3.5: Trajectories and optimal centroidal Voronoi regions with double-peak distribution.

Fig. 3.5 demonstrates that the controller has successfully drive the robots such that their positions align with their centroids. Since the density function is uniform within the boundary, we observe that the number of robot deployed per a unit area is constant. Fig. 3.6 verifies that the error between the position and the optimal position is minimised before the estimate settling time $T_{\max}$. In Fig. 3.7, we also see the convergence of the objective function to an optimal value when the centroids are reached. Similar to the phenomenon in the previous gradient-based coverage simulation, the figure shows that the objective value does not equal to zero. The nonholonomic constraints of mobile robots might also cause this phenomenon. However, the objective value of this simulation is significantly smaller compared to the previous one because the information distribution of this

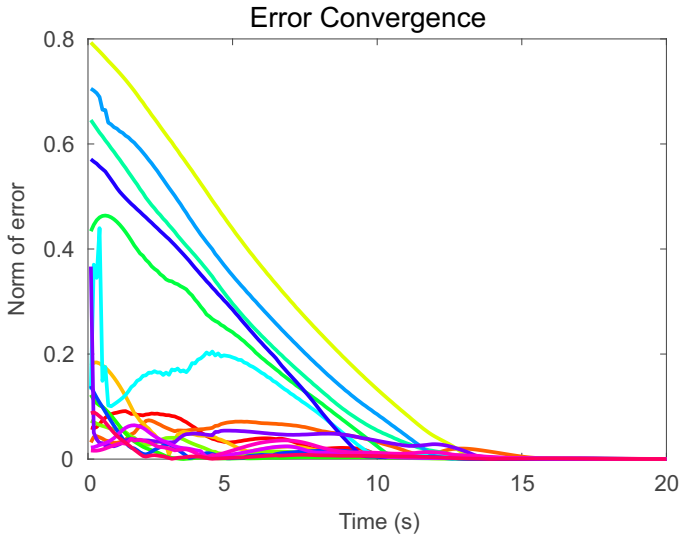


Figure 3.6: Convergence result of the centroid errors of coverage problem with double-peak distribution.

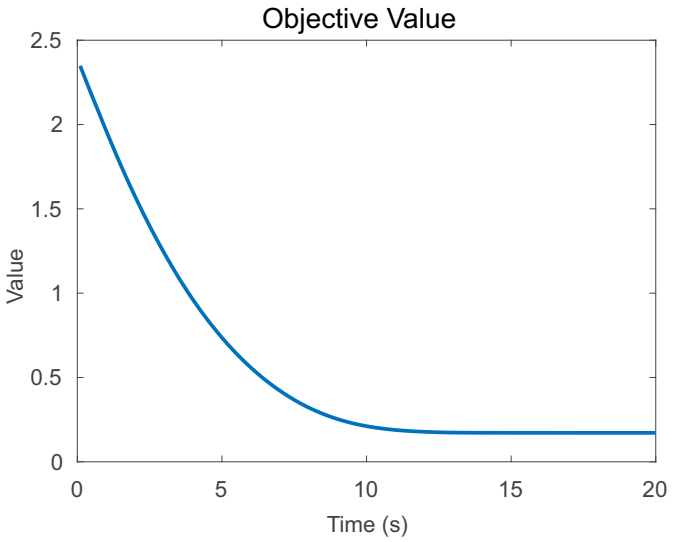


Figure 3.7: Convergence result of the objective function of coverage problem with double-peak distribution.

scenario is also significantly smaller than the gradient-based coverage simulation. These simulation results imply that the protocol (3.18) can successfully solve the coverage control problem and drive the mobile robots with dynamics (3.27) close to the optimal positions with a finite-time convergence.

3.4.4 Gradient-descent based Coverage Control with Obstacle Avoidance

Given $n = 16$ robots randomly located in a square area $\mathcal{Q} \in \mathbb{R}^2$ whose boundaries are $\{(0,0),(0,2),(2,2),(2,0)\}$. The information distribution of this scenario has one peak given by

$$\phi(x,y) = \omega_0 \exp\left(-k_0(a_0(x-x_0^c)^2 + b_0(y-y_0^c)^2 + (r_0^c)^2)\right), \tag{3.32}$$

with $\omega_0 = 1000,\ k_0 = 2.0,\ a_0 = b_0 = 1.0,\ x_0^c = 1.3,\ y_0^c = 1.3,\ r_0^c = 1.5$. There are two obstacles in the area, located at $p_1^o = (1.3, 0.3)$ and $p_2^o = (0.2, 1.2)$. The gains of the optimisation with obstacle avoidance controller are $k_p = k_c = 1.0$ and $k_r = k_o = 15.0$ for all agents. The radius of the safety regions are $\Delta_r = 0.1$ and $\Delta_o = 0.4$. In the simulation, the numerical integration is performed every $0.1s$ for 200 steps, equivalent to $20s$. It is worth noticing that the obstacles are not located on the straight line connecting the robots' initial positions and their goals to avoid local minimum because of the potential-field-based obstacle avoidance.

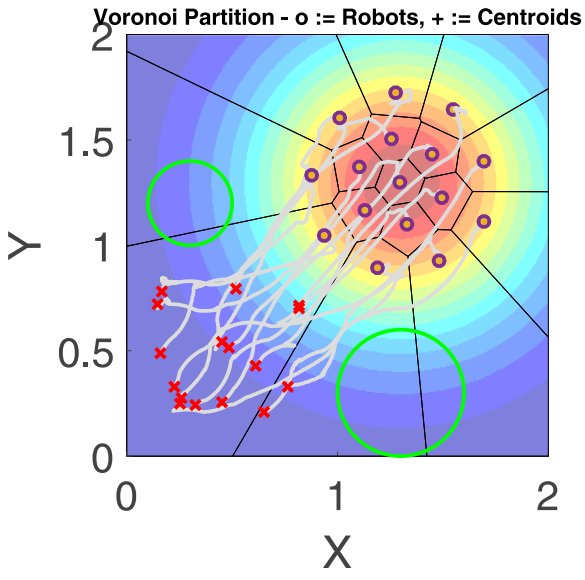


Figure 3.8: Trajectories and optimal centroidal Voronoi regions with double-peak distribution.

By applying the gradient-descent based control protocol in (3.25) to the robot dynamics (3.27), we obtain the resulting trajectory of robots and the Voronoi partition illustrated in Fig. 3.8. We can also observe the trajectory of the

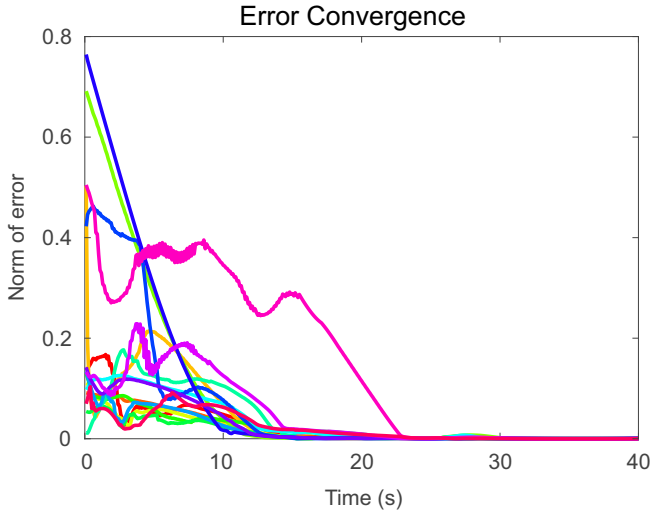


Figure 3.9: Convergence result of the centroid errors of coverage problem with double-peak distribution.

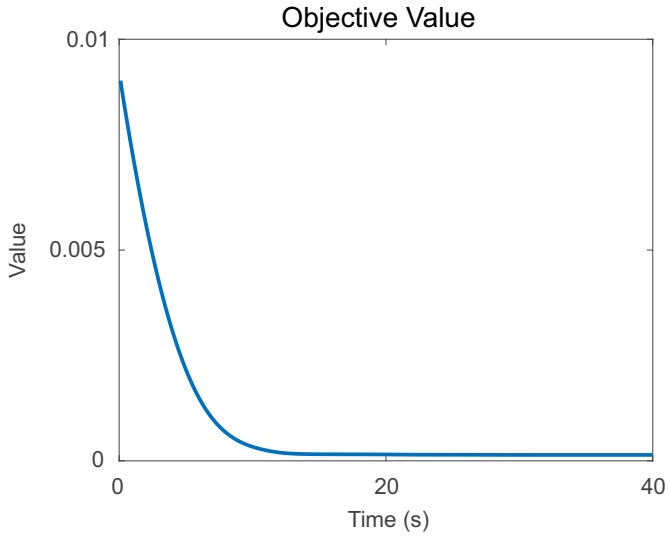


Figure 3.10: Convergence result of the objective function of coverage problem with double-peak distribution.

Euclidean error $\|p_i - C_{V_i}\|$ and the objective function of this obstacle avoidance case depicted in Figs. 3.9 and 3.10, respectively.

Fig. 3.8 demonstrates that the controller has safely drive the robots such that their positions align with their centroids. Since the density function has only one peak, we observe that all robots are deployed around its peak. Fig. 3.9 shows that the error between the position and the centroid converge to zero. In Fig. 3.10, we also see the convergence of the objective function to an optimal value

when the centroids are reached. Similar to the phenomenon in the gradient-based coverage simulation without collision avoidance behaviour, this result also shows that the objective value does not equal to zero due to the nonholonomic constraints of mobile robots. However, the objective value of this simulation is significantly smaller compared to the gradient-based simulation without obstacle because the information distribution in this scenario is also significantly smaller. These simulation results suggest that the protocol (3.25) can successfully solve the coverage control problem and drive the mobile robots with dynamics (3.27) to the optimal positions without colliding obstacle or another neighbouring robot.

3.5 Conclusions

In this chapter, we have investigated some control protocols to overcome coverage control problem with various scenarios. In our analysis of the gradient-descent optimisation of the coverage control problem, we have transformed the traditional locational optimisation into the distributed version. We have proved that the proposed gradient-based protocol can distributedly drive all robots to the centroids of the Voronoi region. Afterwards, to reduce the communication burden, we have proposed a coverage controller that guarantees the convergence to the centroid positions in finite time. By choosing some appropriate parameters, the protocol has been proved to be capable of guiding the robots to the optimal solutions before the maximum settling time regardless of their initial positions. Furthermore, we have also considered the scenario when obstacles are in the environment, which an obstacle-avoidance based coverage protocol is proposed accordingly using the repulsive potential field method. By choosing some large avoidance parameters, the protocol is capable of preventing collisions among agents and between an agent and obstacles. Finally, some simulations have been carried out on a group of nonholonomic mobile robots to validate the performance of our proposed controllers.

Although the proposed algorithms have successfully solved coverage control problem in a distributed manner, the implementation of these algorithms requires all agents to know the information distribution of the environment prior to the

execution. To relax this situation, investigation on a distributed estimation algorithm will be carried out in the next chapter to enable implementation of coverage control problem of robotic sensor network in an area with unknown information distribution.

Chapter 4

Distributed Unscented Kalman Filter with Communication Protocol

In this chapter, we present an optimal design of nonlinear Kalman filter to estimate the state of a dynamical process in a distributed manner using the shared information among the agents. Different from existing techniques in literature, such as in [75, 14, 77, 78, 13], our algorithm generalises the distributed Kalman filter to accomodate any communication mechanism which uses not only the measurement from an agent's own sensor as consideration for estimating the process, but also the shared information from the neighbouring agents. In addition to standard Kalman filter, this distributed filtering algorithm also considers noisy communicated information within the network. The proposed technique alters the procedure in the prediction and update steps of the unscented Kalman filter. In the prediction, the prior state, measurement and communication estimates and their covariances are designed utilising the sigma points calculated from the previous estimate and covariance. While, in the subsequent update step, the posterior estimator of this filter comprises the prior estimate term, measurement correction term and communication correction term - leading to the posterior covariance containing the prior state, measurement and communication covariances. Afterwards, the Kalman and communication gains of the estimator are optimally designed. It can be said that solving the optimal gain of existing distributed

Kalman filter with consensus algorithm is a special case of this technique. To verify the effectiveness of the proposed algorithm, it is then applied to estimate the information distribution of optimal coverage problem in an area by a sensor network. Different with the state-of-the-art field estimation algorithm reported in [21], the proposed filter in this thesis has considered the noises in measurement and in the communication network and also optimised the estimator gains in every iteration to avoid instability of the system caused by failure to choose the appropriate gains.

This chapter is structured as follows. Section 4.1 defines the formulation of the general distributed estimation problem. Section 4.2 presents the result on the distributed Kalman-based nonlinear estimation, followed by Section 4.3 discussing the application of the proposed filter to the field estimation of the optimal coverage problem. Afterwards, in Section 4.4, comparative simulations on optimal coverage problem using distributed Kalman filter algorithm and the modified-consensus observer in [21] are presented to validate the performance of the proposed methods. Finally, Section 4.5 concludes this work.

4.1 Problem Formulation

Consider a group of n agents connected in a network whose topology is represented by a graph $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$, for $\mathcal{V}_n = \{1, 2, \ldots, n\}$ and $\mathcal{E}_n \subseteq \mathcal{V}_n \times \mathcal{V}_n$. These agents are assigned to estimate a dynamical process represented by a function $f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$, such that

$$\xi_{k+1} = f(\xi_k, \xi_k^v), \tag{4.1}$$

where $\xi_k \in \mathbb{R}^d$ and $\xi_k^v \in \mathbb{R}^d$, respectively, denote the unknown state, and the process noise at time step k, with $k \in \mathbb{Z}^*$.

To estimate the process (4.1) using a group of networked agents in a distributed manner, each agent is equipped with a sensor whose measurement data is modelled by a function $h_i : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^q$ such that

$$\zeta_k^i = h_i(\xi_k, w_k^i), \text{ for } i \in \mathcal{V}_n, \tag{4.2}$$

where $\zeta_k^i \in \mathbb{R}^q$ and $w_k^i \in \mathbb{R}^q$, respectively, denote the sensing data and the measurement noise at time step k.

Since each agent within a network requires information from its neighbours to colaboratively estimate a dynamical process, we need to model the communication network of the agents and define how each agent use the shared information. As a proposed mechanism in the distributed nonlinear Kalman filtering, the communication model of a noisy network at time step k is represented by a function $G : \mathbb{R}^{nd} \times \mathbb{R}^{nr} \to \mathbb{R}^{nr}$, i.e.,

$$z_k = G(x_k, s_k), \tag{4.3}$$

where, in this communication model, $x_k = \mathbf{1}_n \otimes \xi_k$, $z_k = \mathrm{vec}_n(z_k^i) \in \mathbb{R}^{nr}$ and $s_k = \mathrm{vec}_n(s_k^i) \in \mathbb{R}^{nr}$ are the augmented state to estimate, the communicated data and the communication noise, respectively.

Remark 4.1. The communication model of a network provides a representation of the connection topology of a network. For example, consider a group of n robots are assigned to estimate d unknown states, $\xi_k \in \mathbb{R}^d$, in which the data received by agent i is denoted by $z_k^i \in \mathbb{R}^d$ and the communication noise by $s_k^i \in \mathbb{R}^r$. Suppose that the communication topology is modelled via a graph with Laplacian matrix $\hat{\mathcal{L}}_n = \mathcal{L}_n \otimes \mathrm{I}_d \in \mathbb{R}^{nd \times nd}$. Then, using the defined model, the communication process can be expressed as $z_k = \hat{\mathcal{L}}_n x_k + s_k$.

Throughout this chapter, to improve the brevity of the filter design and formulation, augmented expressions of the process, measurement and communication dynamics are equivalently described as

$$x_{k+1} = F(x_k, v_k) \tag{4.4}$$

$$y_k = H(x_k, w_k) \tag{4.5}$$

$$z_k = G(x_k, s_k) \tag{4.6}$$

where $y_k = \mathrm{vec}_n(\zeta_k^i)$, $F = \mathrm{vec}_n(f_i)$, $H = \mathrm{vec}_n(h_i)$, $v_k = \mathbf{1}_n \otimes \xi_k^v$, and $w_k = \mathrm{vec}_n(w_k^i)$ with $\mathrm{vec}_n(\cdot)$ denoting the vectorisation operator.

There are some assumptions made regarding the noises, given as follows.

Assumption 4.2. *The expected values of process, measurement and communication noises are zero, i.e.,* $\mathbb{E}\{v_k\} = 0$, $\mathbb{E}\{w_k\} = 0$ *and* $\mathbb{E}\{s_k\} = 0$.

Assumption 4.3. *The covariances of process, measurement and communication noises of every agent i, j are $\mathbb{E}\{v_k v_k^\top\} = Q_k$, $\mathbb{E}\{w_k w_k^\top\} = R_k$, and $\mathbb{E}\{s_k s_k^\top\} = S_k$, where $Q_{k,ij} = Q_k^i \delta_{ij}$, $R_{k,ij} = R_k^i \delta_{ij}$, and $S_{k,ij} = S_k^i \delta_{ij}$, for some matrices Q_k^i, R_k^i, S_k^i and the Kronecker delta δ_{ij}.*

Assumption 4.4. *The process, measurement and communication noises are uncorrelated, i.e., $\mathbb{E}\{v_k w_k^\top\} = 0$, $\mathbb{E}\{v_k s_k^\top\} = 0$, $\mathbb{E}\{w_k s_k^\top\} = 0$.*

Assumption 4.5. *The process, measurement and communication estimates of agent i and j, for $i \neq j$, are uncorrelated, i.e., their cross-covariance matrices are zero.*

The objective of the distributed estimation is to minimise the collective expected value of mean-squared error (MSE) of the estimation of the agents. The performance index is formulated as

$$\min_{\hat{x}_k} \ \mathbb{E}\{(x_k - \hat{x}_k)^\top (x_k - \hat{x}_k)\} \tag{4.7}$$

where $\hat{x}_k = \mathrm{vec}_n(\hat{x}_k^i)$ is the estimate value of x_k. In the following filter design, the estimate value of x_k refers to the posterior estimate of the Kalman filter.

In order to analyse the boundedness of a filtering algorithm, the following boundedness lemma of a stochastic process, such as presented in [112, 14], is required.

Lemma 4.6 (Stochastic Boundedness)**.** *If, for $\theta \in \mathbb{R}^d$ being a stochastic process at time step k, $k \in \mathbb{Z}^*$, there exists a scalar stochastic process $V(\varepsilon_k)$ with $\varepsilon_k = \theta_k - \theta'_k$ satisfying these conditions:*

1. *$\underline{v}\|\varepsilon_k\|^2 \leqslant V(\varepsilon_k) \leqslant \overline{v}\|\varepsilon_k\|^2$, for $\underline{v}$, $\overline{v} > 0$,*
2. *$\mathbb{E}\{V(\varepsilon_{k+1})|\varepsilon_k\} \leqslant \mu + (1-\sigma)V(\varepsilon_k)$, for $\mu > 0$, $0 \leqslant \sigma \leqslant 1$.*

Then, the stochastic process ε_k is exponentially bounded in mean square such that

$$\mathbb{E}\{\|\varepsilon_k\|^2\} \leqslant \frac{\overline{v}}{\underline{v}}\mathbb{E}[\|\varepsilon_0\|^2](1-\sigma)^k + \frac{\mu}{\underline{v}}\sum_{l=1}^{k-1}(1-\sigma)^l. \tag{4.8}$$

There are some useful variables that are necessary for the filter design. $\hat{x}_k$, $\bar{x}_k$, $\bar{y}_k$, and $\bar{z}_k$, respectively, denote the posterior state estimate of x_k, prior state estimate of x_k, prior measurement estimate of y_k, and prior communication estimate

of z_k at time step k, for $k \in \mathbb{Z}^*$. Related to the covariance update, $\hat{P}_{xx,k}$ denotes the posterior covariance of the error of the posterior estimates; $\bar{P}_{xx,k}$, $\bar{P}_{yy,k}$, $\bar{P}_{zz,k}$, $\bar{P}_{xy,k}$, $\bar{P}_{xz,k}$, and $\bar{P}_{yz,k}$, respectively, denote the prior state-to-state, measurement-to-measurement, communication-to-communication, state-to-measurement, state-to-communication and measurement-to-communication covariances.

4.2 Distributed Nonlinear Filter Design

In this filter design, the state, measurement and communication models are given in (4.4), (4.5) and (4.6), respectively, and these functions are considered to be nonlinear.

4.2.1 Prediction Update

The means of unscented transformation used in the Unscented Kalman Filter is modified in the prediction step to produce the predicted state, measurement and communication estimates and their covariances.

Define $\tilde{x}_k = [\hat{x}_k^\top \; v_k^\top \; w_k^\top \; s_k^\top]^\top$ as a vector of random variables augmenting the state and noises; while $\tilde{P}_k = \text{diag}(P_{xx,k}, Q_k, R_k, S_k)$ as the augmented covariance at time step k. The augmented form of sigma vectors at time step $k-1$ is defined as $\mathcal{X}_{k-1} = \left[(\mathcal{X}_{k-1}^x)^\top \; (\mathcal{X}_{k-1}^v)^\top \; (\mathcal{X}_{k-1}^w)^\top \; (\mathcal{X}_{k-1}^s)^\top\right]^\top \in \mathbb{R}^{a\times(2a+1)}$, $a = (2d+q+r)n$. Its entries are given by

$$\mathcal{X}_{k-1} = \left[\; \tilde{x}_{k-1} \quad (\mathbf{1}_a^\top \otimes \tilde{x}_{k-1}) + \sqrt{\kappa \tilde{P}_{k-1}} \quad (\mathbf{1}_a^\top \otimes \tilde{x}_{k-1}) - \sqrt{\kappa \tilde{P}_{k-1}} \;\right], \tag{4.9}$$

with $\kappa = a + b$. Expression $\mathbf{1}_a^\top \otimes \tilde{x}_{k-1}$ augments the previous posterior estimates to comply with the dimension of the augmented state covariance. Here, b is a scaling parameter expressed as $b = \varrho_1^2(a + \varrho_2) - a$, where ϱ_1 and ϱ_2 correlate with the spread of sigma points around $\tilde{x}_{k-1}$. The details regarding the criterion of these parameters are provided in [113].

The sigma vector is subsequently transformed through the nonlinear functions. For every l-th column of $\mathcal{X}_{k-1}$, with $l \in \{0, 1, 2, \ldots, 2a\}$, the sigma vector is mapped through (4.4) such that the prior sigma vector of the state can be written

as

$$\bar{\mathcal{X}}_{k,l} = F(\mathcal{X}^x_{k-1,l},\ \mathcal{X}^v_{k-1,l}). \tag{4.10}$$

The prior estimate $\bar{x}_k$ and covariance $\bar{P}_{xx,k}$ are attained using the approximated weighted mean and covariance of the sigma points in the form of

$$\bar{x}_k = \sum_{l=0}^{2a} W_l^m \bar{\mathcal{X}}_{k,l}, \tag{4.11}$$

$$\bar{P}_{xx,k} = \sum_{l=0}^{2a} W_l^c \left[\bar{\mathcal{X}}_{k,l} - \bar{x}_k\right] \left[\bar{\mathcal{X}}_{k,l} - \bar{x}_k\right]^\top, \tag{4.12}$$

where the weights are given by

$$\begin{aligned} W_0^m &= b/\kappa, \\ W_0^c &= b/\kappa + (1 - \varrho_1^2 + \varrho_2), \\ W_l^m &= W_l^c = 1/(2\kappa),\ 0 < l \leqslant 2a. \end{aligned}$$

To predict the measurement data, the sigma vector of the state in (4.10) is also mapped through the nonlinear measurement function (4.5) such that the sigma vector of the measurement is formulated as

$$\bar{\mathcal{Y}}_{k,l} = H(\bar{\mathcal{X}}_{k,l},\ \mathcal{X}^w_{k-1,l}). \tag{4.14}$$

Accordingly, the prior measurement estimate and its covariance, $\bar{y}_k$ and $\bar{P}_{yy,k}$, are calculated using the approximation of weighted mean and covariance, i.e.,

$$\bar{y}_k = \sum_{l=0}^{2a} W_l^m \bar{\mathcal{Y}}_{k,l}, \tag{4.15}$$

$$\bar{P}_{yy,k} = \sum_{l=0}^{2a} W_l^c \left[\bar{\mathcal{Y}}_{k,l} - \bar{y}_k\right] \left[\bar{\mathcal{Y}}_{k,l} - \bar{y}_k\right]^\top, \tag{4.16}$$

respectively.

The prediction of the communication data and its covariance can also be attained by mapping the sigma vector (4.10) to the communication function (4.6), yielding the communication sigma vector given by

$$\bar{\mathcal{Z}}_{k,l} = G(\bar{\mathcal{X}}_{k,l},\ \mathcal{X}^s_{k-1,l}). \tag{4.17}$$

The associated approximation of the weighted mean and covariance can be expressed as

$$\bar{z}_k = \sum_{l=0}^{2a} W_l^m \bar{\mathcal{Z}}_{k,l}, \tag{4.18}$$

$$\bar{P}_{zz,k} = \sum_{l=0}^{2a} W_l^c \left[\bar{\mathcal{Z}}_{k,l} - \bar{z}_k\right] \left[\bar{\mathcal{Z}}_{k,l} - \bar{z}_k\right]^\top. \tag{4.19}$$

The prior cross covariance matrices between the state estimate and measurement data, between state estimate and communication data, and between measurement and communication data can, respectively, also be formulated utilising the sigma vectors declared previously in the form of

$$\bar{P}_{xy,k} = \sum_{l=0}^{2a} W_l^c \left[\bar{\mathcal{X}}_{k,l} - \bar{x}_k\right] \left[\bar{\mathcal{Y}}_{k,l} - \bar{y}_k\right]^\top, \tag{4.20}$$

$$\bar{P}_{xz,k} = \sum_{l=0}^{2a} W_l^c \left[\bar{\mathcal{X}}_{k,l} - \bar{x}_k\right] \left[\bar{\mathcal{Z}}_{k,l} - \bar{z}_k\right]^\top, \tag{4.21}$$

and

$$\bar{P}_{yz,k} = \sum_{l=0}^{2a} W_l^c \left[\bar{\mathcal{Y}}_{k,l} - \bar{y}_k\right] \left[\bar{\mathcal{Z}}_{k,l} - \bar{z}_k\right]^\top. \tag{4.22}$$

Remark 4.7. Notice that the prior state, measurement and communication covariances are symmetric matrices, i.e., $\bar{P}_{xx,k} = \bar{P}_{xx,k}^\top$, $\bar{P}_{yy,k} = \bar{P}_{yy,k}^\top$, and $\bar{P}_{zz,k} = \bar{P}_{zz,k}^\top$.

4.2.2 Measurement Update

In this chapter, an estimator at time step k append the additional communication term as a correcting parameter to ensure the collective estimate are achieved. The proposed estimator is formulated as

$$\hat{x}_k = \bar{x}_k + K_k(y_k - \bar{y}_k) + C_k(z_k - \bar{z}_k). \tag{4.23}$$

This estimator comprises three terms: the first is the prior state estimate; the second term is the conditional-mean update of Bayesian estimation method to interfere the measurement data with the prior measurement estimate; and the last term is the proposed communication term to correct the predicted communication data with the information obtained from neighbouring agents. The parameters $K_k = \text{diag}_n(K_k^i) \in \mathbb{R}^{nd*nq}$ and $C_k = \text{diag}_n(C_k^i) \in \mathbb{R}^{nd*nr}$, respectively, denote the

Kalman gain and communication gains. In order to minimise the cost function (4.7), one has to find the optimal value of these gains.

Recall the cost function in (4.7) which can equivalently be expressed as

$$\mathbb{E}\{(x_k - \hat{x}_k)^\top (x_k - \hat{x}_k)\} = \mathbb{E}\{\mathrm{tr}\left((x_k - \hat{x}_k)(x_k - \hat{x}_k)^\top\right)\}, \tag{4.24}$$

in which $\mathbb{E}[\mathrm{tr}\{u\}] = \mathrm{tr}\{\mathbb{E}[u]\}$ and $\mathrm{tr}\{u^\top u\}] = \mathrm{tr}\{uu^\top\}]$ have been applied. Based on (4.24), minimising the mean-squared error is equivalent to minimising the trace of the covariance of the estimate error, i.e.,

$$\hat{P}_{xx,k} = \mathbb{E}\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)^\top\}. \tag{4.25}$$

Let $\bar{e}_k^x = x_k - \bar{x}_k$ denote the error between the unknown state and the prior estimate; $\bar{e}_k^y = y_k - \bar{y}_k$ the error between the measurement data and its prior estimate; and $\bar{e}_k^z = z_k - \bar{z}_k$ the error between communication data and its prior estimate. Using these error parameters, the error of the posterior estimate $\hat{e}_k^x = x_k - \hat{x}_k$ is given by

$$\hat{e}_k^x = \bar{e}_k^x - K_k \bar{e}_k^y - C_k \bar{e}_k^z \tag{4.26}$$

where the posterior estimate in (4.23) has been inserted. Afterwards, substituting (4.26) to the posterior covariance in (4.25) and applying the linearity property of expected value result in the posterior covariance of estimate error:

$$\begin{aligned}\hat{P}_{xx,k} = {} & \mathbb{E}\{\bar{e}_k^x(\bar{e}_k^x)^\top\} - K_k\mathbb{E}\{\bar{e}_k^y(\bar{e}_k^x)^\top\} - \mathbb{E}\{\bar{e}_k^x(\bar{e}_k^y)^\top\}K_k^\top - C_k\mathbb{E}\{\bar{e}_k^z\bar{e}_x^\top\} - \mathbb{E}\{\bar{e}_k^x(\bar{e}_k^z)^\top\}C_k^\top \\ & + K_k\mathbb{E}\{\bar{e}_k^x(\bar{e}_k^x)^\top\}K_k^\top + C_k\mathbb{E}\{\bar{e}_k^y(\bar{e}_k^y)^\top\}C_k^\top + K_k\mathbb{E}\{\bar{e}_k^y(\bar{e}_k^z)^\top\}C_k^\top + C_k\mathbb{E}\{\bar{e}_k^z(\bar{e}_k^y)^\top\}K_k^\top.\end{aligned} \tag{4.27}$$

Notice that $\bar{P}_{xx,k} = \mathbb{E}[\bar{e}_k^x(\bar{e}_k^x)^\top]$, $\bar{P}_{yy,k} = \mathbb{E}[\bar{e}_k^y(\bar{e}_k^y)^\top]$, $\bar{P}_{zz,k} = \mathbb{E}[\bar{e}_k^z(\bar{e}_k^z)^\top]$, $\bar{P}_{xy,k} = \mathbb{E}[\bar{e}_k^x(\bar{e}_k^y)^\top]$, $\bar{P}_{xz,k} = \mathbb{E}[\bar{e}_k^x(\bar{e}_k^z)^\top]$, $\bar{P}_{yz,k} = \mathbb{E}[\bar{e}_k^y(\bar{e}_k^z)^\top]$. Also, notice that $\bar{P}_{xy,k}^\top = \bar{P}_{yx,k}$, $\bar{P}_{xz,k}^\top = \bar{P}_{zx,k}$, $\bar{P}_{yz,k}^\top = \bar{P}_{zy,k}$. Thus, it is straightforward to conclude that the covariance matrix can be formulated as

$$\begin{aligned}\hat{P}_{xx,k} = {} & \bar{P}_{xx,k} + K_k\bar{P}_{yy,k}K_k^\top + C_k\bar{P}_{zz,k}C^\top + K_k\bar{P}_{yz,k}C_k^\top + C_k\bar{P}_{yz,k}^\top K_k^\top \\ & - K_k\bar{P}_{xy,k}^\top - \bar{P}_{xy,k}K_k^\top - C_k\bar{P}_{xz,k}^\top - \bar{P}_{xz,k}C_k^\top\end{aligned} \tag{4.28}$$

The result about the gains update is stated in the following theorem.

Theorem 4.8 (Kalman and Communication Gains Update)**.** *Let (4.7) be the objective function of the distributed nonlinear Kalman filter. Let the posterior estimate be given by (4.23) and the covariance by (4.28). Then, the optimal communication and Kalman gains at time step $k \in \mathbb{Z}^*$ are, respectively, formulated by*

$$C_k = \bar{P}_{xz,k}\bar{P}_{zz,k}^{-1}, \tag{4.29}$$

$$K_k = (\bar{P}_{xy,k} - C_k\bar{P}_{yz,k}^{\top})\bar{P}_{yy,k}^{-1}. \tag{4.30}$$

Moreover, with these gains, the covariance in (4.28) can be expressed as

$$\hat{P}_{xx,k} = \bar{P}_{xx,k} - K_k\bar{P}_{yy,k}K_k^{\top} - C_k\bar{P}_{zz,k}C_k^{\top}. \tag{4.31}$$

Proof. Recall the cost function in (4.7). This cost function can also be written as

$$\mathbb{E}\{(x_k - \hat{x}_k)^{\top}(x_k - \hat{x}_k)\} = \mathbb{E}\{\mathrm{tr}\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)^{\top}\}\}, \tag{4.32}$$

in which $\mathbb{E}[\mathrm{tr}\{u\}] = \mathrm{tr}\{\mathbb{E}[u]\}$ and $\mathrm{tr}\{u^{\top}u\} = \mathrm{tr}\{uu^{\top}\}$ have been applied. Since minimising the mean-squared error (4.32) is equivalent to minimising the trace of the covariance of the estimate error, the optimal Kalman and communication gain can be derived by taking the derivative of the trace of posterior covariance matrix with respect to the Kalman and communication gain. Thus, taking the derivative of (4.28) with respect to the Kalman gain gives

$$\frac{\partial \mathrm{tr}(\hat{P}_{xx,k})}{\partial K_k} = K_k\bar{P}_{yy,k} + C_k\bar{P}_{yz,k}^{\top} - \bar{P}_{xy,k} = 0$$

which leads to

$$\bar{P}_{xy,k} = K_k\bar{P}_{yy,k} + C_k\bar{P}_{yz,k}^{\top}. \tag{4.33}$$

Plugging (4.33) into (4.28) yields

$$\hat{P}_{xx,k} = \bar{P}_{xx,k} - K_k\bar{P}_{yy,k}K_k^{\top} + C_k\bar{P}_{zz,k}C_k^{\top} - C_k\bar{P}_{xz,k}^{\top} - \bar{P}_{xz,k}C_k^{\top}. \tag{4.34}$$

Thus, taking the derivative of (4.34) with respect to C_k, i.e.,

$$\frac{\partial \mathrm{tr}(\hat{P}_{xx,k})}{\partial C_k} = C_k\bar{P}_{zz,k} - \bar{P}_{xz,k} = 0$$

results in the Communication gain given by

$$C_k = \bar{P}_{xz,k}\bar{P}_{zz,k}^{-1}. \tag{4.35}$$

Moreover, rearranging (4.35) to obtain $\bar{P}_{xz,k}$ and plugging it to (4.34) give the compact covariance expressed as

$$\hat{P}_{xx,k} = \bar{P}_{xx,k} - K_k\bar{P}_{yy,k}K_k^\top - C_k\bar{P}_{zz,k}C_k^\top. \tag{4.36}$$

This completes the proof. □

4.2.3 Instrumental Matrices

To analyse the performance of the estimation algorithm we have designed, some additional matrices are utilised – which has also been used in Kalman-filter based performance analysis such as in [114]. In the following analysis, $\bar{e}_k^x = x_k - \bar{x}_k$, $\hat{e}_k^x = x_k - \hat{x}_k$, $\bar{e}_k^y = y_k - \bar{y}_k$, and $\bar{e}_k^z = z_k - \bar{z}_k$ denote as the prior, posterior estimate, measurement and communication errors at time step $k \in \mathbb{Z}^*$, respectively.

Since x_k depends on the state estimate at $k-1$, by employing Taylor expansion to x_k in (4.4) around $\hat{x}_{k-1}$, we have

$$x_k = F(\hat{x}_{k-1}) + \nabla F(\hat{x}_{k-1})\hat{e}_{k-1}^x + \frac{1}{2}\nabla^2 F(\hat{x}_{k-1})(\hat{e}_{k-1}^x)^2 + \cdots + v_{k-1}, \tag{4.37}$$

where

$$\nabla^n F(\hat{x}_k)(\hat{e}_k^x)^n = \left(\sum_{j=1}^{P} \hat{e}_{k,j}^x \frac{\partial}{\partial x_j}\right)^n F(x)\bigg|_{x=\hat{x}_k},$$

for x_j and $\hat{e}_{k,j}^x$ referring to the j-th element of x and $\hat{e}_k^x$. Similarly, expanding $\bar{x}_k$ in (4.11) about $\hat{x}_{k-1}$ and applying Assumption 4.2 yields

$$\bar{x}_k = F(\hat{x}_{k-1}) + \frac{1}{2}\nabla^2 F(\hat{x}_{k-1})(\hat{e}_{k-1}^x)^2 + \cdots. \tag{4.38}$$

Thus, by subtracting (4.37) by (4.38), the estimate error caused by the prior estimate in (4.11) can be approximated by

$$\bar{e}_k^x \approx \hat{F}_{k-1}\hat{e}_{k-1}^x + v_{k-1}, \tag{4.39}$$

where

$$\hat{F}_k = \left(\frac{\partial F(x)}{\partial x}\bigg|_{x=\hat{x}_k}\right). \tag{4.40}$$

However, to accomodate the residuals emerging from this Taylor approximation, we utilise a diagonal matrix defined as $\alpha_k = \text{diag}(\alpha_{k,1},\ \alpha_{k,2}, \cdots,\ \alpha_{k,M})$, such that we have the approximation formula of error caused by the prior estimate in (4.11) at time step k given by

$$\bar{e}_k^x = \alpha_{k-1}\hat{F}_{k-1}\hat{e}_{k-1}^x + v_{k-1}. \tag{4.41}$$

For measurement approximation, since the measurement estimate is dependent on the prior estimate of x_k, the Taylor expansion of y_k in (4.5) is about $\bar{x}_k$. The Taylor expansion of the measurement dynamics at time step k is given by

$$y_k = H(\bar{x}_k) + \nabla H(\bar{x}_k)\bar{e}_k^x + \frac{1}{2}\nabla^2 H(\bar{x}_k)(\bar{e}_k^x)^2 + \cdots + w_k, \tag{4.42}$$

where

$$\nabla^n H(\bar{x}_k)(\bar{e}_k^x)^n = \left(\sum_{j=1}^{P} \bar{e}_{k,j}^x \frac{\partial}{\partial x_j}\right)^n H(x)\bigg|_{x=\bar{x}_k},$$

for x_j and $\bar{e}^{k,j}$ referring to the j-th element of x and $\bar{e}_k^x$. Subsequently, expanding the predicted measurement in (4.15) about $\bar{x}_k$ yields

$$\bar{y}_k = H(\bar{x}_k) + \frac{1}{2}\nabla^2 H(\bar{x}_k)(\bar{e}_k^x)^2 + \cdots. \tag{4.43}$$

Hence, subtracting (4.42) by (4.43) leads to

$$\bar{e}_k^y \approx \hat{H}_k \bar{e}_k^x + w_k, \tag{4.44}$$

where

$$\hat{H}_k = \left(\frac{\partial H(x)}{\partial x}\bigg|_{x=\bar{x}_k}\right). \tag{4.45}$$

An instrumental diagonal matrix related to the measurement prediction added to accomodate the residuals of this approximation is $\beta_k = \text{diag}(\beta_{k,1},\ \beta_{k,2}, \cdots,\ \beta_{k,M})$. Therefore, we have the error of the measurement prediction in (4.15) formulated as

$$\bar{e}_k^y = \beta_k \hat{H}_k \bar{e}_k^x + w_k. \tag{4.46}$$

In communication approximation, one may follow similar procedures to obtain the approximation of the measurement error. Since the communication estimate

is also dependent on the prior state estimate, the expansion of the communication data in (4.6) is also about $\bar{x}_k$. The Taylor expansion of the communication data at time step k is

$$z_k = G(\bar{x}_k) + \nabla G(\bar{x}_k)\bar{e}_k^x + \frac{1}{2}\nabla^2 G(\bar{x}_k)(\bar{e}_k^x)^2 + \cdots + s_k, \tag{4.47}$$

where

$$\nabla^n G(\bar{x}_k)(\bar{e}_k^x)^n = \left(\sum_{j=1}^{P} \bar{e}_{k,j}^x \frac{\partial}{\partial x_j}\right)^n G(x)\bigg|_{x=\bar{x}_k},$$

for x_j and $\bar{e}_{k,j}^x$ referring to the j-th element of x and $\bar{e}_k^x$. Subsequently, the Taylor expansion of (4.18) about $\bar{x}_k$ is

$$\bar{z}_k = G(\bar{x}_k) + \frac{1}{2}\nabla^2 G(\bar{x}_k)(\bar{e}_k^x)^2 + \cdots . \tag{4.48}$$

Hence, subtracting (4.47) by (4.48) yields the approximation error of the predicted communication data in (4.18) given by

$$\bar{e}_k^z \approx \hat{G}_k \bar{e}_k^x + s_k, \tag{4.49}$$

where

$$\hat{G}_k = \left(\frac{\partial G(x)}{\partial x}\bigg|_{x=\bar{x}_k}\right). \tag{4.50}$$

An instrumental diagonal matrix to deal with the residuals of the approximation of this communication data is $\gamma_k = \text{diag}(\gamma_{k,1},\ \gamma_{k,2}, \cdots,\ \gamma_{k,M})$. Hence, the approximate of the communication data satisfies the following equality:

$$\bar{e}_k^z = \gamma_k \hat{G}_k \bar{e}_k^x + s_k. \tag{4.51}$$

The unscented transformation of the covariance also requires us to design a approximation of the estimate covariance matrices. In the case of the prior estimate covariance, we insert (4.41) to $\bar{P}_{xx,k}^* = \mathbb{E}[\bar{e}_k^x(\bar{e}_k^x)^\top]$ and use Assumption 4.3 resulting in

$$\bar{P}_{xx,k}^* = \alpha_{k-1}\hat{F}_{k-1}\hat{P}_{xx,k-1}\hat{F}_{k-1}^\top\alpha_{k-1}^\top + Q_k + \delta_{xx,k-1} \tag{4.52}$$

as the approximation of the prior estimate covariance. Parameter $\delta_{xx,k}$ is to accomodate the residual approximation error of $\bar{P}_{xx,k}$ by $\bar{P}_{xx,k}^*$. For the error

covariance matrix of the measurement, with Assumption 4.3 and $\bar{e}_k^y$ from (4.46), the approximate value is given by $\bar{P}^*_{yy,k} = \mathbb{E}[\bar{e}_k^y(\bar{e}_k^y)^\top]$, i.e.,

$$\bar{P}^*_{yy,k} = \beta_k \hat{H}_k \bar{P}^*_{xx,k} \hat{H}_k^\top \beta_k^\top + R_k + \delta_{yy,k}, \tag{4.53}$$

where $\delta_{yy,k}$ denotes the approximate error caused by $\bar{P}^*_{yy,k}$. Subsequently, the error covariance matrix of the communication data, the approximation is given by $\bar{P}^*_{zz,k} = \mathbb{E}[\bar{e}_k^z(\bar{e}_k^z)^\top]$. Substituting (4.51) into it and using Assumption 4.3 yield

$$\bar{P}^*_{zz,k} = \gamma_k \hat{G}_k \bar{P}^*_{xx,k} \hat{G}_k^\top \gamma_k^\top + S_k + \delta_{zz,k}, \tag{4.54}$$

where in this case $\delta_{zz,k}$ is the approximate error of $\bar{P}^*_{zz,k}$.

There are three remaining important approximations that have to be defined prior to the performance analysis, i.e., the error covariance between the estimate and measurement error, between the estimate and communication error, and between the measurement and communication error. These values are $\bar{P}^*_{xy,k} = \mathbb{E}[\bar{e}_k^x(\bar{e}_k^y)^\top]$, $\bar{P}^*_{xz,k} = \mathbb{E}[\bar{e}_k^x(\bar{e}_k^z)^\top]$ and $\bar{P}^*_{yz,k} = \mathbb{E}[\bar{e}_k^y(\bar{e}_k^z)^\top]$. With (4.41), (4.46) and (4.51), we have

$$\bar{P}^*_{xy,k} = \bar{P}^*_{xx,k} \hat{H}_k^\top \beta_k^\top + \delta_{xy,k}, \tag{4.55}$$

$$\bar{P}^*_{xz,k} = \bar{P}^*_{xx,k} \hat{G}_k^\top \gamma_k^\top + \delta_{xz,k}, \tag{4.56}$$

and

$$\bar{P}^*_{yz,k} = \beta_k \hat{H}_k \bar{P}^*_{xx,k} \hat{G}_k^\top \gamma_k^\top + \delta_{yz,k}, \tag{4.57}$$

by denoting $\delta_{xy,k}$, $\delta_{xz,k}$ and $\delta_{yz,k}$ as the approximation errors caused by $\bar{P}^*_{xy,k}$, $\bar{P}^*_{xz,k}$ and $\bar{P}^*_{yz,k}$, respectively. Notice that, in above three equations, we have also applied the conditions in Assumption 4.3.

4.2.4 Performance Analysis

The following assumptions are made prior to analysing the performance of the proposed filter.

Assumption 4.9. *There exist non-zero real numbers* $\underline{\alpha}$, $\underline{\beta}$, $\underline{\gamma}$, $\underline{f}$, $\underline{h}$, $\underline{g}$, *and* $\overline{\alpha}$, $\overline{\beta}$, $\overline{\gamma}$, $\overline{f}$, $\overline{h}$, $\overline{g}$, *for every* $k > 0$, $k \in \mathbb{Z}^*$, *such that*

$$\underline{\alpha}^2 I \leqslant \alpha_k \alpha_k^\top \leqslant \overline{\alpha}^2 I, \tag{4.58}$$

$$\underline{\beta}^2 I \leqslant \beta_k \beta_k^\top \leqslant \overline{\beta}^2 I, \tag{4.59}$$

$$\underline{\gamma}^2 I \leqslant \gamma_k \gamma_k^\top \leqslant \overline{\gamma}^2 I, \tag{4.60}$$

$$\underline{f}^2 I \leqslant \hat{F}_k \hat{F}_k^\top \leqslant \overline{f}^2 I, \tag{4.61}$$

$$\underline{h}^2 I \leqslant \hat{H}_k \hat{H}_k^\top \leqslant \overline{h}^2 I. \tag{4.62}$$

$$\underline{g}^2 I \leqslant \hat{G}_k \hat{G}_k^\top \leqslant \overline{g}^2 I. \tag{4.63}$$

Assumption 4.10. *There exist positive real numbers* $\underline{p}$, $\underline{q}$, $\underline{r}$, $\underline{s}$, $\underline{\delta}_{xx}$, $\underline{\delta}_{yy}$, $\underline{\delta}_{zz}$, $\underline{\delta}_{xy}$, $\underline{\delta}_{xz}$, $\underline{\delta}_{yz}$, *and* $\overline{p}$, $\overline{q}$, $\overline{r}$, $\overline{s}$, $\overline{\delta}_{xx}$, $\overline{\delta}_{yy}$, $\overline{\delta}_{zz}$, $\overline{\delta}_{xy}$, $\overline{\delta}_{xz}$, $\overline{\delta}_{yz}$, *for every* $k > 0$, $k \in \mathbb{Z}^*$, *such that*

$$\underline{p} I \leqslant \hat{P}_{xx,k} \leqslant \overline{p} I, \tag{4.64}$$

$$\underline{q} I \leqslant Q_k \leqslant \overline{q} I, \tag{4.65}$$

$$\underline{r} I \leqslant R_k \leqslant \overline{r} I, \tag{4.66}$$

$$\underline{s} I \leqslant S_k \leqslant \overline{s} I, \tag{4.67}$$

$$\underline{\delta}_{xx} I \leqslant \delta_{xx,k} \leqslant \overline{\delta}_{xx} I, \tag{4.68}$$

$$\underline{\delta}_{yy} I \leqslant \delta_{yy,k} \leqslant \overline{\delta}_{yy} I, \tag{4.69}$$

$$\underline{\delta}_{zz} I \leqslant \delta_{zz,k} \leqslant \overline{\delta}_{zz} I, \tag{4.70}$$

$$\underline{\delta}_{xy} I \leqslant \delta_{xy,k} \leqslant \overline{\delta}_{xy} I, \tag{4.71}$$

$$\underline{\delta}_{xz} I \leqslant \delta_{xz,k} \leqslant \overline{\delta}_{xz} I, \tag{4.72}$$

$$\underline{\delta}_{yz} I \leqslant \delta_{yz,k} \leqslant \overline{\delta}_{yz} I, \tag{4.73}$$

Some lemmas stating the upper and lower boundaries of some parameters are declared as follows.

Lemma 4.11. *Let the conditions in Assumptions 4.9 and 4.10 holds. Let the error of the prior estimates of the nonlinear Kalman filter be approximated using (4.41), (4.46) and (4.51). With conditions in Assumptions 4.9 and 4.10 hold, the prior covariance matrices are bounded by*

$$\underline{\alpha}^2\underline{f}^2\underline{p}+\underline{q}+\underline{\delta}_{xx} \leqslant \bar{P}^*_{xx,k} \leqslant \overline{\alpha}^2\overline{f}^2\overline{p}+\overline{q}+\overline{\delta}_{xx}, \tag{4.74}$$

$$\underline{\beta}^2\underline{h}^2(\underline{\alpha}^2\underline{f}^2\underline{p}+\underline{q}+\underline{\delta}_{xx})+\underline{r}+\underline{\delta}_{yy} \leqslant \bar{P}^*_{yy,k} \leqslant \overline{\beta}^2\overline{h}^2(\overline{\alpha}^2\overline{f}^2\overline{p}+\overline{q}+\overline{\delta}_{xx})+\overline{r}+\overline{\delta}_{yy}, \tag{4.75}$$

$$\underline{\gamma}^2\underline{g}^2(\underline{\alpha}^2\underline{f}^2\underline{p}+\underline{q}+\underline{\delta}_{xx})+\underline{s}+\underline{\delta}_{zz} \leqslant \bar{P}^*_{zz,k} \leqslant \overline{\gamma}^2\overline{g}^2(\overline{\alpha}^2\overline{f}^2\overline{p}+\overline{q}+\overline{\delta}_{xx})+\overline{s}+\overline{\delta}_{zz}, \tag{4.76}$$

$$(\underline{\alpha}^2\underline{f}^2\underline{p}+\underline{q}+\underline{\delta}_{xx})\hat{H}_k^\top\beta^\top+\underline{\delta}_{xy} \leqslant \bar{P}^*_{xy,k} \leqslant (\overline{\alpha}^2\overline{f}^2\overline{p}+\overline{q}+\overline{\delta}_{xx})\hat{H}_k^\top\beta^\top+\overline{\delta}_{xy}, \tag{4.77}$$

$$(\underline{\alpha}^2\underline{f}^2\underline{p}+\underline{q}+\underline{\delta}_{xx})\hat{G}_k^\top\gamma^\top+\underline{\delta}_{xz} \leqslant \bar{P}^*_{xz,k} \leqslant (\overline{\alpha}^2\overline{f}^2\overline{p}+\overline{q}+\overline{\delta}_{xx})\hat{G}_k^\top\gamma^\top+\overline{\delta}_{xz}, \tag{4.78}$$

$$(\underline{\alpha}^2\underline{f}^2\underline{p}+\underline{q}+\underline{\delta}_{xx})\beta\hat{H}_k\hat{G}_k^\top\gamma^\top+\underline{\delta}_{yz} \leqslant \bar{P}^*_{yz,k} \leqslant (\overline{\alpha}^2\overline{f}^2\overline{p}+\overline{q}+\overline{\delta}_{xx})\beta\hat{H}_k\hat{G}_k^\top\gamma^\top+\overline{\delta}_{yz}. \tag{4.79}$$

Lemma 4.12. *Let the error of the prior estimates of the nonlinear Kalman filter be approximated using (4.41), (4.46) and (4.51); and the prior error covariances bounded by (4.74)-(4.79). Then, with the conditions in Assumptions 4.9 and 4.10 hold, the square of communication gain is bounded by*

$$\underline{c}^2 I \leqslant C_k C_k^\top \leqslant \overline{c}^2 I \tag{4.80}$$

where

$$\underline{c}^2 = \frac{\underline{\gamma}^2\underline{g}^2(\underline{\alpha}^2\underline{f}^2\underline{p}+\underline{q}+\underline{\delta}_{xx})^2+\underline{\delta}_{xz}^2}{(\overline{\gamma}^2\overline{g}^2(\overline{\alpha}^2\overline{f}^2\overline{p}+\overline{q}+\overline{\delta}_{xx})+\overline{s}+\overline{\delta}_{zz})^2} \tag{4.81}$$

and

$$\overline{c}^2 = \frac{2\left(\overline{\gamma}^2\overline{g}^2(\overline{\alpha}^2\overline{f}^2\overline{p}+\overline{q}+\overline{\delta}_{xx})^2+\overline{\delta}_{xz}^2\right)}{(\underline{\gamma}^2\underline{g}^2(\underline{\alpha}^2\underline{f}^2\underline{p}+\underline{q}+\underline{\delta}_{xx})+\underline{r}+\underline{\delta}_{zz})^2}. \tag{4.82}$$

Lemma 4.13. *Let the error of the prior estimates of the nonlinear Kalman filter be approximated using (4.41), (4.46) and (4.51); and the prior error covariances bounded by (4.74)-(4.79). Then, with the conditions in Assumptions 4.9 and 4.10 hold, the square of Kalman gain is bounded by*

$$\underline{k}^2 I \leqslant K_k K_k^\top \leqslant \overline{k}^2 I \tag{4.83}$$

where

$$\underline{k}^2 = 0 \tag{4.84}$$

and

$$\overline{k}^2 = \frac{4\left((\overline{\beta}^2\overline{h}^2 + \overline{c}^2\overline{\beta}^2\overline{\gamma}^2\overline{h}^2\overline{g}^2)(\overline{\alpha}^2\overline{f}^2\overline{p} + \overline{q} + \overline{\delta}_{xx})^2 + \overline{\delta}_{xy}^2 + \overline{\delta}_{yz}^2\right)}{(\underline{\beta}^2\underline{h}^2(\underline{\alpha}^2\underline{f}^2\underline{p} + \underline{q} + \underline{\delta}_{xx}) + \underline{r} + \underline{\delta}_{yy})^2}, \tag{4.85}$$

with $\overline{c}^2$ as the upper bound of the square of communication gain.

Lemma 4.14. *Let K_k in (4.29) and C_k in (4.30) be bounded by (4.83) and (4.80), respectively. Define $M_k = I - K_k\beta_k\hat{H}_k - C_k\gamma_k\hat{G}_k$. Then, with the conditions in Assumptions 4.9 and 4.10 satisfied, the boundaries of $M_kM_k^\top$ are given by*

$$\underline{m}^2 I \leqslant M_kM_k^\top \leqslant \overline{m}^2 I \tag{4.86}$$

where

$$\underline{m}^2 = 0, \tag{4.87}$$

and

$$\overline{m}^2 = 3\left(1 + \overline{k}^2\overline{\beta}^2\overline{h}^2 + \overline{c}^2\overline{\gamma}^2\overline{g}^2\right). \tag{4.88}$$

Lemma 4.15. *Consider the posterior estimate and covariance update of the distributed nonlinear Kalman filter formulated by (4.23) and (4.31), respectively. Define a matrix $M_k = (I - K_k\beta_k\hat{H}_k - C_k\gamma_k\hat{G}_k)$. If the conditions in Assumptions 4.9 and 4.10 hold, then, at every time step $k \in \mathbb{Z}^*$, the inverse of the covariance, denoted by $\Pi_k = P_k^{-1}$, satisfies the following condition*

$$\Pi_k \leqslant \varphi\Pi_{k-1}, \tag{4.89}$$

where

$$\varphi = \frac{1}{\underline{\alpha}^2\underline{f}^2\underline{m}^2}\left(1 + \frac{\underline{m}^2\underline{q} + \underline{k}^2\underline{r} + \underline{c}^2\underline{s} + \underline{m}^2\underline{\delta}_{xx}}{\overline{\alpha}^2\overline{f}^2\overline{m}^2\overline{p}}\right)^{-1}. \tag{4.90}$$

Proof. The error of the posterior estimate is $\hat{e}_k^x = x_k - \hat{x}_k$ which, by plugging (4.23) into it, gives

$$\hat{e}_k^x = \bar{e}_k^x - K_k\bar{e}_k^y - C_k\bar{e}_k^z \tag{4.91}$$

where $\bar{e}_k^x = x_k - \bar{x}_k$, $\bar{e}_k^y = y_k - \bar{y}_k$ and $\bar{e}_k^z = z_k - \bar{z}_k$. The measurement and communication errors can be substituted by the approximate values in (4.46) and (4.51), respectively. From this substitution, we have

$$\hat{e}_k^x = M_k\bar{e}_k^x - K_kw_k - C_ks_k \tag{4.92}$$

where $M_k = (I - K_k\beta_k\hat{H}_k - C_k\gamma_k\hat{G}_k)$. Recall the definition of posterior covariance matrix $\hat{P}_{xx,k} = \mathbb{E}[\hat{e}_k^x(\hat{e}_k^x)^\top]$. Plugging (4.92) into it gives

$$\hat{P}_{xx,k} = M_k\bar{P}_{xx,k}M_k^\top + K_kR_kK_k^\top + C_kS_kC_k^\top \tag{4.93}$$

in which Assumptions 4.3 and 4.4 have been applied. Notice that (4.93) uses the approximate value of $\bar{P}_{xx,k}$. Using (4.52), one may have (4.93) expanded to

$$\hat{P}_{xx,k} = M_k\left(\alpha_{k-1}\hat{F}_{k-1}\hat{P}_{xx,k-1}(\alpha_{k-1}\hat{F}_{k-1})^\top + Q_k + \delta_{xx,k-1}\right)M_k^\top + K_kR_kK_k^\top + C_kS_kC_k^\top. \tag{4.94}$$

Rearranging the terms, we have

$$\begin{aligned}\hat{P}_{xx,k} =& M_k\alpha_{k-1}\hat{F}_{k-1}\Big(I + (M_k\alpha_{k-1}\hat{F}_{k-1})^{-1}K_kR_kK_k^\top(M_k\alpha_{k-1}\hat{F}_{k-1})^{-\top}(\hat{P}_{xx,k-1})^{-1}\\ &+(M_k\alpha_{k-1}\hat{F}_{k-1})^{-1}C_kS_kC_k^\top(M_k\alpha_{k-1}\hat{F}_{k-1})^{-\top}\hat{P}_{xx,k-1}^{-1}\\ &+(\alpha_{k-1}\hat{F}_{k-1})^{-1}Q_k(\alpha_{k-1}\hat{F}_{k-1})^{-\top}\hat{P}_{xx,k-1}^{-1}\\ &+(\alpha_{k-1}\hat{F}_{k-1})^{-1}\delta_{xx,k-1}(\alpha_{k-1}\hat{F}_{k-1})^{-\top}\hat{P}_{xx,k-1}^{-1}\Big)\hat{P}_{xx,k-1}(M_k\alpha_{k-1}\hat{F}_{k-1})^\top.\end{aligned} \tag{4.95}$$

Using Assumptions 4.9 and 4.10 along with Lemmas 4.12, 4.13 and 4.14, (4.95) can be expressed as

$$\hat{P}_{xx,k} \geqslant \underline{\alpha}^2\underline{f}^2\underline{m}^2\left(1 + \frac{\underline{m}^2\underline{q} + \underline{k}^2\underline{r} + \underline{c}^2\underline{s} + \underline{m}^2\underline{\delta}_{xx}}{\overline{\alpha}^2\overline{f}^2\overline{m}^2\overline{p}}\right)\hat{P}_{xx,k-1}. \tag{4.96}$$

Therefore, with $\Pi_k = \hat{P}_{xx,k}^{-1}$, taking the inverse of (4.96) results in the inverse covariance matrix bounded by

$$\Pi_k \leqslant \varphi\Pi_{k-1}, \tag{4.97}$$

where

$$\varphi = \frac{1}{\underline{\alpha}^2\underline{f}^2\underline{m}^2}\left(1 + \frac{\underline{m}^2\underline{q} + \underline{k}^2\underline{r} + \underline{c}^2\underline{s} + \underline{m}^2\underline{\delta}_{xx}}{\overline{\alpha}^2\overline{f}^2\overline{m}^2\overline{p}}\right)^{-1}. \tag{4.98}$$

This completes the proof. □

The following theorem guarantees the boundedness of the proposed distribued nonlinear Kalman filter.

Theorem 4.16 (Stochastic Stabilty of Distributed Nonlinear Kalman Filter)**.** *Let the posterior estimator of the distributed nonlinear Kalman filter be given by (4.23), and the covariance by (4.31). Suppose that Assumptions 4.2 to 4.10 hold. Then, there exists a stochastic function $V \geqslant 0$ such that the estimate error is bounded in mean square exponentially.*

Proof. To proof the theorem, first, let us choose a stochastic function candidate

$$V(\hat{e}_k^x) = \hat{e}_x^\top \Pi_k \hat{e}_k^x \tag{4.99}$$

where $\hat{e}_k^x = x_k - \hat{x}_k$ and $\Pi_k = \hat{P}_{xx,k}^{-1}$. To show the boundedness of V, multiplying the inequality in the condition (4.64) by $(\hat{e}_k^x)^\top \hat{e}_k^x$ results in

$$\frac{1}{\overline{p}}\|\hat{e}_k^x\|^2 \leqslant V(\hat{e}_k^x) \leqslant \frac{1}{\underline{p}}\|\hat{e}_k^x\|^2 \tag{4.100}$$

$$\underline{v}\|\hat{e}_k^x\|^2 \leqslant V(\hat{e}_k^x) \leqslant \overline{v}\|\hat{e}_k^x\|^2. \tag{4.101}$$

Therefore, since $\underline{p}$ and $\overline{p}$ are positive, V is positively bounded with $\underline{v}$, $\overline{v} > 0$. Since V is positively bounded, it can now be used to show that there exist $\mu > 0$ and $0 < \sigma \leqslant 1$ such that the following inequality holds

$$\mathbb{E}[V(\hat{e}_{k+1}^x)|\hat{e}_k^x] \leqslant \mu + (1-\sigma)V(\hat{e}_k^x). \tag{4.102}$$

Let us consider the stochastic function (4.101) at time step $k+1$, i.e.,

$$V(\hat{e}_{k+1}^x) = (\hat{e}_{k+1}^x)^\top \Pi_{k+1} \hat{e}_{k+1}^x. \tag{4.103}$$

Using the definition of the errors for the estimates, i.e., $\bar{e}_{k+1}^x = x_{k+1} - \bar{x}_{k+1}$, $\bar{e}_{k+1}^y = y_{k+1} - \bar{y}_{k+1}$, and $\bar{e}_{k+1}^z = x_{k+1} - \bar{z}_{k+1}$, and plugging them into the posterior estimate (4.23) at time $k+1$ yields the posterior error $\hat{e}_{k+1}^x = x_{k+1} - \hat{x}_{k+1}$ given by

$$\hat{e}_{k+1}^x = \bar{e}_{k+1}^x - K_{k+1}\bar{e}_{k+1}^y - C\bar{e}_{k+1}^z. \tag{4.104}$$

By inserting (4.41), (4.46) and (4.51) at $k+1$ to (4.104), we have

$$\hat{e}_{k+1}^x = M_{k+1}\alpha_k \hat{F}_k \hat{e}_k^x + M_{k+1}v_k - K_{k+1}w_{k+1} - C_{k+1}s_{k+1} \tag{4.105}$$

where $M_{k+1} = (I - K_{k+1}\beta_{k+1}\hat{H}_{k+1} - C_{k+1}\gamma_{k+1}\hat{G}_{k+1})$.

Plugging (4.105) into (4.103) and taking its expected value with respect to $\hat{e}_k^x$ lead to

$$\begin{aligned}\mathbb{E}[V(\hat{e}_{k+1}^x)|\hat{e}_k^x] = \mathbb{E}[&(M_{k+1}\alpha_k\hat{F}_k\hat{e}_k^x)^\top\Pi_{k+1}M_{k+1}\alpha_k\hat{F}_k\hat{e}_k^x + (M_{k+1}v_k)^\top\Pi_{k+1}M_{k+1}v_k \\ &+ (K_{k+1}w_{k+1})^\top\Pi_{k+1}K_{k+1}w_{k+1} + (C_{k+1}s_{k+1})^\top\Pi_{k+1}C_{k+1}s_{k+1}|\hat{e}_k^x]\end{aligned} \tag{4.106}$$

where Assumptions 4.4 and 4.5 have been utilised. With Assumptions 4.9, applying Lemma 4.14 and Lemma 4.15 to the first term of (4.106) leads to

$$\begin{aligned}\mathbb{E}[(M_{k+1}\alpha_k\hat{F}_k\hat{e}_k^x)^\top\Pi_{k+1}M_{k+1}\alpha_k\hat{F}_k\hat{e}_k^x|\hat{e}_k^x] &\leqslant \mathbb{E}[\mathrm{tr}\left(\overline{\alpha}^2\overline{f}^2\overline{m}^2(\hat{e}_k^x)^\top\Pi_{k+1}\hat{e}_k^x\right)|\hat{e}_k^x] \\ &\leqslant \overline{\alpha}^2\overline{f}^2\overline{m}^2\varphi V(\hat{e}_k^x)\end{aligned} \tag{4.107}$$

in which we have applied the properties of matrix trace $\mathrm{tr}(\mathbb{E}[A]) = \mathbb{E}[\mathrm{tr}(A)]$ and $\mathrm{tr}(AB) = \mathrm{tr}(BA)$. Similarly, employing Assumptions 4.2, 4.9, and 4.10, along with Lemmas 4.12, 4.13, and 4.14 to the second, third and fourth terms of (4.107) give

$$\begin{aligned}&\mathbb{E}[(M_{k+1}v_k)^\top\Pi_{k+1}M_{k+1}v_k + (K_{k+1}w_{k+1})^\top\Pi_{k+1}K_{k+1}w_{k+1} + (C_{k+1}s_{k+1})^\top\Pi_{k+1}C_{k+1}s_{k+1}|\hat{e}_k^x] \\ &\qquad\leqslant \frac{1}{\underline{p}}\left(\overline{m}^2\overline{q} + \overline{k}^2\overline{r} + \overline{c}^2\overline{s}\right)\end{aligned} \tag{4.108}$$

where the properties of a trace of matrix have also been used. By denoting

$$\mu = \frac{1}{\underline{p}}\left(\overline{m}^2\overline{q} + \overline{k}^2\overline{r} + \overline{c}^2\overline{s}\right), \tag{4.109}$$

$$\sigma = 1 - \overline{\alpha}^2\overline{f}^2\overline{m}^2\varphi, \tag{4.110}$$

and choosing some constants such that $0 \leqslant \sigma \leqslant 1$ and $\mu \geqslant 0$, we are now able to write (4.108) as

$$\mathbb{E}[V(\hat{e}_{k+1}^x)|\hat{e}_k^x] \leqslant \mu + (1-\sigma)V(\hat{e}_k^x). \tag{4.111}$$

Therefore, by employing Lemma 4.6, we can conclude that $\hat{e}_k^x$ is exponentially bounded in mean square with

$$\mathbb{E}[\|\hat{e}_k^x\|^2] \leqslant \frac{\overline{v}}{\underline{v}}\mathbb{E}[\|\hat{e}_0^x\|^2](1-\sigma)^k + \frac{\mu}{\underline{v}}\sum_{l=1}^{k-1}(1-\sigma)^l. \tag{4.112}$$

This completes the proof. □

Algorithm 4.1 *Distributed Nonlinear Kalman Filter*

Input: $x(0)$, $Q(0), R(0), S(0)$, $F(\cdot)$, $H(\cdot)$, $G(\cdot)$.
Return: $\hat{x}_k$, $\hat{P}_{xx,k}$.
If $k = 0$, execute
Initialisation:

1: Initialise the estimator $\hat{x}_k$ and covariance P:

$$\tilde{x} = [\hat{x}_k(0)^\top \ 0_p \ 0_q \ 0_r]^\top, \text{ for } \hat{x}_k(0) = \mathbb{E}[x(0)],$$
$$\tilde{P} = \text{diag}(P(0), Q(0), R(0), S(0)), \text{ for } P(0) = \mathbb{E}[(x(0) - \hat{x}_k(0))(x(0) - \hat{x}_k(0))^\top].$$

If $k = 1, 2, \ldots, T_{\max}, T_{\max} < \infty$, execute
Prediction:

1: Calculate the sigma vector $\mathcal{X}_{k-1}$ using Eq. (4.9).
2: Propagate the sigma vector through $F(\cdot)$, $H(\cdot)$, $G(\cdot)$ using (4.10), (4.14) and (4.17), respectively.
3: Calculate the prior estimate $\bar{x}_k$ using (4.11), $\bar{y}_k$ using (4.15), and $\bar{z}_k$ using (4.18).
4: Calculate the prior covariances $\bar{P}_{xx,k}$, $\bar{P}_{yy,k}$, $\bar{P}_{zz,k}$, $\bar{P}_{xy,k}$, $\bar{P}_{xz,k}$, and $\bar{P}_{yz,k}$ using Eqs. (4.12), (4.16), (4.19), (4.20), (4.21), and (4.22), respectively.

Update:

1: Update the Kalman and communication gains, K_k and C_k using Eqs. (4.29) and (4.30), respectively.
2: Update the posterior estimate $\hat{x}_k$ using Eq. (4.23).
3: Update the posterior covariance $\hat{P}_{xx,k}$ using Eq. (4.31)).

4.2.5 Practical Algorithm

This proposed algorithm can generally be applied to solve many estimation problems in, but not limited to, engineering field via distributed computation of the filter whenever the dynamical process of a system, measurement and communication update can be formulated as in the discussion. For example, this algorithm can further be employed to estimate the Bellman value and Q function in reinforcement learning, such as solving the learning problem in [71] in a distributed manner using our proposed filtering algorithm. In the case of a sensor network, this algorithm can also be employed to estimate the spatial information of an area, which will be discussed in the next section.

The procedure of implementing the proposed distributed nonlinear Kalman filter is summarised in Algorithm 4.1. In the beginning, to implement the algorithm in a distributed manner, one should supply some initial values of the augmented state variable and the augmented covariance matrix in each agent. Afterwards,

once the system starts, in every iteration, each agent will locally perform the prediction step based on the latest available estimate value and its covariances using the unscented transformation technique. This prediction step produces the prior estimate values and their covariance and cross-covariance matrices. The final stage of each iteration is called the update stage. In this stage, each agent will have to exchange its local prior estimate values and their covariance matrices to its neighbouring agents based on the network topology. This step is followed by calculating the new posterior estimate value and its covariance matrix. These prediction and update procedures repeat iteratively.

4.3 Application to Distributed Coverage Control

In this section, we implement the proposed distributed nonlinear Kalman filter algorithm to specifically solve an existing problem in sensor network. The problem considered in this section is the field estimation of coverage control problem. This part is extended from our previous work in distributed coverage control elaborated in Chapter 3 and published in [115].

4.3.1 Modified Coverage Problem

In this part, we recall the coverage problem discussed in 3 and modify the objective function. Let the augmented vector of the position of robots be defined as $p_k = \text{vec}_n(p_k^i)^\top \in \mathbb{R}^{nd}$, for $i \in \mathcal{V}_n$, and the augmented Laplacian matrix of the Delaunay graph as $\hat{\mathcal{L}}_n = \mathcal{L}_n \otimes I_d \in \mathbb{R}^{nd\times nd}$. From our previous result in Chapter 3, the distributed expression of cost function of coverage problem can be formulated as

$$\min_{p_k^i} \quad \mathcal{H}(\mathcal{P}) = \sum_{i=1}^{n} \int_{V_i} \rho(\|q - p_k^i\|)\phi(q)dq, \tag{4.113a}$$

$$\text{s.t.} \quad \hat{\mathcal{L}}_n(p_k - C_V) = 0, \tag{4.113b}$$

with

$$V_i(p_i) = \{q \in \mathcal{Q} : \|q - p_i\| \leqslant \|q - p_j\|, \forall p_j \in \mathcal{P}, j \neq i\},$$

and $C_V = \text{vec}_n(C_{V_i}) \in \mathbb{R}^{nd}$, for $i \in \mathcal{V}_n$.

Subsequently, we augment the constraint to the objective function using the Lagrange multiplier method. The Lagrangian of this coverage problem, $\tilde{\mathcal{H}} : \mathbb{R}^{nd} \times \mathbb{R}^{nd} \to \mathbb{R}$, with disagreement function of consensus protocol is

$$\tilde{\mathcal{H}}(p_k, \nu_k) = \mathcal{H} + \nu_k^\top \hat{\mathcal{L}}_n(p_k - C_V) + \frac{1}{2}(p_k - C_V)^\top \hat{\mathcal{L}}_n(p_k - C_V), \tag{4.114}$$

where $\nu_k \in \mathbb{R}^{nd}$ defines the Lagrangian multiplier of the constraint. Thus, the control input and Lagrangian multiplier update, which is derived by taking the gradient of $\tilde{\mathcal{H}}(p_k, \nu_k)$ with respect to p_k and ν_k, respectively, can be expressed as

$$u_k = -2\kappa_1 M_V(p_k - C_V) - \kappa_2 \hat{\mathcal{L}}_n(p_k - C_V) - \hat{\mathcal{L}}_n \nu_k, \tag{4.115a}$$

$$\dot{\nu}_k = \kappa_3 \hat{\mathcal{L}}_n(p_k - C_V), \tag{4.115b}$$

where κ_1, κ_2, κ_3, $\in \mathbb{R}_+$.

To apply the estimation update from the online field estimator, the centroid C_{V_i} in the control input is replaced by the estimated centroid $\hat{C}_{V_i}$ such that the control input of agent i is given by

$$u_k = -k_p(p_k - \hat{C}_V) - k_c \hat{\mathcal{L}}_n(p_k - \hat{C}_V) - k_i \hat{\mathcal{L}}_n \int_{t_a}^{t_b} (p_k - \hat{C}_V) dt. \tag{4.116}$$

Notice that, in the above expression, we have simplified the gains expression by choosing k_p, k_c, $k_i \in \mathbb{R}_+$ to be constant gains.

4.3.2 Field Estimation

In this estimation scenario, we consider the information distribution of an area can be expressed as

$$\phi_k(q) = \mathcal{K}(q)^\top \theta_k, \tag{4.117}$$

with $\mathcal{K} : \mathcal{Q} \to \mathbb{R}^m$ denoting a vector of basis function that is known by each robot [21] [116]. Weight $\theta_k \in \mathbb{R}^m$ here denotes an unknown parameter to estimate. From (3.4), it can be seen that the mass M_{V_i} of the centroid C_{V_i} must be greater than zero. This situation requires the information distribution to be positive, $\phi_k(q) > 0$. Therefore, a boundary must be set, i.e.,

$$\theta_{k,l} \geqslant \underline{\theta}, \quad \text{for } l \in \{1, 2, \ldots, m\}, \tag{4.118}$$

with $\underline{\theta} > 0$ denoting the lower bound of l-th entry of θ_k.

Afterwards, several estimation variables are defined. $\hat{\theta}_k^i$ and $\hat{\phi}_k^i$ denote, respectively, the estimate weight and information distribution of robot i. Let $\hat{\phi}_k^i = \mathcal{K}^\top \hat{\theta}_k^i$ denote the estimate of ϕ of i-th robot. Then, we may have the estimate of mass, moment of inertia and centroid of Voronoi region formulated as

$$\hat{M}_{V_i} = \int_{V_i} \hat{\phi}_k^i(q)dq,\ \hat{\mathcal{I}}_{V_i} = \int_{V_i} q\hat{\phi}_k^i(q)dq,\ \text{and}\ \hat{C}_{V_i} = \frac{\hat{\mathcal{I}}_{V_i}}{\hat{M}_{V_i}}, \tag{4.119}$$

respectively.

By letting ϕ_k^i be the measurement value read by sensor i, the estimation error of θ_k and ϕ_k can, respectively, be expressed as $\tilde{\theta}_k^i = \theta_k - \hat{\theta}_k^i$ and $\tilde{\phi}_k^i = \phi_k^i - \hat{\phi}_k^i = \mathcal{K}(q)^\top \tilde{\theta}_k^i$. With these notations, we have the error between true and estimate value of the rigid body properties in the form of

$$\tilde{M}_{V_i} = \int_{V_i} \mathcal{K}^\top(q)\tilde{\theta}_k^i dq,\ \tilde{\mathcal{I}}_{V_i} = \int_{V_i} q\mathcal{K}^\top(q)\tilde{\theta}_k^i dq, \tag{4.120}$$

and

$$\tilde{C}_{V_i} = \frac{\tilde{\mathcal{I}}_{V_i}}{\tilde{M}_{V_i}}. \tag{4.121}$$

Therefore, the distributed field estimation of the coverage control problem is solved if the following conditions are satisfied:

1. $\lim_{k\to\infty} \|p_k^i - \hat{C}_{V_i}\| = 0,\ \text{for}\ i \in \mathcal{V}_n,$
2. $\lim_{k\to\infty} \mathcal{K}^\top \tilde{\theta}_k^i = 0,\ \text{for}\ i \in \mathcal{V}_n.$

It can also be concluded that if $\hat{\phi}_k^i = \phi_k$, then $\hat{M}_{V_i} = M_{V_i}$, $\hat{\mathcal{I}}_{V_i} = \mathcal{I}_{V_i}$ and $\hat{C}_{V_i} = C_{V_i}$.

4.3.3 Distributed Estimation with Laplacian-based Graph

To obtain the estimation of the centroid required in the control law (4.116), the distributed nonlinear Kalman filter we have designed in Section 4.2 is utilised.

Consider the estimation dynamics, consisting of the weight of the density function, sensor reading and communication dynamics which, in augmented expression,

are given by

$$\theta_{k+1} = \theta_k + v_k, \tag{4.122}$$

$$\psi_k = \phi(q, \theta_k) + w_k, \tag{4.123}$$

$$\tau_k = \hat{\mathcal{L}}_n \theta + s_k, \tag{4.124}$$

respectively. Note that $\theta_k = \text{vec}_n(\theta_k^i)$, $\psi_k = \text{vec}_n(\psi_k^i)$, $\tau_k = \text{vec}_n(\tau_k^i)$, $v_k = \text{vec}_n(v_k^i)$, $w_k = \text{vec}_n(w_k^i)$, $s_k = \text{vec}_n(s_k^i)$ and $\hat{\mathcal{L}}_n = \mathcal{L}_n \otimes I_d$.

Since in this application the network topology is modelled using the Laplacian matrix, the posterior estimator in (4.23) may be rewritten as

$$\hat{x}_k = \bar{x}_k + K_k(y_k - \bar{y}_k) - C_k \hat{\mathcal{L}}_n(\bar{x}_k - x_k). \tag{4.125}$$

Let the element-wise expression of (4.125) be given by

$$\hat{x}_k^i = \bar{x}_k^i + K_k^i(y_k^i - \bar{y}_k^i) + C_k^i \sum_{j=1}^{n} a_{ij} \left((\bar{x}_k^j - x_k^j) - (\bar{x}_k^i - x_k^i) \right), \text{ for } i \in \mathcal{V}_n. \tag{4.126}$$

Since each agent is assigned to track the same dynamical process, we have $x_k^i = x_k^j = x_k$. Thus, the posterior estimator of the Laplacian-based communication topology can be reduced to

$$\hat{x}_k^i = \bar{x}_k^i + K_k^i(y_k^i - \bar{y}_k^i) + C_k^i \sum_{j=1}^{n} a_{ij} \left(\bar{x}_k^j - \bar{x}_k^i \right), \text{ for } i \in \mathcal{V}_n. \tag{4.127}$$

The coverage control procedure with the Kalman-based estimator is summarised in Algorithm 4.2.

Algorithm 4.2 Distributed Coverage Control with Field Estimation

Given information: $\hat{\theta}(0)$, $P(0)$, $Q(0)$, $R(0)$, $S(0)$.
Return: p_k.
If $k = 0$, execute
Initialisation:

1: Initialise the estimator $\hat{x}_k$ and covariance $\hat{P}_{xx,k}$:

$$\tilde{\theta}_0 = [\hat{\theta}(0)^\top \ 0_p \ 0_q \ 0_r]^\top, \text{ for } \hat{\theta}(0) = \mathbb{E}[\theta(0)],$$
$$\tilde{P}_0 = \text{diag}(P(0), Q(0), R(0), S(0)), \text{ for } P(0) = \mathbb{E}[(\theta(0) - \hat{\theta}(0))(\theta(0) - \hat{\theta}(0))^\top].$$

2: Calculate Voronoi partition V.

If $k = 1, 2, \ldots, T_k, T_k < \infty$, execute
Control Update:

1: Calculate the new centroid using Eq. (4.119).
2: Calculate the control input using Eq. (4.116).
3: Update the position of robot

$$p_{k+1} = g(p_k, u_k).$$

4: Update the Voronoi partition V.

Estimation update:

1: With $\hat{x}_k = \hat{\theta}_k$, execute estimation procedures in Algorithm 4.1.
2: Assign $\hat{\theta}_k = \hat{x}_k$.

4.4 Numerical Experiments

For comparison, two simulation scenarios are presented to estimate the spatial information distribution of an area and drive a group of robots to estimate the optimal configurations using two different estimation algorithms. The first scenario, referred to as the modified-consensus observer algorithm, illustrates a drawback arising from using the modified consensus observer in [21]. The second scenario, referred to as the distributed Kalman filter algorithm, demonstrates the performance of the proposed distributed nonlinear Kalman filter given in Algorithm 4.2. The simulations are performed using Python programming language on a computer with a Linux-based operating system, 2.5-GHz processor, and 4-GB RAM.

Given $n = 12$ robots randomly scattered in a square area $\mathcal{Q}$. We consider

the continuous-time dynamics of mobile robots can be represented as a single-integrator system, i.e.,

$$p_{k+1} = p_k + u_k, \tag{4.128}$$

where $p_k = [p_k^x \; p_k^y]^\top$ denotes the $x-y$ position, while $u_k = [u_k^x \; u_k^y]^\top$ is the $x-y$ control input. The boundaries of the area are $\{(0,0),(0,1),(1,1),(1,0)\}$; and the information distribution is in the form of

$$\phi(q) = \mathcal{K}(q)^\top \theta. \tag{4.129}$$

Following the parametrisation in (4.117), the working area is divided into $m = 4$ partitions; and the kernel function, $\mathcal{K}(q) = [\mathcal{K}_1, \mathcal{K}_2, \cdots, \mathcal{K}_m]$, has elements given by

$$\mathcal{K}_l = \frac{\exp(-\frac{1}{2}(q-\mu_l)^\top \Sigma_l^{-1}(q-\mu_l))}{\sqrt{(2\pi)^d |\Sigma_l|}},$$

where d is the dimension of the environment - in this case $d = 2$. The elements of the vectorised kernel functions have peaks whose values are given by $\mu_1 = [0.25, 0.25]^\top$, $\mu_2 = [0.25, 0.75]^\top$, $\mu_3 = [0.75, 0.25]^\top$, and $\mu_4 = [0.75, 0.75]^\top$. The variances of all partitions are similar with $\Sigma_l = 0.02 I_2$, for $l = \{1, 2, 3, 4\}$, where $I_2 \in \mathbb{R}^{2\times 2}$ is an identity matrix. The target value of the weights to track is $\theta = [120.0, 2.5, 2.5, 160.0]^\top$. Since the mobile robot and the modified-consensus protocol are originally in a continuous-time system, in this work we choose to perform the simulation with sampling time $0.1s$.

The parameters required by the coverage controller and the initial values of the estimated weights, $\hat{\theta}_i$, in these two simulations are made identical. Since the modified-consensus observer requires some parameters to be given, we choose $\gamma = 3.63$, $\zeta = 0.6$ and $k = 1.5$ to show the oscillation caused by large gains. The gains chosen here In the distributed Kalman filter, no static constants are set other than the initial values of estimator and covariance matrix.

By applying the modified-consensus observer algorithm in the first scenario and the distributed Kalman filter in the second one, we obtain the trajectory of robots and the Voronoi partitions illustrated in Figs. 4.1a and 4.1b, respectively. The initial positions, final positions and estimated centroids are indicated using red 'x', blue 'o' and orange 'x'. From Fig. 4.1a, it can be seen that there is an

agent that do not converge to the centroid; while according to Fig. 4.1b all robots can successfully converge to the estimated centroids.

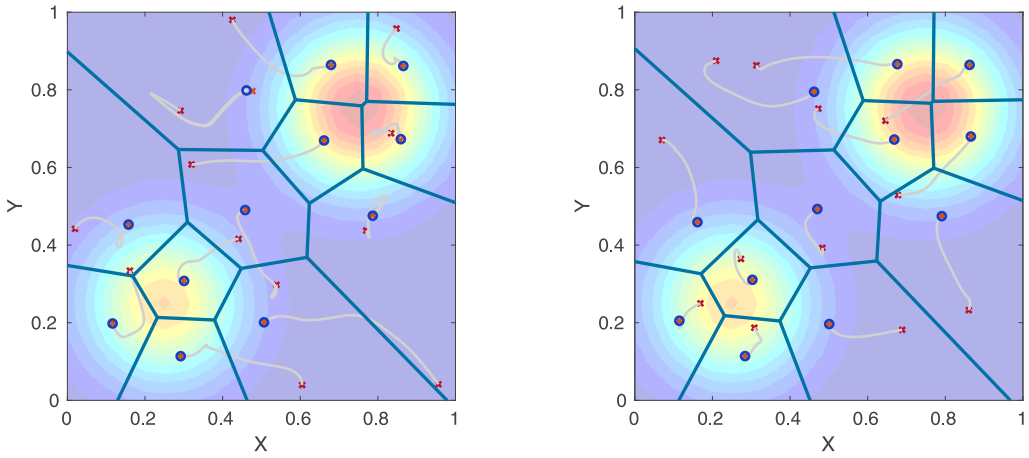
Related to the performance of field estimation algorithm, the implementation of the modified-consensus observer yields the convergence results of the estimated weights $\hat{\theta}_i$ presented in Fig. 4.2a; while the distributed Kalman filter yields the convergence results of the estimated weights given in Fig. 4.2b. These figures shows that the distributed Kalman filter has successfully driven the estimated weights to some values close to the target weights. As comparison, the estimation using modified-consensus observer shows high-amplitude oscillation with constant mean of several estimated weights.

The performance of coverage control using the modified-consensus observer as its field estimator is illustrated in Fig. 4.3a for convergence of the norm of the tracking errors $\|p_k^i - \hat{\mathcal{C}}_{V_i}\|$, and Fig. 4.4a for the convergence of the objective function; while results of the coverage control using the distributed Kalman filter are depicted in Fig. 4.3b for the convergence of the norm of the errors and Fig. 4.4b for the convergence of the objective functions. Since the weights obtained using the modified-consensus observer do not converge to zero, the errors also do not converge to zero. According to Figures 4.3a, the estimated objective function using the modified-consensus observer has steady state error with the true objective function. As comparison, using the distributed Kalman filter, the systems are able to minimise the errors and quickly track the true objective function.

Empirically, in addition to the simulation results presented above, we also calculate the average computation time of 500 iterations of these presented scenarios. Although there exist variations in the computation time for each iteration, the average time required to perform the iterations of coverage control with distributed Kalman filter always less than the one with modified-consensus observer. The average of the computation time of the first scenario with the modified-consensus observer varies along the process with average of $37.0349s$ per iteration, while the second scenario with our algorithm takes $30.5859s$ per iteration in average.

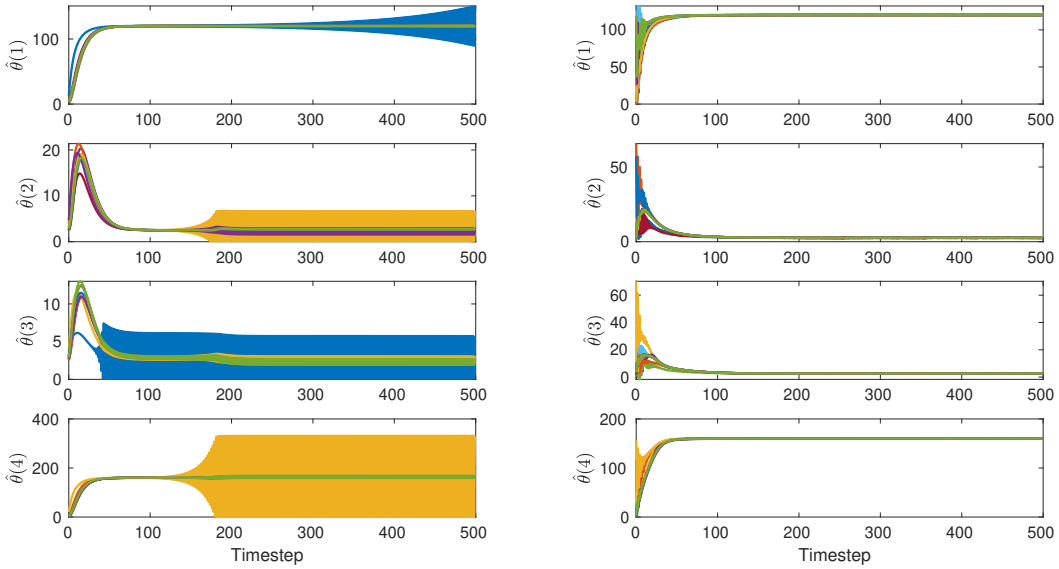
From the illustrated figures, it can be concluded that: 1) Both estimators can be used to estimate the information distribution; 2) It is true that the simulation

result in [21] has shown non-oscillating results with some chosen gains. However, the algorithm still requires the gains to be tuned but no exact upper boundary of the estimation gains has been given. Failure to pick the right gains may lead to oscillating performance and steady-state tracking error of the objective function as shown in this comparative simulation; 3) Unlike the algorithm in [21], the proposed distributed nonlinear Kalman filter needs no constant gains to be tuned to produce stable performance; 4) The computation time, which can also indicate the computation burden, of the distributed Kalman filter is faster than the modified-consensus observer algorithm in [21]. This might be caused by a number of surface integrations performed to adjust some variables in the modified-consensus observer where the distributed Kalman filter requires no additional surface integrations in the estimation process.



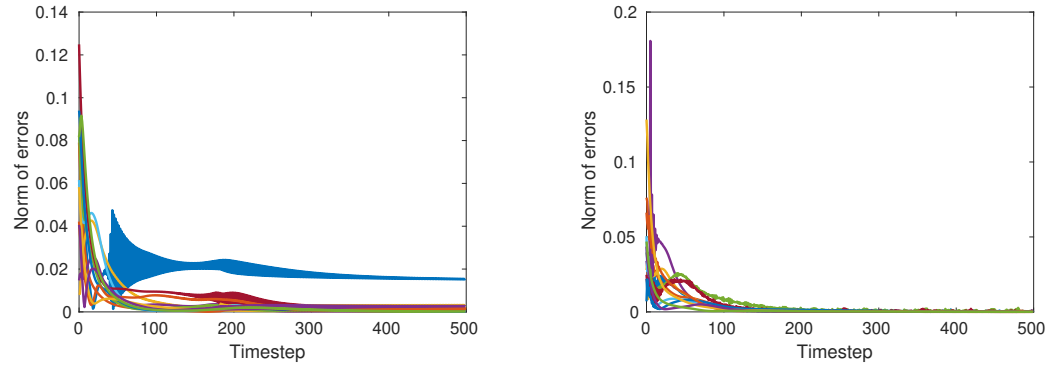
(a) Using modified-consensus observer. (b) Using Kalman-consensus estimator.

Figure 4.1: Position trajectories and optimal centroidal Voronoi regions.



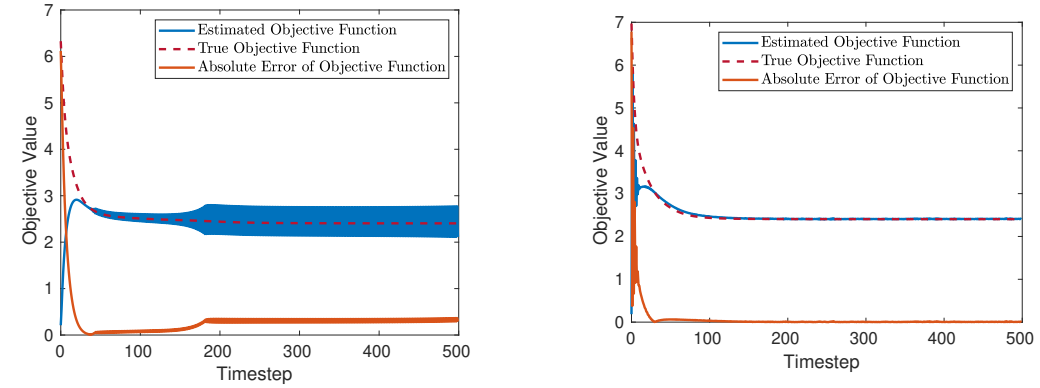
(a) Using modified-consensus observer. (b) Using Kalman-consensus estimator.

Figure 4.2: Convergence result of the estimated weights of the density function $\hat{\theta}_i, \forall i \in \mathcal{V}_n$



(a) Using modified-consensus observer. (b) Using Kalman-consensus estimator.

Figure 4.3: Convergence result of error $\|p_k^i - \hat{C}_{V_i}\|, \forall i \in \mathcal{V}_n$.



(a) Using modified-consensus observer. (b) Using Kalman-consensus estimator.

Figure 4.4: Convergence result of the objective function.

4.5 Conclusions

In this chapter, the distributed nonlinear Kalman filter with general communication scheme has been presented to estimate a dynamical process with additive white Gaussian noises in the system, measurement and communication. The optimal Kalman and communication gains have been provided such that the estimator has capability of using measurement and communicated information to produce an estimate value. By using this mechanism, if the communication topology is represented using a Laplacian matrix and the agents share their prior estimate values, the consensus protocol combined with Kalman filter reported in the literature could be considered as a special case of the proposed disributed unscented Kalman filter with communication protocol. After designing the estimation algorithm, we have analytically demonstrated that the estimate error is exponentially bounded in mean square with regards to some boundaries. As an example, the algorithm has also been applied to coverage problem scenarios with a previously-unidentified information distribution. Combined with the coverage control protocol in our results in Chapter 3, the proposed algorithm was used to estimate the density function and find the optimal deployment of robots. Two scenarios of numerical experiments have been carried out as a comparison with the existing method used for field estimation in coverage control problem. The results have shown that the proposed distributed nonlinear Kalman filter algorithm have outperformed the modified-consensus observer in [21] by successfully estimating the unknown density function of an environment indicated, driving all agents to the optimal centroid positions and minimise the true objective function without oscillations. Moreover, the simulation results indicated that the proposed algorithm needs less computation time than the existing modified-consensus observer.

Although the proposed algorithm can solve the distributed estimation in Euclidean space, direct implementation of this algorithm to estimation problem in general Riemannian manifold might result in estimate value that violates the manifold constraint. This situation necessitates the generalisation of the algorithm by accomodating the curvature of the manifold. Accordingly, in the next chapter,

we will investigate the lower bound of distributed Bayesian estimator in Riemannian manifolds and propose a distributed Kalman filter for such manifold-valued systems.

Chapter 5

Distributed Bayesian Estimator on Riemannian Manifolds

The recursive Bayesian estimator is one of the existing estimators aiming to minimise the mean-squared error between the estimate value and the true value. In the case of linear dynamical Euclidean systems with Gaussian PDFs, Bayesian-based estimation algorithm has lead to Kalman filter where the CRBs of the estimator is achieved. However, the optimality achieved in Euclidean system might not be achievable in Riemannian system due to the curved space.

In this chapter, we focus on developing the recursive version of ICRBs for multi-agent Bayesian estimator. To formulate the lower bound, we firstly define the distributed Bayesian estimation process by appending an additional step to the traditional Bayesian procedure. The proposed Bayesian sequence becomes prediction, measurement and coordination updates. The proposed Bayesian recursion in each agent starts with the prediction process with the prior probability distribution. The following process, that is, the measurement update employs the consensus of the prior probability distribution to calculate the posterior distribution. Afterwards, the proposed coordination process executes the minimisation of the Kullback-Liebler divergence of the posterior probability distributions obtained from neighbouring agents [29]. Accordingly, the intrinsic lower bound of every process is developed recursively by utilising the matrix block inversion as initiated in [81]. Different from the traditional CRB for Euclidean system, the proposed lower bounds will consequently include curvature terms of the manifold

as a function of the Fisher information matrix (FIM).

After presenting the intrinsic CRBs, a distributed nonlinear Kalman filter is designed to tackle the estimation problem of Riemannian information manifolds by following the proposed distributed Bayesian estimation strategy. It is worth noticing that the Riemannian setting requires Riemannian consensus protocol to find the consensus among the posterior estimate values [17]. Application to distributed quaternionic attitude estimation demonstrates the performance of the proposed filter.

This chapter is structured as follows. In Section I, we define the distributed Bayesian estimation scheme investigated in this paper. Section II subsequently presents our main results about the intrinsic CRBs for the distributed Bayesian estimator consisting of three parts: CRBs of local prediction update, local measurement update and local coordination updates. Section III presents a class of Bayesian filter, i.e., extension of DUKF, for systems on Riemannian manifolds, referred to as the distributed Riemannian Kalman filter (DRKF). Thereafter, in Section IV, the proposed filtering algorithm is applied to a distributed attitude estimation of multi-robot systems. Finally, Section V summarises the result of our work in this chapter.

5.1 Bayesian Estimation Problem

Let n agents be connected in a network represented by a graph $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$, for $\mathcal{V}_n = \{1, 2 \ldots, n\}$ and $\mathcal{E}_n \subseteq \mathcal{V}_n \times \mathcal{V}_n$. The dynamical process to be estimated by these agents is modelled as a function of state and process noise living in Riemannian manifolds. Let $\mathcal{P}_x$ and $\mathcal{P}_v$, respectively, denote the Riemannian manifolds of the process state and its noise with $d = \dim(\mathcal{P}_x) = \dim(\mathcal{P}_v)$; while $\mathcal{S}_y$ and $\mathcal{S}_w$ denote the manifolds where the measurement signal and its noise evolve with $q = \dim(\mathcal{S}_y) = \dim(\mathcal{S}_w)$, respectively. Denoting the process function by $f : \mathcal{P}_x \times \mathcal{P}_v \to \mathcal{P}_x$, process state by $x_k \in \mathcal{P}_x$ and process noise by $v_k \in \mathcal{P}_v$, the corresponding Riemannian discrete-time process considered in this paper is given by

$$x_{k+1} = f(x_k, v_k). \tag{5.1}$$

The measurement data obtained by each agent can be modelled using measurement function $h_i : \mathcal{P}_x \times \mathcal{S}_w \to \mathcal{P}_y$, such that the measurement update can be expressed as

$$y_k^i = h_i(x_k, w_k^i),\ \forall i \in \mathcal{V}_n, \tag{5.2}$$

with $y_k^i \in \mathcal{S}_y$ and $w_k^i \in \mathcal{S}_w$ being the measurement and the measurement noise by agent i, respectively.

A distributed intrinsic estimation problem is to determine the most likely value of $x_k \in \mathcal{P}_x$ based on a measured signal $y_k^i \in \mathcal{S}_y$ such that the difference between the estimated and true values are minimised at every time step k. This problem could be addressed by minimising the mean-squared error between the true and estimate values formulated using the logarithm mapping on manifold [84, 91], i.e.,

$$\min_{\hat{x}_k^i}\ \mathbb{E}\left\{\left(\mathrm{Log}_{x_k^i}(\hat{x}_k^i)\right)^\top\left(\mathrm{Log}_{x_k^i}(\hat{x}_k^i)\right)\right\}, \tag{5.3}$$

where the $\hat{x}_k^i \in \mathcal{P}_x$ and $x_k^i \in \mathcal{P}_x$ denote the estimate and true values of agent i at time step k, respectively. Since the agents are to observe the same process in (5.1), it could simply be said that $x_k^i = x_k$.

To minimise the objective function, we consider a Bayesian online estimator where at every time step k new measurement data are acquired to update the estimate values. The sequence of the states up to time step k can be augmented to become $X_k^i := (x_1^i, x_2^i, \ldots, x_k^i) \in \tilde{\mathcal{P}}_x$; and the measurements to become $Y_k^i := (y_1^i, y_2^i, \ldots, y_k^i) \in \tilde{\mathcal{S}}_y$. The proposed technique to minimise (5.3) adopted the distributed Bayesian estimation procedure utilising the Kullback-Liebler averaging method implemented by each agent in a network originated in [29].

Definition 5.1 (Distributed Recursive Bayesian Estimator)**.** Distributed recursive Bayesian estimator is a class of estimator which aims to find the estimate of state x_k^i, for $i \in \mathcal{V}_n$, distributedly by performing the following steps at every time step k in every agent i:

1. *Prediction*: predicting the subsequent value of x_k^i given the measurement signal up to time step $k-1$, Y_{k-1}^i, based on the prior probability distribution

$$p_i(x_k^i|Y_{k-1}^i) = \int p_i(x_k^i|x_{k-1}^i)\tilde{p}_i(x_{k-1}^i|Y_{k-1}^i)dx_{k-1}^i; \tag{5.4}$$

2. *Measurement*: acquiring new measurement signal y_k and updating new posterior estimate value based on the posterior probability distribution

$$p_i(x_k^i|Y_k^i) = \frac{p_i(y_k^i|x_k^i)p_i(x_k^i|Y_{k-1}^i)}{p_i(y_k^i|Y_{k-1}^i)}. \tag{5.5}$$

3. *Coordination*: communicating with the neighbouring agents to calculate the consensus value of the prior probability distributions using

$$\tilde{p}_i(x_k^i|Y_{k-1}^i) = \bigoplus_{j\in N_i} \left(\pi_{ij} \odot p_j(x_k^j|Y_{k-1}^j)\right), \tag{5.6}$$

where π_{ij} is the element (i, j) of the Perron matrix Π corresponding to the network graph $\mathcal{G}_n$;

Remark 5.2. Suppose that the probability density function of the state and measurement is Gaussian, and the state and measurement is Euclidean. Then, the well-known distributed Kalman filter and its variants are the optimal realisation of the recursive Bayesian estimator.

The objectives of this work are then to (i) formulate the intrinsic CRB of the distributed recursive Bayesian estimator; and (ii) design the Kalman filter algorithm suitable for distributed estimation in Riemannian systems.

5.2 Intrinsic Cramér-Rao Bounds for Recursive Bayesian Filter

This section presents the intrinsic CRBs for the estimator defined in the previous section. The discussion is splitted into three parts: formulation of intrinsic CRBs for local prediction, measurement, and finally measurement with coordination updates.

Continued from the description in Section 5.1, consider a state manifold $\mathcal{P}_x$ with $d = \dim(\mathcal{P}_x)$. For every agent $i \in \mathcal{V}_n$, the online orthogonal basis of the state manifold at time step k is denoted by

$$e_k^i = (e_{k,1}^i, e_{k,2}^i, \ldots, e_{k,d}^i), \text{ for } e_{k,j}^i \in \mathrm{T}_{x_k^i}\mathcal{P}_x. \tag{5.7}$$

Define an augmented orthogonal basis of agent i up to time step k as $\tilde{e}_k^i = e_{1:k}^i := (e_1^i, e_2^i, \ldots, e_k^i)$, with $\tilde{e}_{k,j}^i = e_{1:k,j}^i := (e_{1,j}^i, e_{2,j}^i, \ldots, e_{k,j}^i)$. Recall the properties of

Riemannian manifold that the product of Riemannian manifolds is also Riemannian manifold. It follows that, using the product of all the state manifolds stacked up to time step k, $\tilde{\mathcal{P}}_x$, the orthogonal basis of the state manifolds of agent i can be expressed as

$$\tilde{e}^i_k = (\tilde{e}^i_{k,1}, \tilde{e}^i_{k,2}, \ldots, \tilde{e}^i_{k,d}), \text{ for } \tilde{e}^i_{k,j} \in \mathrm{T}_{X^i_k}\tilde{\mathcal{P}}_x. \tag{5.8}$$

5.2.1 Local Prediction Update

Let $\bar{x}^i_k \in \mathcal{P}_x$ and $\bar{X}^i_k \in \tilde{\mathcal{P}}_x$ be the prior estimate or the prediction of a state x^i_k at time step k, and the augmented prior estimate up to time step k, of agent $i \in \mathcal{V}_n$, respectively. The prior estimate at time step k is evaluated by executing the prediction of the Bayesian estimation mechanism in (5.4). Our interest in this discussion is to establish the lower bound of the prior covariance matrix, that is, $\mathbb{E}\{\mathrm{Log}_{\bar{x}^i_k}(x^i_k)\mathrm{Log}_{\bar{x}^i_k}(x^i_k)^\top\} = \mathbb{E}\{\mathrm{Log}_{x^i_k}(\bar{x}^i_k)\mathrm{Log}_{x^i_k}(\bar{x}^i_k)^\top\}$.

A predictor of agent i at time step k, $\bar{x}^i_k : \tilde{\mathcal{S}}_y \to \mathcal{P}_x : Y^i_{k-1} \mapsto \bar{x}^i_k(Y^i_{k-1})$, is a mapping which associates the previous measurement manifolds up to time step $k-1$ to the current state manifold at time step k based on the prior distribution of the recursive Bayesian estimator. The augmented predictor can be defined as a mapping $\bar{X}^i_k : \tilde{\mathcal{S}}_y \to \tilde{\mathcal{P}}_x : Y^i_{k-1} \mapsto \bar{X}^i_k(Y^i_{k-1})$ from augmented measurement manifold available up to time step $k-1$ to the augmented state manifold up to time step k.

Define the prediction error at time step k as the difference between the true value x^i_k and the predicted value $\bar{x}^i_k$. Due to non-zero curvature, the error is obtained using the logarithm map of two points on manifold $\mathcal{P}_x$ such that

$$\bar{\xi}^i_k = \mathrm{Log}_{x^i_k}(\bar{x}^i_k) \in \mathrm{T}_{x^i_k}\mathcal{P}_x. \tag{5.9}$$

Let $\bar{\varrho}^i_k \in \mathbb{R}^p$ be a prediction vector of the corresponding prediction error on the state manifold $\mathcal{P}_x$. The entries of the prediction vector can be expressed as

$$\bar{\varrho}^i_{k,j} = \langle \mathrm{Log}_{x^i_k}(\bar{x}^i_k), e^i_{k,j} \rangle. \tag{5.10}$$

In this case, the norm of the prediction error is obtained via

$$\|\bar{\varrho}^i_k\| = \|\mathrm{Log}_{x^i_k}\bar{x}^i_k\| = \mathrm{dist}(x^i_k, \bar{x}^i_k). \tag{5.11}$$

As the time step increases along the iterations, an expanding parameter composed of the prediction errors up to time step k is then stacked as the augmented prediction error given by

$$\bar{\zeta}_k^i = \mathrm{Log}_{X_k^i}(\bar{X}_k^i) \in \mathrm{T}_{X_k^i}\tilde{\mathcal{P}}_x. \tag{5.12}$$

Accordingly, let an augmented prediction vector of the state up to time step k be denoted by $\bar{\vartheta}_k^i \in \mathbb{R}^{kd}$, for $d = \dim(\mathcal{P}_x)$. The entries of the augmented prediction vector is

$$\bar{\vartheta}_{k,j}^i = \langle \mathrm{Log}_{X_k^i}(\bar{X}_k^i), \tilde{e}_{k,j}^i \rangle, \tag{5.13}$$

leading to the norm of the augmented prediction error given by

$$\|\bar{\vartheta}_k^i\| = \|\mathrm{Log}_{X_k^i}\bar{X}_k^i\| = \mathrm{dist}(X_k^i, \bar{X}_k^i). \tag{5.14}$$

By employing the prediction vector of agent $i \in \mathcal{V}_n$ at time step k, the covariance matrix of the prediction error at time step k is defined as a symmetric, positive semidefinite matrix $\bar{C}_k^i \in \mathbb{R}^{p\times p}$ written as

$$\bar{C}_k^i = \mathbb{E}\{\bar{\varrho}_k^i(\bar{\varrho}_k^i)^\top\}, \tag{5.15}$$

which is elementarily equivalent to

$$\bar{C}_{k,rs}^i = \mathbb{E}\{\langle \mathrm{Log}_{x_k^i}(\bar{x}_k^i), e_{k,r}^i \rangle \cdot \langle \mathrm{Log}_{x_k^i}(\bar{x}_k^i), e_{k,s}^i \rangle\}.$$

The expanding covariance matrix resulted from augmenting the predictions up to time step k can be obtained using the augmented prediction vector, $\bar{\Omega}_k^i \in \mathbb{R}^{kp\times kp}$. The prediction covariance matrix can be written as

$$\bar{\Omega}_k^i = \mathbb{E}\{\bar{\vartheta}_k^i(\bar{\vartheta}_k^i)^\top\}, \tag{5.16}$$

whose entries are

$$\bar{\Omega}_{k,rs}^i = \mathbb{E}\{\langle \mathrm{Log}_{X_k^i}(\bar{X}_k^i), \tilde{e}_{k,r}^i \rangle \cdot \langle \mathrm{Log}_{X_k^i}(\bar{X}_k^i), \tilde{e}_{k,s}^i \rangle\}.$$

In this discussion, it is assumed that the probability distribution of estimations at time step k_1 at k_2, for $k_1 \neq k_2$, is uncorrelated, i.e., $\mathbb{E}\{\bar{\varrho}_{k_1}^i \bar{\varrho}_{k_2}^i\} = 0$. This assumption allows us to write $\bar{\Omega}_k^i = \mathrm{diag}(\bar{C}_1^i, \bar{C}_2^i, \ldots, \bar{C}_k^i)$.

Before presenting the definition of Fisher Information Matrix (FIM), we require the logarithm of PDF of a predictor in Bayesian estimation. Define a function $\tilde{L} : \tilde{\mathcal{P}}_x \to \mathbb{R} : X \mapsto \log p(X, \cdot)$, as the natural logarithm of a given probability distribution that maps an augmented probability space to a scalar. Let the joint probability distribution of augmented states and previous measurements be parametrised as $p_i(X_k^i, Y_{k-1}^i)$, for $X_k^i \in \tilde{\mathcal{P}}_x$ and $Y_{k-1}^i \in \tilde{\mathcal{S}}_y$. The logarithm of the joint distribution $p_i(X_k^i, Y_{k-1}^i)$ is

$$\tilde{L}(X_k^i) = \log p_i(X_k^i, Y_{k-1}^i). \tag{5.17}$$

The logarithm of the joint PDF updated every time step k, $L : \mathcal{P}_x \to \mathbb{R} : x \mapsto \log p(x, \cdot)$, can be expressed as

$$L(x_k^i) = \log p_i(x_k^i, Y_{k-1}^i). \tag{5.18}$$

In order to measure of the steepness of the log-distribution and reflect the sensitivity to changes with respect to the parameter values, we also need a parameter referred to as score vector. For $i \in \mathcal{V}_n$, define a score function of a parameter at time step k as $s_k^i : \mathcal{P}_x \to \mathbb{R}^d$, which maps a probability space to a vector – referred to as the score vector. The score vector is resulted from the directional derivative of a log-distribution function $L(x_k^i) = \log p(x_k^i, \cdot)$ with respect to the state at time step k, x_k^i in the direction of basis vector $e_{k,j}^i$ on the manifold $\mathcal{P}_x$. Correspondingly, the score vector of the predicted value at time step k, also referred to as prior score vector, is defined using (5.18) and elementarily given by

$$\bar{s}_{k,r}^i = s_{k,r}^i(x_k^i | Y_{k-1}^i) = D_k^i L(x_k^i | Y_{k-1}^i)[e_{k,r}^i], \tag{5.19}$$

where $\log p_i(x_k^i, Y_{k-1}^i) = \log p_i(x_k^i | Y_{k-1}^i) + \log p_i(Y_{k-1}^i)$ has been employed and the derivative of the second term vanishes. Furthermore, let $\tilde{s}_k^i : \tilde{\mathcal{P}}_x \to \mathbb{R}^{kd}$ be an augmented score function up to time step k. In the prediction step, stacking the prior score vectors up to time step k gives the augmented prior score vector based on (5.17), $\bar{\sigma}_k^i \in \mathbb{R}^{kd}$, whose elements are

$$\bar{\sigma}_{k,r}^i = \tilde{s}_{k,r}^i(X_k^i | Y_{k-1}^i) = D_k^i \tilde{L}(X_k^i | Y_{k-1}^i)[\tilde{e}_{k,r}^i]. \tag{5.20}$$

The prior score function entails the historical prediction up to time step k such that $\tilde{s}_k^i(X_k^i | Y_{k-1}^i) = \tilde{s}_k^i(X_k^i | X_{k-1}^i) + \tilde{s}_k^i(X_{k-1}^i | Y_{k-1}^i)$. Based on the definition of

the score vectors, it is straightforward to demonstrate that the above prior score vectors have zero mean, that is, $\mathbb{E}\{\bar{s}_k^i\} = \mathbb{E}\{\bar{\sigma}_k^i\} = 0$.

In our discussion, the subscripted time-step index attached to the score vectors indicates the time step of the state to which the gradient of the directional derivative is taken. For example, consider $\bar{s}_{\kappa,r}^i = s_{\kappa,r}^i(x_k^i|\cdot) = D_\kappa^i L(x_k^i|\cdot)[e_{k,r}^i]$, for some scalar κ. In this case, κ indicates that the directional derivative of $\log p(x_k^i|\cdot)$ is taken with respect to the state at specified time step x_κ^i, which might not equal to x_k^i.

Prior FIM at time step k, $\bar{F}_k^i \in \mathbb{R}^{d\times d}$, is a symmetric, positive semidefinite matrix correponding to the prediction score vector (5.19) defined as

$$\bar{F}_k^i = \mathbb{E}\{\bar{s}_k^i(\bar{s}_k^i)^\top\}, \tag{5.21}$$

whose entries are

$$\bar{F}_{k,rs}^i = \mathbb{E}\{D_k^i L(x_k^i|Y_{k-1}^i)[e_{k,r}^i] \cdot D_k^i L(x_k^i|Y_{k-1}^i)[e_{k,s}^i]\}.$$

The FIMs from the first to the k-th iterations can be augmented to form another symmetric, positive semidefinite matrix $\bar{\Gamma}_k^i \in \mathbb{R}^{kd\times kd}$ expressed as

$$\bar{\Gamma}_k^i = \mathbb{E}\left\{\bar{\sigma}_k^i(\bar{\sigma}_k^i)^\top\right\}, \tag{5.22}$$

and elementarily as

$$\bar{\Gamma}_{k,rs}^i = \mathbb{E}\{D_k^i \tilde{L}(X_k^i|Y_{k-1}^i)[\tilde{e}_{k,r}^i] \cdot D_k^i \tilde{L}(X_k^i|Y_{k-1}^i)[\tilde{e}_{k,s}^i]\}.$$

It could naturally be observed that the matrix will always be expanding as the time step increases.

To establish the CRB, we require several lemmas stated as follows. The first lemma provides a solution of the Jacobi field equation on a Riemannian manifold with constant sectional curvature. This lemma is a restructured from a part of the proof in [117].

Lemma 5.3 (Solution of Jacobi Field Equation for Constant Sectional Curvature)**.** *Consider a Riemannian manifold $\mathcal{M}$ with constant sectional curvature $K \in \mathbb{R}$ and a geodesic $\gamma : I \subset \mathbb{R} \to \mathcal{M}$. Then, the Jacobi vector field along the geodesic is expressed as*

$$J(t) = u_1(t)E(t) + u_2(t)\dot{\gamma}(t), \tag{5.23}$$

where $E(t)$ denotes a parallel vector field along γ and is orthogonal to $\dot{\gamma}(t)$; the scalar functions $u_1 : \mathbb{R} \to \mathbb{R}$ and $u_2 : \mathbb{R} \to \mathbb{R}$ are, respectively, given by

$$u_1(t) = \begin{cases} t & \text{if } K = 0, \\ \frac{1}{\sin(\sqrt{K}\|\dot{\gamma}\|)} \sin(\sqrt{K}\|\dot{\gamma}\|t) & \text{if } K > 0, \\ \frac{1}{\sinh(\sqrt{-K}\|\dot{\gamma}\|)} \sinh(\sqrt{-K}\|\dot{\gamma}\|t) & \text{if } K < 0, \end{cases}$$

and $u_2(t) = \beta t$, for some values $u_1(0) = u_2(0) = 0$, $u_1(1) = 1$ and $u_2(1) = \beta$. Furthermore, the covariant derivative of the Jacobi field is

$$\nabla_{\dot{\gamma}} J(t) = u_1'(t)E(t) + u_2'(t)\dot{\gamma}(t), \tag{5.24}$$

where

$$u_1'(t) = \begin{cases} 1 & \text{if } K = 0, \\ \frac{\sqrt{K}\|\dot{\gamma}\|}{\sin(\sqrt{K}\|\dot{\gamma}\|)} \cos(\sqrt{K}\|\dot{\gamma}\|t) & \text{if } K > 0, \\ \frac{\sqrt{K}\|\dot{\gamma}\|}{\sinh(\sqrt{-K}\|\dot{\gamma}\|)} \cosh(\sqrt{-K}\|\dot{\gamma}\|t) & \text{if } K < 0. \end{cases}$$

and $u_2'(t) = \beta$.

Proof. The claim follows straightforwardly after plugging (5.23) into the Jacobi field equation,

$$\nabla_{\dot{\gamma}}^2 J(t) + R(J(t), \dot{\gamma}(t))\dot{\gamma}(t) = 0,$$

where $R(u, v)w$ is the Riemannian curvature tensor for $u, v, w \in \mathcal{X}(\mathcal{M})$. □

To establish the our result in the prediction update, the proofs of the following lemma and theorem require the defined instrumental vectors and matrices: the augmented prediction error in (5.12), prediction vector in (5.13), prior covariance matrix in (5.16), prior score vector in (5.20) and prior FIM in (5.22).

The following lemma states the cross-covariance matrix of prior score and prediction vectors which is adapted from [117] and [118].

Lemma 5.4 (Score-Predition Covariance Matrix)**.** *Let $\mathcal{P}_x$ denote a d-dimensional Riemannian manifold of the state parameter with constant sectional curvature $K \in \mathbb{R}$. Consider a predictor in a recursive Bayesian estimation with prediction error, prediction vector, covariance matrix, and prior score vector at time step k*

given in (5.9), (5.10), (5.15) and (5.19), respectively. Then, the score-prediction covariance matrix is given by

$$\mathbb{E}\{\bar{s}_k^i(\bar{\varrho}_k^i)^\top\} = \mathbb{E}\{\bar{\varrho}_k^i(\bar{s}_k^i)^\top\} = I - \frac{1}{3}R_{\mathrm{m}}(\bar{C}_k^i) + \mathcal{O}\left(\mathbb{E}\{K^2\|\bar{\xi}_k^i\|^4\}\right) \tag{5.25}$$

where $R_{\mathrm{m}}(\bar{C}_k^i) : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ *is a linear map given elementarily by*

$$R_{\mathrm{m}}(\bar{C}_k^i)_{rs} = \sum_{j,l}\langle R(e_{k,j}^i, e_{k,r}^i)e_{k,s}^i, e_{k,l}^i\rangle \bar{C}_{k,jl}^i. \tag{5.26}$$

Proof. Let us express the expected value augmented prediction error stacked in agent i, for $i \in \mathcal{V}_n$, up to time step k in (5.12) as

$$\mathbb{E}\{\bar{\zeta}_k^i\} = \int_{\tilde{\mathcal{P}}_x} \bar{\zeta}_k^i(X_k^i)p_i(X_k^i|Y_{k-1}^i)dX_k^i = 0, \tag{5.27}$$

where $\bar{\zeta}_k^i = \mathrm{Log}_{X_k^i}(\bar{X}_k^i)$.

Consider two arbitrary tangent vector fields on manifold $\tilde{\mathcal{P}}_x$ at point X_k^i, $\tilde{U}, \tilde{V} \in T_{X_k^i}\tilde{\mathcal{P}}_x$ with vector components $\tilde{u}, \tilde{v} \in \mathbb{R}^{kd}$, where $\tilde{U} = \sum_{j=1}^{kd}\tilde{u}_j\tilde{e}_{k,j}^i$ and $\tilde{V} = \sum_{j=1}^{kd}\tilde{v}_j\tilde{e}_{k,j}^i$. Then, taking the covariant derivative with respect to $\tilde{U}$ on both sides of (5.27), followed by taking the Riemannian inner product with $\tilde{U}$ yields

$$\int_{\tilde{\mathcal{P}}_x}\langle\nabla_{\tilde{U}}\bar{\zeta}_k^i(X_k^i)p_i(X_k^i|Y_{k-1}^i), \tilde{V}\rangle dX_k^i = 0. \tag{5.28}$$

Recall the product rule of a covariant derivative of a scalar-multiplied vector field, i.e., $\nabla_Z fX = f\nabla_Z X + Df[Z]X$, for a scalar function f and vector field X, Z with appropriate dimension. Hence, with the property of the derivative of logarithm function,

$$D_k^i\tilde{L}(X_k^i|Y_{k-1}^i)[\tilde{U}] = \frac{D_k^i p_i(X_k^i|Y_{k-1}^i)[\tilde{U}]}{p_i(X_k^i|Y_{k-1}^i)},$$

(5.28) could be expanded to

$$\int_{\tilde{\mathcal{P}}_x}\langle D_k^i\tilde{L}(X_k^i|Y_{k-1}^i)[\tilde{U}]\cdot\bar{\zeta}_k^i(X_k^i) + \nabla_{\tilde{U}}\bar{\zeta}_k^i(X_k^i), \tilde{V}\rangle p_i(X_k^i|Y_{k-1}^i)dX_k^i = 0. \tag{5.29}$$

Since $D_k^i\tilde{L}(X_k^i|Y_{k-1}^i)[\tilde{U}] = \langle\nabla_{X_k^i}\tilde{L}(X_k^i|Y_{k-1}^i), \tilde{U}\rangle = \tilde{u}^\top\bar{\sigma}_k^i$ and $\langle\bar{\zeta}_k^i, \tilde{V}\rangle = (\bar{\vartheta}_k^i)^\top\tilde{v}$, from (5.29) one have

$$\tilde{u}^\top\mathbb{E}\{\bar{\sigma}_k^i(\bar{\vartheta}_k^i)^\top\}\tilde{v} = -\mathbb{E}\{\langle\nabla_{\tilde{U}}\bar{\zeta}_k^i, \tilde{V}\rangle\}. \tag{5.30}$$

For every agent $i \in \mathcal{V}_n$ at time step k, let the geodesic connecting $X_k^i \in \tilde{\mathcal{P}}_x$ and $\bar{X}_k^i \in \tilde{\mathcal{P}}_x$ be expressed as

$$\gamma_k^i(t) = \mathrm{Exp}_{\bar{X}_k^i}(t\mathrm{Log}_{\bar{X}_k^i}(X_k^i))$$

where $\gamma_k^i(0) = \bar{X}_k^i$, $\gamma_k^i(1) = X_k^i$ and $\dot{\gamma}_k^i(1) = \bar{\zeta}_k^i$. In the subsequent steps, we utilise the fact shown in [119] that

$$-\nabla_{\tilde{U}}\bar{\zeta}_k^i = \nabla_{\dot{\gamma}} J_k^i(1), \tag{5.31}$$

for J_k^i being a vector field along γ_k^i satisfying

$$\nabla_{\dot{\gamma}}^2 J_k^i(t) + R(J_k^i(t), \dot{\gamma}_k^i(t))\dot{\gamma}_k^i(t) = 0, \tag{5.32}$$

where $\|\dot{\gamma}_k^i(t)\| = \|\bar{\zeta}_k^i\|$ is the geodesic distance between $\bar{X}_k^i$ and X_k^i.

Let the vector field $\tilde{U}$ be constructed as $\tilde{U} = \tilde{U}_\perp + \beta\bar{\zeta}_k^i$, with $\langle \tilde{U}_\perp, \bar{\zeta}_k^i \rangle = 0$ and a scalar β. Let $E(t)$ be a parallel vector field along $\gamma_k^i(t)$ such that, at $t = 1$, $E(1) = \tilde{U}_\perp$. By employing Lemma 5.3, the covariant derivative of the solution of (5.32) at $t = 1$ is

$$\nabla_{\dot{\gamma}} J_k^i(1) = u_1'(1)\tilde{U}_\perp + u_2'(t)\bar{\zeta}_k^i, \tag{5.33}$$

where

$$u_1'(1) = \begin{cases} 1 & \text{if } K = 0, \\ \sqrt{K}\|\bar{\zeta}_k^i\| \cot(\sqrt{K}\|\bar{\zeta}_k^i\|) & \text{if } K > 0, \\ \sqrt{-K}\|\bar{\zeta}_k^i\| \coth(\sqrt{-K}\|\bar{\zeta}_k^i\|) & \text{if } K < 0, \end{cases}$$

and $u_2'(1) = \beta$. Applying Taylor expansion series, i.e., $x\cot(x) = 1 - x^2/3 + \mathcal{O}(x^4)$ and $x\coth(x) = 1 + x^2/3 + \mathcal{O}(x^4)$, to $u_1'(1)$ yields (5.33) rewritten as

$$\nabla_{\dot{\gamma}} J_k^i(1) = \tilde{U} - \frac{1}{3}K\|\bar{\zeta}_k^i\|^2\tilde{U}_\perp + \mathcal{O}\left(\mathbb{E}\{K^2\|\bar{\zeta}_k^i\|^4\}\right)\tilde{U}_\perp. \tag{5.34}$$

By subtituting (5.34) to (5.30) using (5.31), it follows that

$$\begin{aligned} \tilde{u}^\top \mathbb{E}\{\bar{\sigma}_k^i(\bar{\vartheta}_k^i)^\top\}\tilde{v} &= \mathbb{E}\{\langle \nabla_{\dot{\gamma}} J_k^i(1), \tilde{V}\rangle\} \\ &= \mathbb{E}\{\langle \tilde{U}, \tilde{V}\rangle - \langle \frac{1}{3}K\|\bar{\zeta}_k^i\|^2\tilde{U}_\perp, \tilde{V}\rangle + \langle \mathcal{O}\left(\mathbb{E}\{K^2\|\bar{\zeta}_k^i\|^4\}\right)\tilde{U}_\perp, \tilde{V}\rangle\} \end{aligned} \tag{5.35}$$

Recall the relationship between Riemannian sectional curvature and curvature tensor, we have $\langle R(\bar{\zeta}_k^i, \tilde{U}_\perp)\tilde{V}, \bar{\zeta}_k^i\rangle = K(\langle \tilde{U}_\perp, \tilde{V}\rangle\langle \bar{\zeta}_k^i, \bar{\zeta}_k^i\rangle - \langle \bar{\zeta}_k^i, \tilde{V}\rangle\langle \tilde{U}_\perp, \bar{\zeta}_k^i\rangle)$. Since $\langle \tilde{U}_\perp, \bar{\zeta}_k^i\rangle = 0$, we have $\langle R(\bar{\zeta}_k^i, \tilde{U}_\perp)\tilde{V}, \bar{\zeta}_k^i\rangle = K(\langle \tilde{U}_\perp, \tilde{V}\rangle\langle \bar{\zeta}_k^i, \bar{\zeta}_k^i\rangle)$. Due to skew-symmetry of curvature tensor, we also have $R(\bar{\zeta}_k^i, \tilde{U}_\perp)\tilde{V} = R(\bar{\zeta}_k^i, \tilde{U})\tilde{V}$. Injecting these facts to the second term of the right-hand side of (5.35) leads to

$$\tilde{u}^\top \mathbb{E}\{\bar{\sigma}_k^i(\bar{\vartheta}_k^i)^\top\}\tilde{v} = \mathbb{E}\{\langle \tilde{U}, \tilde{V}\rangle - \frac{1}{3}\langle R(\bar{\zeta}_k^i, \tilde{U})\tilde{V}, \bar{\zeta}_k^i\rangle + \mathcal{O}\left(\mathbb{E}\{K^2\|\bar{\zeta}_k^i\|^4\}\right)\langle \tilde{U}_\perp, \tilde{V}\rangle\}. \tag{5.36}$$

Consequently, the augmented score-prediction covariance matrix can be expressed as

$$\mathbb{E}\{\bar{\sigma}_k^i(\bar{\vartheta}_k^i)^\top\} = I - \frac{1}{3}\tilde{R}_{\mathrm{m}}\left(\bar{\Omega}_k^i\right) + \mathcal{O}\left(\mathbb{E}\{K^2\|\bar{\zeta}_k^i\|^4\}\right), \tag{5.37}$$

where $\tilde{R}_{\mathrm{m}}(\bar{\Omega}_k^i) : \mathbb{R}^{kd\times kd} \to \mathbb{R}^{kd\times kd}$ is given by

$$\begin{aligned}\tilde{R}_{\mathrm{m}}(\bar{\Omega}_k^i)_{rs} &= \mathbb{E}\{\langle R(\bar{\zeta}_k^i, \tilde{e}_{k,r}^i)\tilde{e}_{k,s}^i, \bar{\zeta}_k^i\rangle\} \\ &= \sum_{j,l}\langle R(\tilde{e}_{k,j}^i, \tilde{e}_{k,r}^i)\tilde{e}_{k,s}^i, \tilde{e}_{k,l}^i\rangle\bar{\Omega}_{k,jl}^i.\end{aligned}$$

Let us explicitly arrange the augmented prediction vector at agent i in (5.13) stacked up to time step k as $\bar{\vartheta}_k^i = [(\bar{\vartheta}_{k-1}^i)^\top, (\bar{\varrho}_k^i)^\top]^\top$. Likewise, the augmented score vector in (5.20) is decomposed to $\bar{\sigma}_k^i = [(\bar{\sigma}_{k-1}^i)^\top, (\bar{s}_k^i)^\top]^\top$. By utilising these decompositions, the left-hand side of (5.37) can be stacked as

$$\mathbb{E}\{\bar{\sigma}_k^i(\bar{\vartheta}_k^i)^\top\} = \mathbb{E}\left\{\begin{bmatrix}\bar{\sigma}_{k-1}^i(\bar{\vartheta}_{k-1}^i)^\top & 0 \\ 0 & \bar{s}_k^i(\bar{\varrho}_k^i)^\top\end{bmatrix}\right\}. \tag{5.38}$$

Let the prediction error up to time step k be arranged as $\bar{\zeta}_k^i = (\bar{\zeta}_{k-1}^i, \bar{\xi}_k^i)$ and the orthogonal basis as $\tilde{e}_{k,r}^i = (\tilde{e}_{i\{k-1\},r}, e_{k,r}^i)$. Applying them to the second term of the right-hand side of (5.36) leads to $\tilde{R}_{\mathrm{m}}(\bar{\Omega}_k^i)$ structured as

$$\tilde{R}_{\mathrm{m}}(\bar{\Omega}_k^i) = \begin{bmatrix}\tilde{R}_{\mathrm{m}}(\bar{\Omega}_{k-1}^i) & 0 \\ 0 & R_{\mathrm{m}}(\bar{C}_k^i)\end{bmatrix}, \tag{5.39}$$

with R_{m} defined in (5.26). Notice that the implication of the independency of information at time step $k-1$ and k has been employed to both (5.38) and (5.39), i.e., the expected value of the cross-covariance between two variables at time step k and $k-1$ equals to zero.

From (5.38), it can be observed that the state-prediction covariance matrix at time step k can be obtained by extracting the lower-right entries of matrices $\mathbb{E}\{\bar{\sigma}_k^i(\bar{\vartheta}_k^i)^\top\}$ and $\tilde{R}_{\mathrm{m}}(\bar{\Omega}_k^i)$. Finally, plugging (5.38) and (5.39) into (5.37), then taking the lower-right sub-matrix of all terms yields

$$\mathbb{E}\{\bar{s}_k^i(\bar{\varrho}_k^i)^\top\} = \mathbb{E}\{\bar{\varrho}_k^i(\bar{s}_k^i)^\top\} = I - \frac{1}{3}R_{\mathrm{m}}(\bar{C}_k^i) + \mathcal{O}\left(\mathbb{E}\{K^2\|\bar{\xi}_k^i\|^4\}\right). \tag{5.40}$$

This completes the proof. □

The following theorem states our first result about the intrinsic CRB for Bayesian prediction update adapting the arguments in [117], and [118] and [81].

Theorem 5.5 (Intrinsic Prior CRB for Bayesian Prediction)**.** *Consider a group of n agents connected via a graph $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$. Let $\mathcal{P}_x$ and $\mathcal{S}_y$ be Riemannian state and measurement manifolds, respectively. For each agent $i \in \mathcal{V}_n$, let $\bar{x}_k^i \in \mathcal{P}_x$ be a predictor, equipped with score function in (5.19), prior covariance matrix in (5.15) and FIM in (5.21), at time step k following the prediction step of recursive Bayesian estimator in (5.4). Let the orthogonal basis of the predictor be given by $e_k^i = (e_{k,1}^i, e_{k,2}^i, \ldots, e_{ik,d})$, for $d = \dim(\mathcal{P}_x)$. Then, with the prediction error and prediction vector at time step k given by (5.9) and (5.10), respectively, the prior covariance matrix satisfies the following inequality:*

$$\bar{C}_k^i \geqslant (\bar{F}_k^i)^{-1} - \frac{1}{3}\left((\bar{F}_k^i)^{-1}R_{\mathrm{m}}((\bar{F}_k^i)^{-1}) + R_{\mathrm{m}}((\bar{F}_k^i)^{-1})(\bar{F}_k^i)^{-1}\right) + \mathcal{O}\left(\mathbb{E}\{K^2\|\bar{\xi}_k^i\|^4\}\right), \tag{5.41}$$

where $R_{\mathrm{m}}((\bar{F}_k^i)^{-1})_{rs} = \sum_{j,l}\langle R(e_{k,j}^i, e_{k,r}^i)e_{k,s}^i, e_{k,l}^i\rangle((\bar{F}_k^i)^{-1})_{jl}$ and

$$\bar{F}_k^i = S_k^i - (B_k^i)^\top(\hat{F}_{k-1}^i + A_k^i)^{-1}B_k^i \tag{5.42}$$

with

$$\begin{aligned}
S_k^i &= \mathbb{E}\{s_k^i(x_k^i|x_{k-1}^i)s_k^i(x_k^i|x_{k-1}^i)^\top\},\\
B_k^i &= \mathbb{E}\{s_{k-1}^i(x_k^i|x_{k-1}^i)s_k^i(x_k^i|x_{k-1}^i)^\top\} = (B_k^i)^\top,\\
A_k^i &= \mathbb{E}\{s_{k-1}^i(x_k^i|x_{k-1}^i)s_{k-1}^i(x_k^i|x_{k-1}^i)^\top\},\ \textit{and}\\
\hat{F}_k^i &= \mathbb{E}\{s_{k-1}^i(x_{k-1}^i|y_{k-1}^i)s_{k-1}^i(x_{k-1}^i|y_{k-1}^i)^\top\}.
\end{aligned}$$

Proof. For agent $i \in \mathcal{V}_n$, let a random vector formed up to time step k be given by $\bar{\nu}_k^i \in \mathbb{R}^{kd}$, where $d = \dim(\mathcal{P}_x)$, such that $\mathbb{E}\{\bar{\nu}_k^i(\bar{\nu}_k^i)^\top\} \geqslant 0$ holds. This vector

can be chosen as

$$\bar{\nu}_k^i = \bar{\vartheta}_k^i - (\bar{\Gamma}_k^i)^{-1}\bar{\sigma}_k^i,$$

whose expected value of $\bar{\nu}_k^i$ is zero. By utilising this vector, consider a covariance matrix $\mathbb{E}\{\bar{\nu}_k^i(\bar{\nu}_k^i)^\top\}$:

$$\begin{aligned}\mathbb{E}\left\{\bar{\nu}_k^i(\bar{\nu}_k^i)^\top\right\} = \mathbb{E}\left\{\bar{\vartheta}_k^i(\bar{\vartheta}_k^i)^\top\right\} + (\bar{\Gamma}_k^i)^{-1}\mathbb{E}\left\{\bar{\sigma}_k^i(\bar{\sigma}_k^i)^\top\right\}(\bar{\Gamma}_k^i)^{-1} \\ - (\bar{\Gamma}_k^i)^{-1}\mathbb{E}\left\{\bar{\sigma}_k^i(\bar{\vartheta}_k^i)^\top\right\} - \mathbb{E}\left\{\bar{\vartheta}_k^i(\bar{\sigma}_k^i)^\top\right\}(\bar{\Gamma}_k^i)^{-1} \geqslant 0. \quad (5.43)\end{aligned}$$

Now, let us inspect the element of each term in (5.43) by firstly noting that $\bar{\vartheta}_k^i = [(\bar{\vartheta}_{k-2}^i)^\top, (\bar{\varrho}_{k-1}^i)^\top, (\bar{\varrho}_k^i)^\top]^\top$, $\bar{\sigma}_k^i = [(\bar{\sigma}_{k-2}^i)^\top, (\bar{s}_{k-1}^i)^\top, (\bar{s}_k^i)^\top]^\top$, $\mathbb{E}\{\bar{\sigma}_k^i(\bar{\sigma}_k^i)^\top\} = \bar{\Gamma}_k^i$ and $\mathbb{E}\{\bar{\vartheta}_k^i(\bar{\vartheta}_k^i)^\top\} = \bar{\Omega}_k^i$. It follows that

$$\bar{\Omega}_k^i = \begin{bmatrix}\mathbb{E}\left\{\bar{\vartheta}_{k-2}^i(\bar{\vartheta}_{k-2}^i)^\top\right\} & 0 & 0 \\ 0 & \mathbb{E}\left\{\bar{\varrho}_{k-1}^i(\bar{\varrho}_{k-1}^i)^\top\right\} & 0 \\ 0 & 0 & \mathbb{E}\left\{\bar{\varrho}_k^i(\bar{\varrho}_k^i)^\top\right\}\end{bmatrix}, \quad (5.44)$$

$$(\bar{\Gamma}_k^i)^{-1} = \begin{bmatrix}\mathbb{E}\left\{\bar{\sigma}_{k-2}^i(\bar{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\bar{\sigma}_{k-2}^i(\bar{s}_{k-1}^i)^\top\right\} & 0 \\ \mathbb{E}\left\{\bar{s}_{k-1}^i(\bar{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\bar{s}_{k-1}^i(\bar{s}_{k-1}^i)^\top\right\} & \mathbb{E}\left\{\bar{s}_{k-1}^i(\bar{s}_k^i)^\top\right\} \\ 0 & \mathbb{E}\left\{\bar{s}_k^i(\bar{s}_{k-1}^i)^\top\right\} & \mathbb{E}\left\{\bar{s}_k^i(\bar{s}_k^i)^\top\right\}\end{bmatrix}^{-1}, \quad (5.45)$$

and

$$\begin{aligned}(\bar{\Gamma}_k^i)^{-1}\mathbb{E}\left\{\bar{\sigma}_k^i(\bar{\vartheta}_k^i)^\top\right\} = &\begin{bmatrix}\mathbb{E}\left\{\bar{\sigma}_{k-2}^i(\bar{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\bar{\sigma}_{k-2}^i(\bar{s}_{k-1}^i)^\top\right\} & 0 \\ \mathbb{E}\left\{\bar{s}_{k-1}^i(\bar{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\bar{s}_{k-1}^i(\bar{s}_{k-1}^i)^\top\right\} & \mathbb{E}\left\{\bar{s}_{k-1}^i(\bar{s}_k^i)^\top\right\} \\ 0 & \mathbb{E}\left\{\bar{s}_k^i(\bar{s}_{k-1}^i)^\top\right\} & \mathbb{E}\left\{\bar{s}_k^i(\bar{s}_k^i)^\top\right\}\end{bmatrix}^{-1} \\ &\times \begin{bmatrix}\mathbb{E}\left\{\bar{\sigma}_{k-2}^i(\bar{\vartheta}_{k-2}^i)^\top\right\} & 0 & 0 \\ 0 & \mathbb{E}\left\{\bar{s}_{k-1}^i(\bar{\vartheta}_{k-1}^i)^\top\right\} & 0 \\ 0 & 0 & \mathbb{E}\left\{\bar{s}_k^i(\bar{\vartheta}_k^i)^\top\right\}\end{bmatrix}. \quad (5.46)\end{aligned}$$

To establish the CRB at every time step k, we need to extract the lower-right submatrices of each term in (5.44), (5.45) and (5.46). However, the presence of inverse of $(\bar{\Gamma}_k^i)^{-1}$ requires additional modification. By utilising the following property of a block matrix,

$$\begin{bmatrix}P & Q \\ Q^\top & R\end{bmatrix}^{-1} = \begin{bmatrix}(P - QR^{-1}Q^\top)^{-1} & -(P - QR^{-1}Q^\top)^{-1}QR^{-1} \\ -R^{-1}Q^\top(P - QR^{-1}Q^\top)^{-1} & (R - Q^\top P^{-1}Q)^{-1}\end{bmatrix},$$

for P, Q, R being matrices with appropriate dimension, the Fisher Information Matrix at time step k can be obtained via

$$\begin{aligned}\bar{F}_k^i &= S_k^i - \begin{bmatrix}0 & (B_k^i)^\top\end{bmatrix}\begin{bmatrix}\mathbb{E}\left\{\bar{\sigma}_{k-2}^i(\bar{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\bar{\sigma}_{k-2}^i(\bar{s}_{k-1}^i)^\top\right\} \\ \mathbb{E}\left\{\bar{s}_{k-1}^i(\bar{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\bar{s}_{k-1}^i(\bar{s}_{k-1}^i)^\top\right\}\end{bmatrix}^{-1}\begin{bmatrix}0 \\ B_k^i\end{bmatrix} \\ &= S_k^i - (B_k^i)^\top(\hat{F}_{k-1}^i + A_k^i)^{-1}B_k^i\end{aligned}$$

where

$$\begin{aligned}S_k^i &= \mathbb{E}\{s_k^i(x_k^i|x_{k-1}^i)s_k^i(x_k^i|x_{k-1}^i)^\top\}, \\ B_k^i &= \mathbb{E}\{s_{k-1}^i(x_k^i|x_{k-1}^i)s_k^i(x_k^i|x_{k-1}^i)^\top\} = (B_k^i)^\top, \\ A_k^i &= \mathbb{E}\{s_{k-1}^i(x_k^i|x_{k-1}^i)s_{k-1}^i(x_k^i|x_{k-1}^i)^\top\}, \text{ and} \\ \hat{F}_k^i &= \mathbb{E}\{s_{k-1}^i(x_{k-1}^i|y_{k-1}^i)s_{k-1}^i(x_{k-1}^i|y_{k-1}^i)^\top\}.\end{aligned}$$

Subsequently, plugging (5.44), (5.45) and (5.46) to (5.43) and extracting the lower-right submatrix of each term in (5.43) yield

$$\bar{C}_k^i + \frac{1}{3}((\bar{F}_k^i)^{-1}R_{\mathrm{m}}(\bar{C}_k^i) + R_{\mathrm{m}}(\bar{C}_k^i)(\bar{F}_k^i)^{-1}) \geqslant (\bar{F}_k^i)^{-1} + \mathcal{O}\left(\mathbb{E}\{K^2\|\bar{\xi}_k^i\|^4\}\right), \quad (5.47)$$

wherein Lemma 5.4 has been applied to the third and fourth term. Define an operator $\Delta : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ and a matrix identity operator $I_d : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$. By using these notations, it follows that the left-hand side of (5.47) could be arranged to

$$(I_d + \Delta)\bar{C}_k^i = \frac{1}{3}((\bar{F}_k^i)^{-1}R_{\mathrm{m}}(\bar{C}_k^i) + R_{\mathrm{m}}(\bar{C}_k^i)(\bar{F}_k^i)^{-1}).$$

Multiplying both sides of (5.47) by $(I_d + \Delta)^{-1}$ and employing the Taylor series $(I_d + \Delta)^{-1} = I_d - \Delta + \Delta^2 - \ldots$ yields

$$\bar{C}_k^i \geqslant (\bar{F}_k^i)^{-1} - \frac{1}{3}((\bar{F}_k^i)^{-1}R_{\mathrm{m}}((\bar{F}_k^i)^{-1}) + R_{\mathrm{m}}((\bar{F}_k^i)^{-1})(\bar{F}_k^i)^{-1}) + \mathcal{O}\left(\mathbb{E}\{K^2\|\bar{\xi}_k^i\|^4\}\right),$$

which completes the proof. □

5.2.2 Local Measurement Update

Let $\hat{x}_k^i \in \mathcal{P}_x$ and $\hat{X}_k^i \in \tilde{\mathcal{P}}_x$ be the posterior estimate or the estimate value of a state x_k^i at time step k, and the augmented posterior estimate up to time

step k, of agent $i \in \mathcal{V}_n$, respectively. The estimate at time step k is evaluated by executing the measurement update of the Bayesian estimation mechanism in (5.5). Our interest in this discussion is to establish the lower bound of the posterior covariance matrix, that is, $\mathbb{E}\{\mathrm{Log}_{\hat{x}_k^i}(x_k^i)\mathrm{Log}_{\hat{x}_k^i}(x_k^i)^\top\} = \mathbb{E}\{\mathrm{Log}_{x_k^i}(\hat{x}_k^i)\mathrm{Log}_{x_k^i}(\hat{x}_k^i)^\top\}$.

An estimator of agent i at time step k, $\hat{x}_k^i : \tilde{\mathcal{S}}_y \to \mathcal{P}_x : Y_k^i \mapsto \hat{x}_k^i(Y_k^i)$, is a mapping which associates the measurement manifolds up to time step k to the current state manifold at time step k based on the posterior distribution of the recursive Bayesian estimator. The augmented estimator can be defined as a mapping $\hat{X}_k^i : \tilde{\mathcal{S}}_y \to \tilde{\mathcal{P}}_x : Y_k^i \mapsto \hat{X}_k^i(Y_k^i)$ from augmented measurement manifold available up to time step k to the augmented state manifold up to time step k.

Define the estimation error at time step k as the difference between the true value x_k^i and the estimate value $\hat{x}_k^i$. Due to non-zero curvature of the manifold, the error requires the logarithm map of two points on manifold $\mathcal{P}_x$:

$$\hat{\xi}_k^i = \mathrm{Log}_{x_k^i}(\hat{x}_k^i) \in \mathrm{T}_{x_k^i}\mathcal{P}_x. \tag{5.48}$$

Let $\hat{\varrho}_k^i \in \mathbb{R}^d$ be an estimation vector of the corresponding estimation error on the state manifold $\mathcal{P}_x$. The entries of the estimate vector can be expressed as

$$\hat{\varrho}_{k,j}^i = \langle \mathrm{Log}_{x_k^i}(\hat{x}_k^i), e_{k,j}^i \rangle. \tag{5.49}$$

Correspondingly, the norm of the estimation error could be calculated using

$$\|\hat{\varrho}_k^i\| = \|\mathrm{Log}_{x_k^i}\hat{x}_k^i\| = \mathrm{dist}(x_k^i, \hat{x}_k^i). \tag{5.50}$$

As the time step increases along the iterations, an expanding parameter composed of the estimation errors up to time step k is then stacked to form the augmented estimation error given by

$$\hat{\zeta}_k^i = \mathrm{Log}_{X_k^i}(\hat{X}_k^i) \in \mathrm{T}_{X_k^i}\tilde{\mathcal{P}}_x. \tag{5.51}$$

Accordingly, let an augmented estimation vector of the state up to time step k be denoted by $\hat{\vartheta}_k^i \in \mathbb{R}^{kd}$, for $d = \dim(\mathcal{P}_x)$. The entries of the augmented estimation vector is

$$\hat{\vartheta}_{k,j}^i = \langle \mathrm{Log}_{X_k^i}(\hat{X}_k^i), \tilde{e}_{k,j}^i \rangle, \tag{5.52}$$

leading to the norm of the augmented estimation error given by

$$\|\hat{\vartheta}_k^i\| = \|\mathrm{Log}_{X_k^i}\hat{X}_k^i\| = \mathrm{dist}(X_k^i, \hat{X}_k^i). \tag{5.53}$$

By employing the estimation vector of agent $i \in \mathcal{V}_n$ at time step k, the covariance matrix of the estimation error at time step k is defined as a symmetric, positive semidefinite matrix $\hat{C}_k^i \in \mathbb{R}^{d\times d}$ written as

$$\hat{C}_k^i = \mathbb{E}\{\hat{\varrho}_k^i(\hat{\varrho}_k^i)^\top\}, \tag{5.54}$$

which is elementarily equivalent to

$$\hat{C}_{k,rs}^i = \mathbb{E}\{\langle \mathrm{Log}_{x_k^i}(\hat{x}_k^i), e_{k,r}^i\rangle \cdot \langle \mathrm{Log}_{x_k^i}(\hat{x}_k^i), e_{k,s}^i\rangle\}.$$

The expanding covariance matrix resulted from augmenting the estimate values up to time step k can be obtained by augmentation of the estimation vector, $\hat{\Omega}_k^i \in \mathbb{R}^{kd\times kd}$. The estimation covariance matrix can be written as

$$\hat{\Omega}_k^i = \mathbb{E}\{\hat{\vartheta}_k^i(\hat{\vartheta}_k^i)^\top\}, \tag{5.55}$$

whose entries are

$$\hat{\Omega}_{k,rs}^i = \mathbb{E}\{\langle \mathrm{Log}_{X_k^i}(\hat{X}_k^i), \tilde{e}_{k,r}^i\rangle \cdot \langle \mathrm{Log}_{X_k^i}(\hat{X}_k^i), \tilde{e}_{k,s}^i\rangle\}.$$

In this discussion, it is assumed that the probability distribution of the estimation processes at time step k_1 at k_2, for $k_1 \neq k_2$, is uncorrelated, i.e., $\mathbb{E}\{\hat{\varrho}_{i\{k_1\}}\hat{\varrho}_{i\{k_2\}}\} = 0$. It follows that $\hat{\Omega}_k^i = \mathrm{diag}(\hat{C}_1^i, \hat{C}_2^i, \ldots, \hat{C}_k^i)$.

To form the definition of posterior FIM, we require the logarithm of PDF of an estimator in Bayesian recursion. Continued from the definition about the natural logarithm of a PDF in the prediction update, let the joint probability distribution of augmented states and previous measurements calculated by agent $i \in \mathcal{V}_n$ be parametrised as $p_i(X_k^i, Y_k^i)$, for $X_k^i \in \tilde{\mathcal{P}}_x$ and $Y_k^i \in \tilde{\mathcal{S}}_y$. Accordingly, the logarithm of the joint distribution is given by

$$\tilde{L}(X_k^i) = \log p_i(X_k^i, Y_k^i). \tag{5.56}$$

The logarithm of the joint PDF updated every time step k, $L : \mathcal{P}_x \to \mathbb{R} : x \mapsto \log p(x, \cdot)$, can be expressed as

$$L(x_k^i) = \log p_i(x_k^i, Y_k^i). \tag{5.57}$$

In order to measure of the steepness of the log-distribution and reflect the sensitivity to changes with respect to the parameter values, we continue the score function explained in the prediction update, that is, $s_k^i : \mathcal{P}_x \to \mathbb{R}^d$. The score vector of the estimate value at time step k, also referred to as posterior score vector, is defined using (5.57) and elementarily given by

$$\hat{s}_{k,r}^i = s_{k,r}^i(x_k^i|Y_k^i) = D_k^i L(x_k^i|Y_k^i)[e_{k,r}^i], \tag{5.58}$$

where $\log p_i(x_k^i, Y_k^i) = \log p_i(x_k^i|Y_k^i) + \log p_i(Y_k^i)$ has been employed and the derivative of the second term vanishes. In the measurement update, by utilising $\tilde{s}_k^i : \tilde{\mathcal{P}}_x \to \mathbb{R}^{kd}$, stacking the posterior score vectors up to time step k gives the augmented posterior score vector based on (5.56), $\hat{\sigma}_k^i \in \mathbb{R}^{kd}$, whose elements are

$$\hat{\sigma}_{k,r}^i = \tilde{s}_{k,r}^i(X_k^i|Y_k^i) = D_k^i \tilde{L}(X_k^i|Y_k^i)[\tilde{e}_{k,r}^i]. \tag{5.59}$$

Neglecting the coordination process, the posterior score function of the measurement process up to time step k is given by $\tilde{s}_k^i(X_k^i|Y_k^i) = \tilde{s}_k^i(Y_k^i|X_k^i) + \tilde{s}_k^i(X_k^i|X_{k-1}^i) + \tilde{s}_k^i(X_{k-1}^i|Y_{k-1}^i)$. It is worth noting that, in this case, the derivative of the denominator of (5.5) equals to zero because it is no longer a function of X_k^i. Based on the definition of the score vectors, it is straightforward to demonstrate that the above posterior score vectors have zero mean, that is, $\mathbb{E}\{\hat{s}_k^i\} = \mathbb{E}\{\hat{\sigma}_k^i\} = 0$.

Posterior FIM at time step k, $\hat{F}_k^i \in \mathbb{R}^{d\times d}$, is a symmetric, positive semidefinite matrix correponding to the estimation score vector (5.58) defined as

$$\hat{F}_k^i = \mathbb{E}\{\hat{s}_k^i(\hat{s}_k^i)^\top\}, \tag{5.60}$$

whose entries are

$$\hat{F}_{k,rs}^i = \mathbb{E}\{D_k^i L(x_k^i|Y_k^i)[e_{k,r}^i] \cdot D_k^i L(x_k^i|Y_k^i)[e_{k,s}^i]\}.$$

Augmenting the FIMs from the first to the k-th iterations forms another symmetric, positive semidefinite matrix $\hat{\Gamma}_k^i \in \mathbb{R}^{kd\times kd}$ expressed as

$$\hat{\Gamma}_k^i = \mathbb{E}\left\{\hat{\sigma}_k^i(\hat{\sigma}_k^i)^\top\right\}, \tag{5.61}$$

and elementarily as

$$\hat{\Gamma}_{k,rs}^i = \mathbb{E}\{D_k^i \tilde{L}(X_k^i|Y_k^i)[\tilde{e}_{k,r}^i] \cdot D_k^i \tilde{L}(X_k^i|Y_k^i)[\tilde{e}_{k,s}^i]\}.$$

It could naturally be observed that the matrix will always be expanding as the time step increases.

To establish the our result in the measurement update, the proofs of the following lemma and theorem require the defined instrumental vectors and matrices: the augmented estimation error in (5.51), estimation vector in (5.52), posterior covariance matrix in (5.55), posterior score vector in (5.59) and posterior FIM in (5.61).

Identical to Lemma 5.4, an additional lemma is required to establish the CRB for the measurement update. The following lemma states the cross-covariance matrix of posterior score and estimate vectors which is adapted from [117] and [118].

Lemma 5.6 (Score-Estimate Covariance Matrix)**.** *Let $\mathcal{P}_x$ denote a d-dimension Riemannian manifold of the state parameter with constant sectional curvature $K \in \mathbb{R}$. Consider a predictor in a recursive Bayesian estimation with estimate error, estimate vector, covariance matrix, and posterior score vector at time step k given in (5.48), (5.49), (5.54) and (5.58), respectively. Then, the score-estimate covariance matrix is given by*

$$\mathbb{E}\{\hat{s}_k^i(\hat{\varrho}_k^i)^\top\} = \mathbb{E}\{\hat{\varrho}_k^i(\hat{s}_k^i)^\top\} = I - \frac{1}{3}R_{\mathrm{m}}(\hat{C}_k^i) + \mathcal{O}\left(\mathbb{E}\{K^2\|\hat{\xi}_k^i\|^4\}\right) \tag{5.62}$$

where $R_{\mathrm{m}}(\hat{C}_k^i) : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ is a linear map given elementarily by

$$R_{\mathrm{m}}(\hat{C}_k^i)_{rs} = \sum_{j,l}\langle R(e_{k,j}^i, e_{k,r}^i)e_{k,s}^i, e_{k,l}^i\rangle \hat{C}_{k,jl}^i. \tag{5.63}$$

Proof. Let us express the expected value augmented estimate error stacked in agent i, for $i \in \mathcal{V}_n$, up to time step k in (5.51) as

$$\mathbb{E}\{\hat{\zeta}_k^i\} = \int_{\tilde{\mathcal{P}}_x} \hat{\zeta}_k^i(X_k^i)p_i(X_k^i|Y_k^i)dX_k^i = 0, \tag{5.64}$$

where $\hat{\zeta}_k^i = \mathrm{Log}_{X_k^i}(\hat{X}_k^i)$.

Consider two arbitrary tangent vector fields on manifold $\tilde{\mathcal{P}}_x$ at point X_k^i, $\tilde{U}, \tilde{V} \in T_{X_k^i}\tilde{\mathcal{P}}_x$ with vector components $\tilde{u}, \tilde{v} \in \mathbb{R}^{kd}$, where $\tilde{U} = \sum_{j=1}^{kd} \tilde{u}_j \tilde{e}_{k,j}^i$ and $\tilde{V} = \sum_{j=1}^{kd} \tilde{v}_j \tilde{e}_{k,j}^i$. Then, taking the covariant derivative with respect to $\tilde{U}$ on both sides of (5.64), followed by taking the Riemannian inner product with $\tilde{U}$ yields

$$\int_{\tilde{\mathcal{P}}_x} \langle \nabla_{\tilde{U}} \hat{\zeta}_k^i(X_k^i)p_i(X_k^i|Y_k^i), \tilde{V}\rangle dX_k^i = 0. \tag{5.65}$$

Recall the product rule of a covariant derivative of a scalar-multiplied vector field, i.e., $\nabla_Z fX = f\nabla_Z X + Df[Z]X$, for a scalar function f and vector field X, Z with appropriate dimension. Hence, with the property of the derivative of logarithm function,

$$D_k^i \tilde{L}(X_k^i|Y_k^i)[\tilde{U}] = \frac{D_k^i p_i(X_k^i|Y_k^i)[\tilde{U}]}{p_i(X_k^i|Y_k^i)},$$

(5.65) could be expanded to

$$\int_{\tilde{\mathcal{P}}_x} \langle D_k^i \tilde{L}(X_k^i|Y_k^i)[\tilde{U}] \cdot \hat{\zeta}_k^i(X_k^i) + \nabla_{\tilde{U}} \hat{\zeta}_k^i(X_k^i), \tilde{V}\rangle p_i(X_k^i|Y_k^i) dX_k^i = 0. \quad (5.66)$$

Since $D_k^i \tilde{L}(X_k^i|Y_k^i)[\tilde{U}] = \langle \nabla_{X_k^i} \tilde{L}(X_k^i|Y_{k-1}^i), \tilde{U}\rangle = \tilde{u}^\top \hat{\sigma}_k^i$ and $\langle \hat{\zeta}_k^i, \tilde{V}\rangle = (\hat{\vartheta}_k^i)^\top \tilde{v}$, from (5.66) one have

$$\tilde{u}^\top \mathbb{E}\{\hat{\sigma}_k^i (\hat{\vartheta}_k^i)^\top\} \tilde{v} = -\mathbb{E}\{\langle \nabla_{\tilde{U}} \hat{\zeta}_k^i, \tilde{V}\rangle\}. \quad (5.67)$$

For every agent $i \in \mathcal{V}_n$ at time step k, let the geodesic connecting $X_k^i \in \tilde{\mathcal{P}}_x$ and $\hat{X}_k^i \in \tilde{\mathcal{P}}_x$ be expressed as

$$\gamma_k^i(t) = \mathrm{Exp}_{\hat{X}_k^i}(t\mathrm{Log}_{\hat{X}_k^i}(X_k^i))$$

where $\gamma_k^i(0) = \hat{X}_k^i$, $\gamma_k^i(1) = X_k^i$ and $\dot{\gamma}_k^i(1) = \hat{\zeta}_k^i$. In the subsequent steps, we utilise the fact shown in [119] that

$$-\nabla_{\tilde{U}} \hat{\zeta}_k^i = \nabla_{\dot{\gamma}} J_k^i(1), \quad (5.68)$$

for J_k^i being a vector field along γ_k^i satisfying

$$\nabla_{\dot{\gamma}}^2 J_k^i(t) + R(J_k^i(t), \dot{\gamma}_k^i(t))\dot{\gamma}_k^i(t) = 0, \quad (5.69)$$

where $\|\dot{\gamma}_k^i(t)\| = \|\hat{\zeta}_k^i\|$ is the geodesic distance between $\hat{X}_k^i$ and X_k^i.

Let the vector field $\tilde{U}$ be constructed as $\tilde{U} = \tilde{U}_\perp + \beta \hat{\zeta}_k^i$, with $\langle \tilde{U}_\perp, \hat{\zeta}_k^i\rangle = 0$ and a scalar β. Let $E(t)$ be a parallel vector field along $\gamma_k^i(t)$ such that, at $t = 1$, $E(1) = \tilde{U}_\perp$. By employing Lemma 5.3, the covariant derivative of the solution of (5.69) at $t = 1$ is

$$\nabla_{\dot{\gamma}} J_k^i(1) = u_1'(1)\tilde{U}_\perp + u_2'(t)\hat{\zeta}_k^i, \quad (5.70)$$

where

$$u_1'(1) = \begin{cases} 1 & \text{if } K = 0, \\ \sqrt{K}\|\hat{\zeta}_k^i\| \cot(\sqrt{K}\|\hat{\zeta}_k^i\|) & \text{if } K > 0, \\ \sqrt{-K}\|\hat{\zeta}_k^i\| \coth(\sqrt{-K}\|\hat{\zeta}_k^i\|) & \text{if } K < 0, \end{cases}$$

and $u_2'(1) = \beta$. Applying Taylor expansion series, i.e., $x\cot(x) = 1 - x^2/3 + \mathcal{O}(x^4)$ and $x\coth(x) = 1 + x^2/3 + \mathcal{O}(x^4)$, to $u_1'(1)$ yields (5.70) rewritten as

$$\nabla_{\dot{\gamma}} J_k^i(1) = \tilde{U} - \frac{1}{3}K\|\hat{\zeta}_k^i\|^2 \tilde{U}_\perp + \mathcal{O}\left(\mathbb{E}\{K^2\|\hat{\zeta}_k^i\|^4\}\right)\tilde{U}_\perp. \tag{5.71}$$

By subtituting (5.71) to (5.67) using (5.68), it follows that

$$\begin{aligned} \tilde{u}^\top \mathbb{E}\{\hat{\sigma}_k^i(\hat{\vartheta}_k^i)^\top\}\tilde{v} &= \mathbb{E}\{\langle \nabla_{\dot{\gamma}} J_k^i(1), \tilde{V}\rangle\} \\ &= \mathbb{E}\{\langle \tilde{U}, \tilde{V}\rangle - \langle \frac{1}{3}K\|\hat{\zeta}_k^i\|^2 \tilde{U}_\perp, \tilde{V}\rangle + \langle \mathcal{O}\left(\mathbb{E}\{K^2\|\hat{\zeta}_k^i\|^4\}\right)\tilde{U}_\perp, \tilde{V}\rangle\} \end{aligned} \tag{5.72}$$

Recall the relationship between Riemannian sectional curvature and curvature tensor, we have $\langle R(\hat{\zeta}_k^i, \tilde{U}_\perp)\tilde{V}, \hat{\zeta}_k^i\rangle = K(\langle \tilde{U}_\perp, \tilde{V}\rangle\langle \hat{\zeta}_k^i, \hat{\zeta}_k^i\rangle - \langle \hat{\zeta}_k^i, \tilde{V}\rangle\langle \tilde{U}_\perp, \hat{\zeta}_k^i\rangle)$. Since $\langle \tilde{U}_\perp, \hat{\zeta}_k^i\rangle = 0$, we have $\langle R(\hat{\zeta}_k^i, \tilde{U}_\perp)\tilde{V}, \hat{\zeta}_k^i\rangle = K(\langle \tilde{U}_\perp, \tilde{V}\rangle\langle \hat{\zeta}_k^i, \hat{\zeta}_k^i\rangle)$. Due to skew-symmetry of curvature tensor, we also have $R(\hat{\zeta}_k^i, \tilde{U}_\perp)\tilde{V} = R(\hat{\zeta}_k^i, \tilde{U})\tilde{V}$. Injecting these facts to the second term of the right-hand side of (5.72) leads to

$$\tilde{u}^\top \mathbb{E}\{\hat{\sigma}_k^i(\hat{\vartheta}_k^i)^\top\}\tilde{v} = \mathbb{E}\{\langle \tilde{U}, \tilde{V}\rangle - \frac{1}{3}\langle R(\hat{\zeta}_k^i, \tilde{U})\tilde{V}, \hat{\zeta}_k^i\rangle + \mathcal{O}\left(\mathbb{E}\{K^2\|\hat{\zeta}_k^i\|^4\}\right)\langle \tilde{U}_\perp, \tilde{V}\rangle\}. \tag{5.73}$$

Consequently, the augmented score-estimate covariance matrix can be expressed as

$$\mathbb{E}\{\hat{\sigma}_k^i(\hat{\vartheta}_k^i)^\top\} = I - \frac{1}{3}\tilde{R}_{\mathrm{m}}\left(\hat{\Omega}_k^i\right) + \mathcal{O}\left(\mathbb{E}\{K^2\|\hat{\zeta}_k^i\|^4\}\right), \tag{5.74}$$

where $\tilde{R}_{\mathrm{m}}(\hat{\Omega}_k^i) : \mathbb{R}^{kd\times kd} \to \mathbb{R}^{kd\times kd}$ is given by

$$\begin{aligned} \tilde{R}_{\mathrm{m}}(\hat{\Omega}_k^i)_{rs} &= \mathbb{E}\{\langle R(\hat{\zeta}_k^i, \tilde{e}_{k,r}^i)\tilde{e}_{k,s}^i, \hat{\zeta}_k^i\rangle\} \\ &= \sum_{j,l}\langle R(\tilde{e}_{k,j}^i, \tilde{e}_{k,r}^i)\tilde{e}_{k,s}^i, \tilde{e}_{k,l}^i\rangle\hat{\Omega}_{k,jl}^i. \end{aligned}$$

Let us explicitly arrange the augmented estimate vector at agent i in (5.52) stacked up to time step k as $\hat{\vartheta}_k^i = [(\hat{\vartheta}_{k-1}^i)^\top, (\hat{\varrho}_k^i)^\top]^\top$. Likewise, the augmented

score vector in (5.59) is decomposed to $\hat{\sigma}_k^i = [(\hat{\sigma}_{k-1}^i)^\top, (\hat{s}_k^i)^\top]^\top$. By utilising these decompositions, the left-hand side of (5.74) can be stacked as

$$\mathbb{E}\{\hat{\sigma}_k^i(\hat{\vartheta}_k^i)^\top\} = \mathbb{E}\left\{\begin{bmatrix}\hat{\sigma}_{k-1}^i(\hat{\vartheta}_{k-1}^i)^\top & 0\\ 0 & \hat{s}_k^i(\hat{\varrho}_k^i)^\top\end{bmatrix}\right\}. \tag{5.75}$$

Let the estimate error up to time step k be arranged as $\hat{\zeta}_k^i = (\hat{\zeta}_{k-1}^i, \hat{\xi}_k^i)$ and the orthogonal basis as $\tilde{e}_{k,r}^i = (\tilde{e}_{i\{k-1\},r}, e_{k,r}^i)$. Applying them to the second term of the right-hand side of (5.73) leads to $\tilde{R}_{\mathrm{m}}(\hat{\Omega}_k^i)$ structured as

$$\tilde{R}_{\mathrm{m}}(\hat{\Omega}_k^i) = \begin{bmatrix}\tilde{R}_{\mathrm{m}}(\hat{\Omega}_{k-1}^i) & 0\\ 0 & R_{\mathrm{m}}(\hat{C}_k^i)\end{bmatrix}, \tag{5.76}$$

with R_{m} defined in (5.63). Notice that the implication of the independency of information at time step $k-1$ and k has been employed to both (5.75) and (5.76), i.e., the expected value of the cross-covariance between two variables at time step k and $k-1$ equals to zero.

From (5.75), it can be observed that the state-estimate covariance matrix at time step k can be obtained by extracting the lower-right entries of matrices $\mathbb{E}\{\hat{\sigma}_k^i(\hat{\vartheta}_k^i)^\top\}$ and $\tilde{R}_{\mathrm{m}}(\hat{\Omega}_k^i)$. Finally, plugging (5.75) and (5.76) into (5.74), then taking the lower-right sub-matrix of all terms yields

$$\mathbb{E}\{\hat{s}_k^i(\hat{\varrho}_k^i)^\top\} = \mathbb{E}\{\hat{\varrho}_k^i(\hat{s}_k^i)^\top\} = I - \frac{1}{3}R_{\mathrm{m}}(\hat{C}_k^i) + \mathcal{O}\left(\mathbb{E}\{K^2\|\hat{\xi}_k^i\|^4\}\right). \tag{5.77}$$

This completes the proof. □

The following theorem states our second result about the intrinsic CRB for local Bayesian estimation update whose arguments are adapted from [117], and [118] and [81].

Theorem 5.7 (Intrinsic Posterior CRB for Bayesian Estimator)**.** *Consider a group of n agents connected via a graph $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$. Let $\mathcal{P}_x$ and $\mathcal{S}_y$ be Riemannian state and measurement manifolds, respectively. For each agent $i \in \mathcal{V}_n$, let $\hat{x}_k^i \in \mathcal{P}_x$ be an estimator at time step k following the measurement step of recursive Bayesian estimator in (5.5). Let the orthogonal basis of the estimator be given by $e_k^i = (e_{k,1}^i, e_{k,2}^i, \ldots, e_{ik,d})$, for $d = \dim(\mathcal{P}_x)$. Let the estimator be equipped with the estimate error, estimate vector and posterior score vector at time step*

k given by (5.48), (5.49), (5.58), respectively. Then, with posterior covariance matrix in (5.54) and FIM in (5.60), the posterior covariance matrix satisfies the following inequality:

$$\hat{C}_k^i \geqslant (\hat{F}_k^i)^{-1} - \frac{1}{3}\left((\hat{F}_k^i)^{-1} R_{\mathrm{m}}((\hat{F}_k^i)^{-1}) + R_{\mathrm{m}}((\hat{F}_k^i)^{-1})(\hat{F}_k^i)^{-1}\right) + \mathcal{O}\left(\mathbb{E}\{K^2\|\hat{\xi}_k^i\|^4\}\right), \tag{5.78}$$

where $R_{\mathrm{m}}((\hat{F}_k^i)^{-1})_{rs} = \sum_{j,l}\langle R(e_{k,j}^i, e_{k,r}^i)e_{k,s}^i, e_{k,l}^i\rangle((\hat{F}_k^i)^{-1})_{jl}$ *and*

$$\hat{F}_k^i = G_k^i + S_k^i - (B_k^i)^\top(\hat{F}_{k-1}^i + A_k^i)^{-1}B_k^i \tag{5.79}$$

with

$$\begin{aligned}
G_k^i &= \mathbb{E}\{s_k^i(y_k^i|x_k^i)s_k^i(y_k^i|x_k^i)^\top\},\\
S_k^i &= \mathbb{E}\{s_k^i(x_k^i|x_{k-1}^i)s_k^i(x_k^i|x_{k-1}^i)^\top\},\\
B_k^i &= \mathbb{E}\{s_{k-1}^i(x_k^i|x_{k-1}^i)s_k^i(x_k^i|x_{k-1}^i)^\top\} = (B_k^i)^\top,\\
A_k^i &= \mathbb{E}\{s_{k-1}^i(x_k^i|x_{k-1}^i)s_{k-1}^i(x_k^i|x_{k-1}^i)^\top\},\ \textit{and}\\
\hat{F}_k^i &= \mathbb{E}\{s_{k-1}^i(x_{k-1}^i|y_{k-1}^i)s_{k-1}^i(x_{k-1}^i|y_{k-1}^i)^\top\}.
\end{aligned}$$

Proof. For agent $i \in \mathcal{V}_n$, consider a random vector augmented up to time step k be given by $\hat{\nu}_k^i \in \mathbb{R}^{kd}$, where $d = \dim(\mathcal{P}_x)$, such that $\mathbb{E}\{\hat{\nu}_k^i(\hat{\nu}_k^i)^\top\} \geqslant 0$ holds. This vector can be chosen as

$$\hat{\nu}_k^i = \hat{\vartheta}_k^i - (\hat{\Gamma}_k^i)^{-1}\hat{\sigma}_k^i,$$

whose expected value of $\hat{\nu}_k^i$ is zero. Accordingly, consider a covariance matrix $\mathbb{E}\{\hat{\nu}_k^i(\hat{\nu}_k^i)^\top\}$:

$$\begin{aligned}
\mathbb{E}\left\{\hat{\nu}_k^i(\hat{\nu}_k^i)^\top\right\} = \mathbb{E}\left\{\hat{\vartheta}_k^i(\hat{\vartheta}_k^i)^\top\right\} + (\hat{\Gamma}_k^i)^{-1}\mathbb{E}\left\{\hat{\sigma}_k^i(\hat{\sigma}_k^i)^\top\right\}(\hat{\Gamma}_k^i)^{-1}\\
- (\hat{\Gamma}_k^i)^{-1}\mathbb{E}\left\{\hat{\sigma}_k^i(\hat{\vartheta}_k^i)^\top\right\} - \mathbb{E}\left\{\hat{\vartheta}_k^i(\hat{\sigma}_k^i)^\top\right\}(\hat{\Gamma}_k^i)^{-1} \geqslant 0.
\end{aligned} \tag{5.80}$$

From the defined covariance matrix, let us inspect the element of its term by firstly noting that $\hat{\vartheta}_k^i = [(\hat{\vartheta}_{k-2}^i)^\top, (\hat{\varrho}_{k-1}^i)^\top, (\hat{\varrho}_k^i)^\top]^\top$, $\hat{\sigma}_k^i = [(\hat{\sigma}_{k-2}^i)^\top, (\hat{s}_{k-1}^i)^\top, (\hat{s}_k^i)^\top]^\top$, $\mathbb{E}\{\hat{\sigma}_k^i(\hat{\sigma}_k^i)^\top\} = \hat{\Gamma}_k^i$ and $\mathbb{E}\{\hat{\vartheta}_k^i(\hat{\vartheta}_k^i)^\top\} = \hat{\Omega}_k^i$. It follows that

$$\hat{\Omega}_k^i = \begin{bmatrix}
\mathbb{E}\left\{\hat{\vartheta}_{k-2}^i(\hat{\vartheta}_{k-2}^i)^\top\right\} & 0 & 0\\
0 & \mathbb{E}\left\{\hat{\varrho}_{k-1}^i(\hat{\varrho}_{k-1}^i)^\top\right\} & 0\\
0 & 0 & \mathbb{E}\left\{\hat{\varrho}_k^i(\hat{\varrho}_k^i)^\top\right\}
\end{bmatrix}, \tag{5.81}$$

$$(\hat{\Gamma}_k^i)^{-1} = \begin{bmatrix} \mathbb{E}\left\{\hat{\sigma}_{k-2}^i(\hat{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\hat{\sigma}_{k-2}^i(\hat{s}_{k-1}^i)^\top\right\} & 0 \\ \mathbb{E}\left\{\hat{s}_{k-1}^i(\hat{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\hat{s}_{k-1}^i(\hat{s}_{k-1}^i)^\top\right\} & \mathbb{E}\left\{\hat{s}_{k-1}^i(\hat{s}_k^i)^\top\right\} \\ 0 & \mathbb{E}\left\{\hat{s}_k^i(\hat{s}_{k-1}^i)^\top\right\} & \mathbb{E}\left\{\hat{s}_k^i(\hat{s}_k^i)^\top\right\} \end{bmatrix}^{-1}, \tag{5.82}$$

and

$$\begin{aligned}(\hat{\Gamma}_k^i)^{-1}\mathbb{E}\left\{\hat{\sigma}_k^i(\hat{\vartheta}_k^i)^\top\right\} = &\begin{bmatrix} \mathbb{E}\left\{\hat{\sigma}_{k-2}^i(\hat{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\hat{\sigma}_{k-2}^i(\hat{s}_{k-1}^i)^\top\right\} & 0 \\ \mathbb{E}\left\{\hat{s}_{k-1}^i(\hat{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\hat{s}_{k-1}^i(\hat{s}_{k-1}^i)^\top\right\} & \mathbb{E}\left\{\hat{s}_{k-1}^i(\hat{s}_k^i)^\top\right\} \\ 0 & \mathbb{E}\left\{\hat{s}_k^i(\hat{s}_{k-1}^i)^\top\right\} & \mathbb{E}\left\{\hat{s}_k^i(\hat{s}_k^i)^\top\right\} \end{bmatrix}^{-1} \\ &\times \begin{bmatrix} \mathbb{E}\left\{\hat{\sigma}_{k-2}^i(\hat{\vartheta}_{k-2}^i)^\top\right\} & 0 & 0 \\ 0 & \mathbb{E}\left\{\hat{s}_{k-1}^i(\hat{\vartheta}_{k-1}^i)^\top\right\} & 0 \\ 0 & 0 & \mathbb{E}\left\{\hat{s}_k^i(\hat{\vartheta}_k^i)^\top\right\} \end{bmatrix}.\end{aligned} \tag{5.83}$$

To establish the posterior CRB at every time step k, we need to extract the lower-right submatrices of each term in (5.81), (5.82) and (5.83). However, the presence of inverse of $(\hat{\Gamma}_k^i)^{-1}$ requires additional modification. By utilising the following property of a block matrix,

$$\begin{bmatrix} P & Q \\ Q^\top & R \end{bmatrix}^{-1} = \begin{bmatrix} (P - QR^{-1}Q^\top)^{-1} & -(P - QR^{-1}Q^\top)^{-1}QR^{-1} \\ -R^{-1}Q^\top(P - QR^{-1}Q^\top)^{-1} & (R - Q^\top P^{-1}Q)^{-1} \end{bmatrix},$$

for P, Q, R being matrices with appropriate dimension, the Fisher Information Matrix at time step k can be obtained via

$$\begin{aligned}\hat{F}_k^i &= S_k^i - \begin{bmatrix} 0 & (B_k^i)^\top \end{bmatrix} \begin{bmatrix} \mathbb{E}\left\{\hat{\sigma}_{k-2}^i(\hat{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\hat{\sigma}_{k-2}^i(\hat{s}_{k-1}^i)^\top\right\} \\ \mathbb{E}\left\{\hat{s}_{k-1}^i(\hat{\sigma}_{k-2}^i)^\top\right\} & \mathbb{E}\left\{\hat{s}_{k-1}^i(\hat{s}_{k-1}^i)^\top\right\} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ B_k^i \end{bmatrix} \\ &= G_k^i + S_k^i - (B_k^i)^\top(\hat{F}_{k-1}^i + A_k^i)^{-1}B_k^i\end{aligned}$$

where

$$\begin{aligned} G_k^i &= \mathbb{E}\{s_k^i(y_k^i|x_k^i)s_k^i(y_k^i|x_k^i)^\top\}, \\ S_k^i &= \mathbb{E}\{s_k^i(x_k^i|x_{k-1}^i)s_k^i(x_k^i|x_{k-1}^i)^\top\}, \\ B_k^i &= \mathbb{E}\{s_{k-1}^i(x_k^i|x_{k-1}^i)s_k^i(x_k^i|x_{k-1}^i)^\top\} = (B_k^i)^\top, \\ A_k^i &= \mathbb{E}\{s_{k-1}^i(x_k^i|x_{k-1}^i)s_{k-1}^i(x_k^i|x_{k-1}^i)^\top\}, \text{ and} \\ \hat{F}_k^i &= \mathbb{E}\{s_{k-1}^i(x_{k-1}^i|y_{k-1}^i)s_{k-1}^i(x_{k-1}^i|y_{k-1}^i)^\top\}. \end{aligned}$$

Subsequently, plugging (5.81), (5.82) and (5.83) to (5.80) and extracting the lower-right submatrix of each term in (5.80) yield

$$\hat{C}_k^i + \frac{1}{3}((\hat{F}_k^i)^{-1} R_{\mathrm{m}}(\hat{C}_k^i) + R_{\mathrm{m}}(\hat{C}_k^i)(\hat{F}_k^i)^{-1}) \geqslant (\hat{F}_k^i)^{-1} + \mathcal{O}\left(\mathbb{E}\{K^2 \|\hat{\xi}_k^i\|^4\}\right), \quad (5.84)$$

wherein Lemma 5.6 has been applied to the third and fourth term. Define an operator $\Delta : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ and a matrix identity operator $I_d : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$. By using these notations, it follows that the left-hand side of (5.84) could be arranged to

$$(I_d + \Delta)\hat{C}_k^i = \frac{1}{3}((\hat{F}_k^i)^{-1} R_{\mathrm{m}}(\hat{C}_k^i) + R_{\mathrm{m}}(\hat{C}_k^i)(\hat{F}_k^i)^{-1}).$$

Multiplying both sides of (5.84) by $(I_d + \Delta)^{-1}$ and employing the Taylor series $(I_d + \Delta)^{-1} = I_d - \Delta + \Delta^2 - \ldots$ yields

$$\hat{C}_k^i \geqslant (\hat{F}_k^i)^{-1} - \frac{1}{3}((\hat{F}_k^i)^{-1} R_{\mathrm{m}}((\hat{F}_k^i)^{-1}) + R_{\mathrm{m}}((\hat{F}_k^i)^{-1})(\hat{F}_k^i)^{-1}) + \mathcal{O}\left(\mathbb{E}\{K^2 \|\hat{\xi}_k^i\|^4\}\right)$$

which completes the proof. □

The following corollary states a more compact form of the intrinsic posterior CRB presented in Theorem 5.7.

Corollary 5.8. *By invoking the FIM in the intrinsic prior CRB,*

$$\bar{F}_k^i = S_k^i - (B_k^i)^\top (\hat{F}_{k-1}^i + A_k^i)^{-1} B_k^i,$$

the posterior CRB could equivalently be expressed as

$$\hat{C}_k^i \geqslant (\hat{F}_k^i)^{-1} - \frac{1}{3}\left((\hat{F}_k^i)^{-1} R_{\mathrm{m}}((\hat{F}_k^i)^{-1}) + R_{\mathrm{m}}((\hat{F}_k^i)^{-1})(\hat{F}_k^i)^{-1}\right) + \mathcal{O}\left(\mathbb{E}\{K^2 \|\hat{\xi}_k^i\|^4\}\right), \quad (5.85)$$

where

$$\hat{F}_k^i = \bar{F}_k^i + G_k^i. \quad (5.86)$$

The above discussion have demonstrated the intrinsic CRBs for both local prediction and measurement of Bayesian estimator. The established CRBs can be utilised to calculate the lowest value of MSE of a Bayesian estimator. For simplicity, assume that there is only $n = 1$ agent. By removing the subscripted i

and the higher-order term of the posterior CRB, the minimum value of the MSE a Bayesian estimator could achieve at every time step k could be calculated via

$$\begin{aligned}\mathrm{MSE}(x_k, \hat{x}_k) &= \mathrm{tr}(\hat{C}_k) \\ &\geqslant \mathrm{tr}(\hat{F}_k^{-1}) - \frac{1}{3}\mathrm{tr}\left(\hat{F}_k^{-1} R_{\mathrm{m}}(\hat{F}_k^{-1}) + R_{\mathrm{m}}(\hat{F}_k^{-1})\hat{F}_k^{-1}\right).\end{aligned}$$

5.2.3 Coordination Update

Based on the scenario of distributed Bayesian estimator presented in Section 5.1, the corresponding intrinsic CRB is sequenced as follows. Firstly, each agent computes the prior CRB followed by computing the posterior CRB locally. Secondly, the distributed posterior CRB is calculated using the local posterior CRBs from neighbouring agents.

The intrinsic posterior CRB for the distributed Bayesian estimation is stated in the following theorem.

Theorem 5.9 (Intrinsic Posterior CRB for Distributed Bayesian Estimator)**.** *Let $\mathcal{P}_x$ and $\mathcal{S}_y$, respectively, be Riemannian state and measurement manifolds with $p = \dim(\mathcal{P}_x)$. Let a group of n agents be connected via a graph $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ with doubly-stochastic primitive Perron matrix $\Pi = [\pi_{ij}] \in \mathbb{R}^{nd\times nd}$. For each agent $i \in \mathcal{V}_n$, let $\hat{x}_k^i \in \mathcal{P}_x$ be an estimator at time step k following the measurement step of recursive Bayesian estimator in (5.5). Let the orthogonal basis of the estimator be given by $e_k^i = (e_{k,1}^i, e_{k,2}^i, \ldots, e_{ik,d})$. Let the estimator be equipped with the estimate error, estimate vector and posterior score vector at time step k given by (5.48), (5.49), (5.58), respectively. Then, with posterior covariance matrix in (5.54) and FIM in (5.60), the posterior covariance matrix satisfies the following inequality:*

$$\tilde{C}_k^i \geqslant (\tilde{F}_k^i)^{-1} - \frac{1}{3}\left((\tilde{F}_k^i)^{-1} R_{\mathrm{m}}((\tilde{F}_k^i)^{-1}) + R_{\mathrm{m}}((\tilde{F}_k^i)^{-1})(\tilde{F}_k^i)^{-1}\right) + \mathcal{O}\left(\mathbb{E}\{K^2\|\hat{\xi}_k^i\|^4\}\right), \tag{5.87}$$

where $R_{\mathrm{m}}((\tilde{F}_k^i)^{-1})_{rs} = \sum_{j,l}\langle R(e_{k,j}^i, e_{k,r}^i)e_{k,s}^i, e_{k,l}^i\rangle((\tilde{F}_k^i)^{-1})_{jl}$. The distributed posterior FIM is given by

$$\tilde{F}_k^i = \sum_{j=1}^{n} \pi_{ij}\hat{F}_{jk}, \ \textit{for } j \in \mathcal{N}_i. \tag{5.88}$$

Proof. The proof of this theorem follows similar arguments in the proof of Theorem 5.7 with slight modification by employing the primitive Perron matrix to collect the posterior FIMs from neighbouring agents. □

Remark 5.10. The primitivity of the Perron matrix Π implies that $\lim_{k\to\infty}\Pi^k = vw^\top$ with $\Pi v = v$ and $w^\top\Pi = w^\top$. Due to doubly-stochastic assumption, we have $v = \mathbf{1}$ and $w = (1/n)\mathbf{1}$. It follows that the value of prior FIM as $k \to \infty$ is given by $\lim_{k\to\infty}\Pi^k\bar{F}_{i0} = (1/n)\mathbf{1}^\top\bar{F}_{i0}$, i.e., average of the initial value of the FIMs.

In this n-agent estimator scenario, by neglecting the higher-order term of the CRB, the MSE of the distributed Bayesian estimator is lower-bounded by

$$\begin{aligned}\mathrm{MSE}(x_k,\hat{x}_k) &= \sum_{i=1}^{n}\mathrm{tr}(\tilde{C}_k^i)\\ &\geqslant \sum_{i=1}^{n}\mathrm{tr}((\tilde{F}_k^i)^{-1}) - \frac{1}{3}\sum_{i=1}^{n}\mathrm{tr}\left((\tilde{F}_k^i)^{-1}R_{\mathrm{m}}((\tilde{F}_k^i)^{-1}) + R_{\mathrm{m}}((\tilde{F}_k^i)^{-1})(\tilde{F}_k^i)^{-1}\right).\end{aligned}$$

5.3 Distributed Riemannian Kalman Filter

In this section, we propose a class of Bayesian estimator to solve the distributed estimation problem on Riemannian manifolds whose system dynamic and measurement model is defined in (5.1) and (5.2), respectively. By assuming that the probability density function (PDF) of the noises to being Gaussian, we can design a distributed Riemannian Kalman filter based on the distributed UKF designed in Chapter 4 with some adjustments to accommodate the geodesic curvature.

Before designing the distributed filter, we require several assumptions regarding the process and measurement PDFs, communication topology of the network, and the Riemannian manifolds of the process and measurement:

1. the Riemannian manifolds of the process and measurement are geodesically complete,

2. all points are located inside the maximal domain where the exponential mapping is still a diffeomorphism,

3. the system has sufficiently large SNR that allows the state and measurement to stay inside the maximal geodesic all points such that there exists only one Fréchet mean,

4. the communication noise is zero and the graph topology of the network can be modelled via Laplacian matrix,

5. the process and measurement noises are Gaussian and uncorrelated, while the process and measurement data are jointly Gaussian.

These assumptions allow us to generalise the distributed UKF developed in Chapter 4 without communication data by utilising the exponential and logarithm mappings on manifolds to accommodate the non-zero sectional curvature. Since the communication noise is assumed to be zero, the distributed UKF developed in Euclidean systems is extendable to Riemannian manifolds with several modification to accommodate curved space.

5.3.1 Prediction Update

The means of unscented transformation utilised in the UKF is modified in the prediction update to calculate the prior estimate and measurement values together with their covariances.

For each agent $i \in \mathcal{V}_n$, define $\tilde{x}^i_{k,\text{aug}} = [(\tilde{x}^i_k)^\top \ (v^i_k)^\top \ (w^i_k)^\top]^\top$ as a random variable augmenting the posterior estimate, and the process and measurement noises; while $\tilde{P}^i_{xx,k,\text{aug}} = \text{diag}(\tilde{P}^i_{xx,k}, Q^i_k, R^i_k)$ as an augmented covariances containing the covariance of the posterior estimate, and the covariance matrices of the process and measurement noises at time step k. Accordingly, the sigma vector at time step $k-1$ is defined as $\mathcal{X}^i_{k-1} = [(\mathcal{X}^i_{x,k-1})^\top \ (\mathcal{X}^i_{v,k-1})^\top \ (\mathcal{X}^i_{w,k-1})^\top]^\top \in \mathbb{R}^{a\times(2a+1)}$, $a = 2d+q$ whose entries are detailed as

$$\mathcal{X}^i_{k-1,l} = \tilde{x}^i_{k-1,\text{aug}}, \text{ for } l = 0, \tag{5.89a}$$

$$\mathcal{X}^i_{k-1,l} = \text{Exp}_{\tilde{x}^i_{k-1,\text{aug}}}\Big(\sqrt{(a+b)\tilde{P}^i_{xx,k-1,\text{aug}}}\Big)_l, \text{ for } 1 \leqslant l \leqslant a, \tag{5.89b}$$

$$\mathcal{X}^i_{k-1,l} = \text{Exp}_{\tilde{x}^i_{k-1,\text{aug}}}\Big(-\sqrt{(a+b)\tilde{P}^i_{xx,k-1,\text{aug}}}\Big)_l, \text{ for } a+1 \leqslant l \leqslant 2a, \tag{5.89c}$$

where we already employed the exponential mapping on the manifolds. We also use $b = \varrho_1^2(a+\varrho_2) - a$ as a scaling parameter, where ϱ_1 and ϱ_2 are some parameters related to the spread of sigma points around $\tilde{x}^i_{k-1,\text{aug}}$ [113].

The sigma vectors are subsequently transformed through the process function as defined in (5.1). Every l-th column of $\mathcal{X}^i_{k-1}$, with $l \in \{0,1,2,\ldots,2a\}$, is

transformed via the process function such that the prior sigma vector of the state can be written as

$$\bar{\mathcal{X}}^i_{k,l} = f(\mathcal{X}^i_{x,k-1,l},\ \mathcal{X}^i_{v,k-1,l}). \tag{5.90}$$

Accordingly, the prior estimate $\bar{x}^i_k$ and covariance $\bar{P}^i_{xx,k}$ can be calculated using the approximated weighted mean and covariance of the sigma points in the form of

$$\bar{x}^i_k = \arg\min_{\alpha\in\mathcal{P}_x} \sum_{l=0}^{2a} w^m_l \mathrm{dist}^2(\alpha, \bar{\mathcal{X}}^i_{k,l}), \tag{5.91}$$

$$\bar{P}^i_{xx,k} = \sum_{l=0}^{2a} w^c_l \left(\mathrm{Log}_{\bar{x}^i_k} \bar{\mathcal{X}}^i_{k,l}\right)\left(\mathrm{Log}_{\bar{x}^i_k} \bar{\mathcal{X}}^i_{k,l}\right)^\top, \tag{5.92}$$

where the weights are given by

$$\begin{aligned} w^m_0 &= b/\kappa, \\ w^c_0 &= b/\kappa + (1 - \varrho^2_1 + \varrho_2), \\ w^m_l &= W^c_l = 1/(2\kappa),\ 0 < l \leqslant 2a, \end{aligned}$$

satisfying $\sum_{l=0}^{2a} w^m_l = 1$ and $\sum_{l=0}^{2a} w^c_l = 1$. The optimisation in manifold is presented because the mean value on manifold, or simply the Riemannian center of mass, is generally obtained via an optimisation process. The mean value could be obtained via, for example, Riemannian gradient-descent as analysed in [120]. It is worth noticing that the prior estimate value is unique because of the maximal domain assumption.

In the prediction of the measurement data, the sigma vector of the state in (5.90) is also mapped through the nonlinear measurement function (5.2) such that the sigma vector of the measurement is formulated as

$$\bar{\mathcal{Y}}^i_{k,l} = h(\bar{\mathcal{X}}^i_{k,l},\ \mathcal{X}^i_{w,k-1,l}). \tag{5.94}$$

Accordingly, similar to the prediction of the state and its covariance, the prior estimate of the measurement and its covariance, $\bar{y}^i_k$ and $\bar{P}^i_{yy,k}$, are also the mean and covariance calculated by solving the Riemannian optimisation problem, i.e.,

$$\bar{y}^i_k = \arg\min_{\alpha\in\mathcal{S}_y} \sum_{l=0}^{2a} w^m_l \mathrm{dist}^2(\alpha, \bar{\mathcal{Y}}^i_{k,l}), \tag{5.95}$$

$$\bar{P}^i_{yy,k} = \sum_{l=0}^{2a} w^c_l \left(\mathrm{Log}_{\bar{y}^i_k} \bar{\mathcal{Y}}^i_{k,l}\right)\left(\mathrm{Log}_{\bar{y}^i_k} \bar{\mathcal{Y}}^i_{k,l}\right)^\top \tag{5.96}$$

respectively. Due to the maximal domain assumption, we will also have a unique prior estimate of the measurement.

The prior cross-covariance matrices between the state estimate and measurement data also be formulated utilising the sigma vectors in the form of

$$\bar{P}^i_{xy,k} = \sum_{l=0}^{2a} w_l^c \left(\mathrm{Log}_{\bar{x}^i_k} \bar{\mathcal{X}}^i_{k,l}\right) \left(\mathrm{Log}_{\bar{y}^i_k} \bar{\mathcal{Y}}^i_{k,l}\right)^\top. \tag{5.97}$$

Notice that the prior state and measurement covariances are symmetric matrices, i.e., $\bar{P}^i_{xx,k} = (\bar{P}^i_{xx,k})^\top$ and $\bar{P}^i_{yy,k} = (\bar{P}^i_{yy,k})^\top$.

5.3.2 Measurement and Coordination Updates

Before presenting the posterior estimation update, we present a Riemannian consensus protocol utilised in the coordination process.

To combine the states from neighbouring agents, the traditional consensus protocol originated for deterministic systems has widely been utilised in distributed filtering and estimation [12, 13]. The following proposition presents the extension of the traditional consensus protocol to Riemannian systems and provides alternative stability proof of the Riemannian consensus protocol.

Proposition 5.11. *Consider a group of n agents connected via a graph $\mathcal{G}_n = \{\mathcal{V}_n, \mathcal{E}_n\}$. Let a Riemannian manifold $\mathcal{M}$ whose sectional curvature is upper-bounded by Δ with injectivity radius $r_\Delta = \frac{1}{2}\min\{\mathrm{inj}\mathcal{M}, \frac{\pi}{\sqrt{\Delta}}\}$. Let the state of the agents be stacked as $x_k = [(x^1_k)^\top, \ldots, (x^n_k)^\top]^\top$ and all states $\{x^i_k \in \mathcal{M} | i = \mathcal{V}_n\} \subset \bar{\mathcal{B}}(y, r)$ with $r < r_\Delta$. Let the Riemannian consensus protocol be given by*

$$x^i_{k+1} = \mathrm{Exp}_{x^i_k}\Big(\varepsilon \sum_{j=1}^{n} a_{ij} \mathrm{Log}_{x^i_k}(x^j_k)\Big), \tag{5.98}$$

where $A = [a_{ij}]$ denotes the adjacency matrix of the graph $\mathcal{G}_n$, $\varepsilon \in (0, \frac{2}{H_\Delta \deg(\mathcal{G}_n)})$ and H_Δ being the Hessian of distance function on manifold $\mathcal{M}$. Then, the consensus of x^i_k, that is $x^1_k = \ldots = x^n_k = x^\star$, is achieved asymptotically.

Proof. Let a twice-differentiable Lyapunov function candidate at $x^\star \in T_{x^i_k}\mathcal{M}$ given by

$$V(x_k) = \frac{1}{2} \sum_{i=1}^{n} \|\mathrm{Log}_{x^\star}(x^i_k)\|^2. \tag{5.99}$$

Consider the consensus protocol of agent i as the geodesic connecting the state x_k^i to x_{k+1}^i, for all $i \in \mathcal{V}_n$. Then, applying Taylor expansion of (5.99) at $k+1$ leads to difference equation given by

$$\begin{aligned} V(x_{k+1}) - V(x_k) =& \varepsilon \sum_{i=1}^{n} \langle \mathrm{Log}_{x^\star}(x_k^i), \sum_{j=1}^{n} a_{ij} \mathrm{Log}_{x_k^i}(x_k^j) \rangle \\ &+ \frac{1}{2} \nabla^2 V(x_k) \varepsilon^2 \sum_{i=1}^{n} \langle \sum_{j=1}^{n} a_{ij} \mathrm{Log}_{x_k^i}(x_k^j), \sum_{j=1}^{n} a_{ij} \mathrm{Log}_{x_k^i}(x_k^j) \rangle. \end{aligned} \tag{5.100}$$

Before continuing the proof, let us consider the consensus protocol in (5.98) at the tangent space of $x^\star$:

$$\mathrm{Log}_{x^\star}(x_{k+1}^i) = \varepsilon \sum_{j=1}^{n} a_{ij} (\mathrm{Log}_{x^\star}(x_k^j) - \mathrm{Log}_{x^\star}(x_k^i)). \tag{5.101}$$

Plugging this to (5.100) leads to the first term of the right-hand side expressed as

$$\begin{aligned} &\varepsilon \sum_{i=1}^{n} \langle \mathrm{Log}_{x^\star}(x_k^i), \sum_{j=1}^{n} a_{ij} \mathrm{Log}_{x_k^i}(x_k^j) \rangle \\ &\qquad = \varepsilon \sum_{i=1}^{n} \langle \mathrm{Log}_{x^\star}(x_k^i), \sum_{j=1}^{n} a_{ij} (\mathrm{Log}_{x^\star}(x_k^j) - \mathrm{Log}_{x^\star}(x_k^i)) \rangle. \end{aligned} \tag{5.102}$$

By utilising the Laplacian matrix of the graph $\hat{\mathcal{L}}_n = \mathcal{L}_n \otimes I_d$, we have

$$\varepsilon \sum_{i=1}^{n} \langle \mathrm{Log}_{x^\star}(x_k^i), \sum_{j=1}^{n} a_{ij} \mathrm{Log}_{x_k^i}(x_k^j) \rangle = -\varepsilon \langle \mathrm{Log}_{x^\star}(x_k), \hat{\mathcal{L}}_n \mathrm{Log}_{x^\star}(x_k) \rangle. \tag{5.103}$$

Similarly, plugging (5.101) to (5.100) and utilising the Laplacian matrix yield the second term of the right-hand side of (5.100) written as

$$\begin{aligned} &\frac{1}{2} \nabla^2 V(x_k) \varepsilon^2 \sum_{i=1}^{n} \Big\langle \sum_{j=1}^{n} a_{ij} \mathrm{Log}_{x_k^i}(x_k^j), \sum_{j=1}^{n} a_{ij} \mathrm{Log}_{x_k^i}(x_k^j) \Big\rangle \\ &\qquad = \frac{1}{2} \varepsilon^2 \nabla^2 V(x_k) \langle \hat{\mathcal{L}}_n \mathrm{Log}_{x^\star}(x_k), \hat{\mathcal{L}}_n \mathrm{Log}_{x^\star}(x_k) \rangle. \end{aligned} \tag{5.104}$$

By substituting (5.103) and (5.104) into (5.100), we have

$$\begin{aligned} V(x_{k+1}) - V(x_k) &\leqslant -\varepsilon \langle \mathrm{Log}_{x^\star}(x_k), \hat{\mathcal{L}}_n \mathrm{Log}_{x^\star}(x_k) \rangle \\ &\quad + \frac{1}{2} H_\Delta \varepsilon^2 \langle \hat{\mathcal{L}}_n \mathrm{Log}_{x^\star}(x_k), \hat{\mathcal{L}}_n \mathrm{Log}_{x^\star}(x_k) \rangle. \\ &= -\varepsilon \langle (\hat{\mathcal{L}}_n^{-1} - \frac{1}{2} H_\Delta \varepsilon) \hat{\mathcal{L}}_n \mathrm{Log}_{x^\star}(x_k), \hat{\mathcal{L}}_n \mathrm{Log}_{x^\star}(x_k) \rangle. \end{aligned} \tag{5.105}$$

By choosing $0 < \varepsilon < 2/(H_\Delta \deg(\mathcal{G}_n))$, we have the difference of the Lyapunov function given by

$$V(x_{k+1}) - V(x_k) \leqslant -\lambda_2^2 \|\mathrm{Log}_{x^\star}(x_k)\|^2, \tag{5.106}$$

with λ_2 denoting the smallest eigenvalue of Laplacian $\mathcal{L}_n$. Therefore, since the difference of the Lyapunov function satisfies $V(x_{k+1}) - V(x_k) \leqslant 0$, we could conclude that the consensus value $x^\star$, such that $x_k^1 = \ldots = x_k^n = x^\star$, can be achieved asymptotically. □

The presented proposition about the Riemannian consensus protocol is applicable to any Riemannian manifold whose Hessian of the distance function exists, for example, a quaternion, special-orthogonal group, special-Euclidean group, etc. By utilising the Riemannian consensus protocol, we are now in position to define the measurement along with the coordination update of the distributed Riemannian Kalman filter. Following the procedure of the recursive Bayesian estimation, the prior estimate and its covariance of agent i is shared to the neighbouring agents $j \neq i$. The Riemannian consensus protocol is utilised to calculate the consensus of the prior estimates; while consensus protocol on the tangent space at the prior estimate is utilised to obtain the consensus of the prior covariance matrix.

The proposed local posterior estimate of each agent is obtained via

$$\hat{x}_k^i = \mathrm{Exp}_{\bar{x}_k^i}\Big(K_k^i \mathrm{Log}_{\bar{y}_k^i}(y_k^i)\Big), \text{ for } i \in \mathcal{V}_n. \tag{5.107}$$

In computation of the posterior convariance matrix, we require a parallel transport of a manifold $\mathcal{M}$ defined as a function $\mathrm{PT} : T_p\mathcal{M} \times \mathcal{M} \times \mathcal{M} \to T_{p^\star}\mathcal{M} : (U, p, p^\star) \mapsto \mathrm{PT}(U, p, p^\star)$ that transport U along the geodesic connecting $p \in \mathcal{M}$ to $p^\star \in \mathcal{M}$. Accordingly, the covariance of the local posterior estimate value can given by

$$\hat{P}_{xx,k}^i = \mathrm{PT}\left(\bar{P}_{xx,k}^i - K_k^i \bar{P}_{yy,k}^i (K_k^i)^\top, \bar{x}_k^i, \hat{x}_k^i\right), \tag{5.108}$$

where parallel transport from $T_{\bar{x}_k^i}\mathcal{P}_x$ to $T_{\hat{x}_k^i}\mathcal{P}_x$ has been used. In the above steps, the Kalman gain is calculated via

$$K_k^i = \bar{P}_{xx,k}^i (\bar{P}_{yy,k}^i)^{-1}. \tag{5.109}$$

To calculate the consensus of the posterior estimates, define a function $\mu_\kappa : \mathcal{P}_x \times \mathcal{P}_x \to \mathcal{P}_x : (\hat{x}^i_{k,0}, \hat{x}^j_{k,0}) \mapsto \hat{x}^i_{k,\kappa+1}$, for $j \in \mathcal{N}_i$, as the κ iterations of the Riemannian consensus protocol employed every time step k via

$$\hat{x}^i_{k,l+1} = \mathrm{Exp}_{\hat{x}^i_{k,l}}\Big(\varepsilon \sum_{j=1}^{n} a_{ij} \mathrm{Log}_{\hat{x}^i_{k,l}}(\hat{x}^j_{k,l})\Big), \text{ for } 0 \leqslant l \leqslant \kappa,$$

with $\tilde{x}^i_{k,0} = \tilde{x}^i_k$, and suitable gains defined in Proposition 5.11. Accordingly, the consensus of posterior estimate value can compactly be expressed as

$$\tilde{x}^i_k = \mu_\kappa(\hat{x}^i_k, \hat{x}^j_k), \text{ for } i, j \in \mathcal{V}_n, \ j \in \mathcal{N}_i. \tag{5.110}$$

Since the local posterior covariance matrix $\hat{P}^i_{xx,k}$ is calculated on the tangent space of $\hat{x}^i_k$, we also require a parallel transport of the consensus of $\hat{P}^i_{xx,k}$ from $T_{\hat{x}^i_k}\mathcal{P}_x$ to $T_{\tilde{x}^i_k}\mathcal{P}_x$ such that the consensus of the posterior covariance is given by

$$\tilde{P}^i_{xx,k} = \mathrm{PT}\Big(\hat{P}^i_{xx,k} - \varepsilon \sum_{j=1}^{n} a_{ij}\Big(\hat{P}^i_{xx,k} - \hat{P}^j_{xx,k}\Big), \hat{x}^i_k, \tilde{x}^i_k\Big), \tag{5.111}$$

wherein the Perron matrix has been applied along with the Laplacian matrix and sufficiently-small constant ε.

5.3.3 Practical Algorithm

The practical algorithm of each agent to implement the distributed Riemannian Kalman filter is summarised in Algorithm 5.1.

To implement the proposed DRKF algorithm, one should supply some initial estimates and their covariances, and determine the model of the process and measurement. As the system starts at time step $k > 0$, every agent executes the prediction update which produces prior estimates and covariance matrices. These values are then utilised by the local measurement update to calculate the local posterior estimate and its covariance matrix. The final step in every iteration is the coordination process where every agent executes κ iterations of Riemannian consensus protocol to combine estimates from other agents and calculate the consensus of the covariance matrix. The process, measurement and coordination updates are executed every time step k recursively.

Algorithm 5.1 *DRKF Executed by Agent i, for $i \in \mathcal{V}_n$*

Input: x_0, v_0, w_0, P_0, Q_0, R_0, $f(\cdot)$, $h(\cdot)$.

Return: $\tilde{x}_k^i$, $\tilde{P}_{xx,k}^i$.

If $k = 0$, execute

Initialisation:

1: Initialise the estimate value, noises, and their covariance matrices:
$\tilde{x}_0^i = x_0,\ v_0^i = v_0,\ w_0^i = w_0$; and $\tilde{P}_{xx,0}^i = P_0,\ Q_0^i = Q_0,\ R_0^i = R_0$.

For $k = 1, 2, \ldots, T_k, T_k < \infty$, execute

Prediction Update:

1: Calculate the sigma vector $\mathcal{X}_{k-1}^i$ using (5.89).
2: Propagate the sigma vector through $f(\cdot)$, $h(\cdot)$, using (5.90) and (5.94), respectively.
3: Calculate the prior estimate $\bar{x}_k^i$ using (5.91), and $\bar{y}_k^i$ using (5.95).
4: Calculate the prior covariances $\bar{P}_{xx,k}^i$, $\bar{P}_{yy,k}^i$, and $\bar{P}_{xy,k}^i$ based on (5.92), (5.96), and (5.97), respectively.

Measurement and Coordination Updates:

1: Agent i exhanges information to and receive information from agent $j \in \mathcal{N}_i$.
2: Update the Kalman and communication gains, K_k^i based on (5.109).
3: Update the local posterior estimate $\hat{x}_k^i$ according to (5.107).
4: Update the local posterior covariance $\hat{P}_{xx,k}^i$ via (5.108).
5: Update the consensus posterior estimate $\tilde{x}_k^i$ according to (5.110).
6: Update the consensus posterior covariance $\tilde{P}_{xx,k}^i$ via (5.111).

5.3.4 Numerical Experiments

For verification, we present a set of 50 simulations of the developed DRKF applied to distributed attitude estimation where the state and measurement belong to unit quaternion, a class of Riemannian manifold with positive curvature, denoted by $\mathbb{H} = \{q = [q_w, q_x, q_y, q_z]^\top \in \mathbb{R}^4 : \|q\|^2 = 1\}$. The following simulation is modified from the estimation scenario in [121] by considering that the tracking is performed by a set of several radars. For instance, consider a group of 4 tracking

radars connected via an undirected graph depicted in Fig. 5.1. The radars are

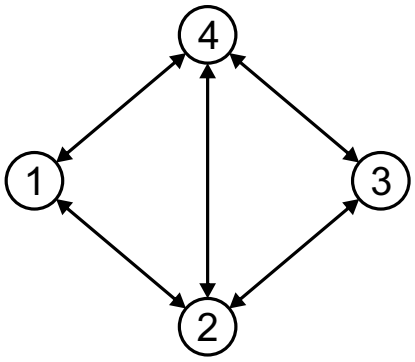


Figure 5.1: Graph topology of the distributed attitude estimation.

assigned to estimate and track an aircraft attitude given by

$$\dot{q}(t) = \frac{1}{2}\omega(t) \otimes q(t)$$

with

$$\omega(t) = \begin{bmatrix} 0 \\ 0.03\sin((\pi t/600)^\circ) \\ 0.03\sin((\pi t/600 - 300)^\circ) \\ 0.03\sin((\pi t/600 - 600)^\circ) \end{bmatrix}.$$

In this case, we use $\otimes$ to denote the quaternion multiplication. The dynamics is integrated with interval $[0, 20]$ and the initial quaternion state is given by $q(0) = [0.96, 0.13, 0.19, \sqrt{1 - 0.96^2 - 0.13^2 - 0.19^2}]^\top$.

Accordingly, the process dynamics to estimate and the measurement model with their noises are given by

$$\begin{aligned} q_{k+1} &= \mathrm{Exp}_{\tilde{\xi}(q_k)}\left(\mathrm{Log}_{\tilde{\xi}(q_k)}\left(\xi(q_k) + v_k\right)\right) \\ y_k^i &= \mathrm{Exp}_{\tilde{\zeta}(q_k)}\left(\mathrm{Log}_{\tilde{\zeta}(q_k)}\left(\zeta(q_k) + w_k^i\right)\right), \text{ for } i \in \mathcal{V}_n, \end{aligned}$$

where

$$\begin{aligned} \xi(q_k) &= \left[\cos(\theta(t)) \ \frac{\omega^\top}{\|\omega\|}\sin(\theta(t))\right]^\top \otimes q_k, \\ \zeta(q_k) &= q_k, \end{aligned}$$

and $\theta(t) = (1/2)\|\omega(t)\|\delta t$. The expected value of the process noise utilised by all radars is given by $\mathbb{E}\{v_k\} = \mathbf{0}_3$ with the covariance of $Q_k = (1.71236 \times 10^{-2})^2 I_3$. The expected values of the measurement of all sensors are similiar, $\mathbb{E}\{w_k^i\} = \mathbf{0}_3$, with different covariance given by $R_k^1 = (5.4\pi/180)^2 I_3$, $R_k^2 = (4.1\pi/180)^2 I_3$, $R_k^3 = (4.5\pi/180)^2 I_3$, and $R_k^4 = (3.1\pi/180)^2 I_3$, respectively. For the Riemannian consensus of the posterior estimates, we allocate $\kappa = 7$ iterations every time step k to guarantee that the consensus value is reached.

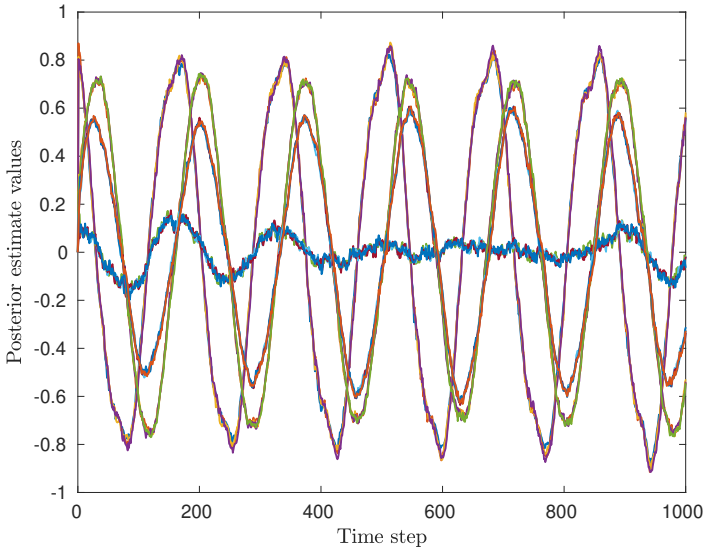


Figure 5.2: Posterior estimate of the quaternion states by all radars.

To illustrate one of the simulation performances, we present the convergence of the state parameters, measurement-tracking ability and MSE in Figs. 5.2, 5.3 and 5.4, respectively. After executing Algorithm 5.1 with the specified parameters, we can observe that the estimate of the system state of every radar can successfully achieve consensus as illustrated in Fig. 5.2. The proposed algorithm allows the system to predict and track the measurement as illustrated in Fig. 5.3 where the predictions of the measurement align with the trajectory of the process almost every time step k, and also filter the information from the noisy measurements. The trajectories of the MSE of the proposed filter and the MSE from the intrinsic CRB are presented in Fig. 5.4. In this case, the final MSE of proposed filter is 4.8780×10^{-3} while the final MSE from the intrinsic CRB is 4.8303×10^{-3}.

In Fig. 5.5, we present the MSEs in our 50 simulations. Based on these simulation results, we can verify that, as predicted by our analysis about intrinsic

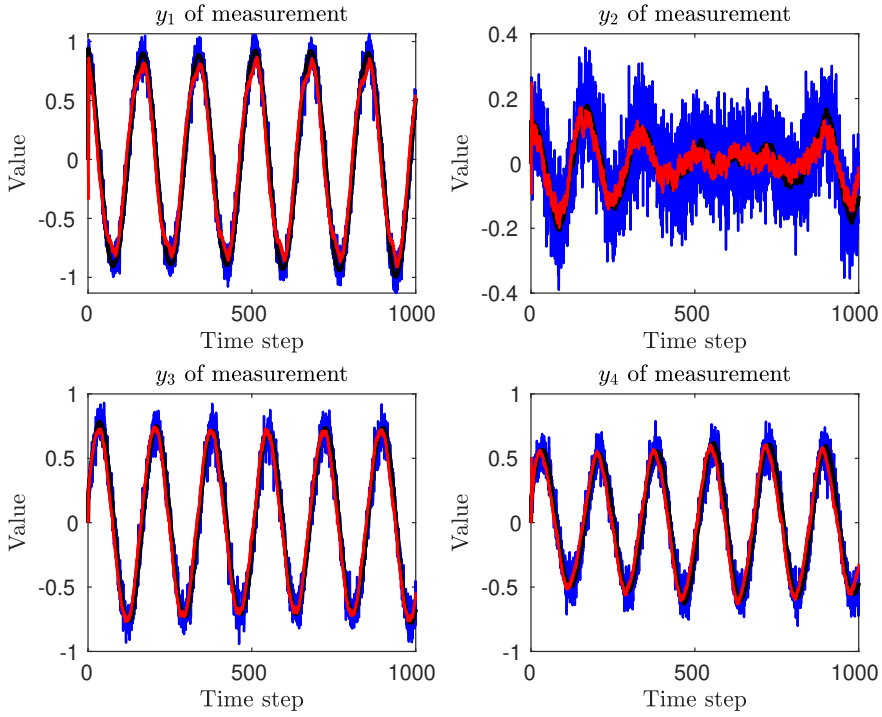


Figure 5.3: Process (black), measurement (blue) and prediction of the measurement (red) by all radars.

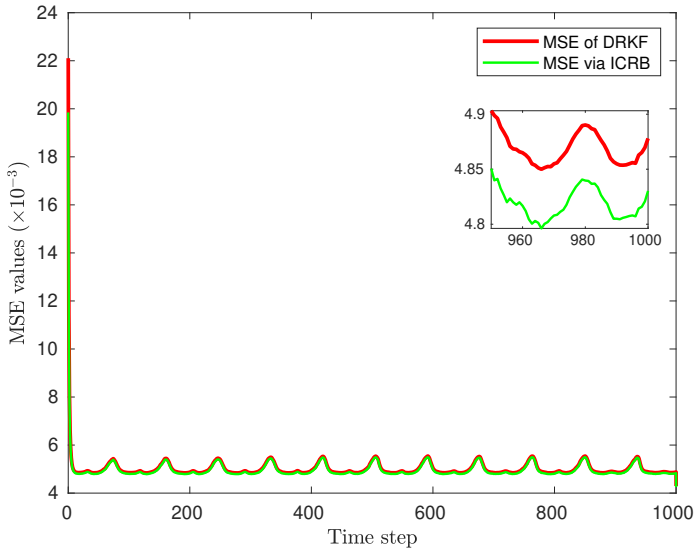


Figure 5.4: Average MSEs of the radars.

CRB for distributed Bayesian estimator, the values of the MSE from the proposed filter is not lower than the lower-bound MSE obtained via the intrinsic CRB.

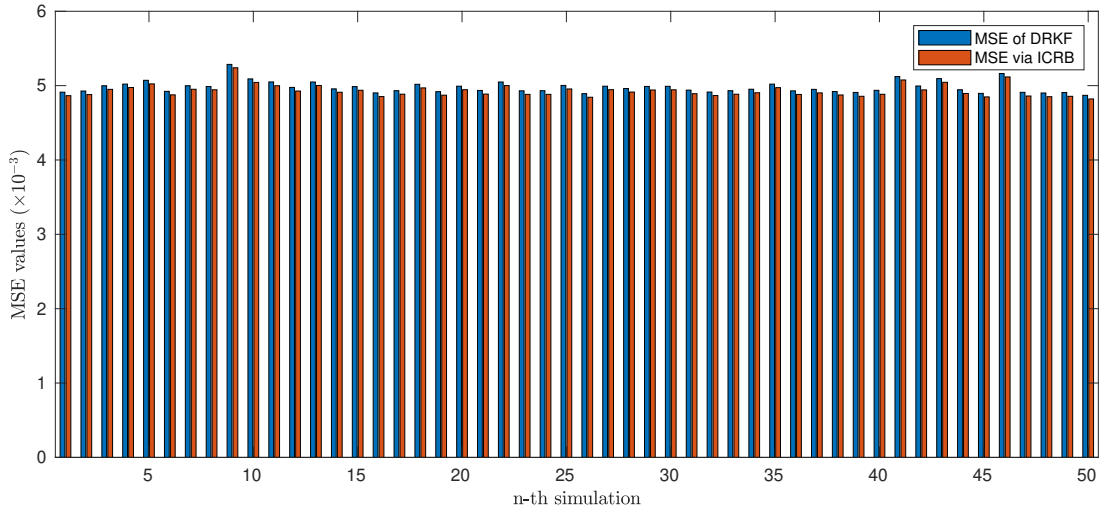


Figure 5.5: MSEs of 50 simulations.

5.4 Conclusions

In this chapter, we have firstly investigated the lower bounds one can expect when designing an estimator following the distributed Bayesian estimator procedure when the state and measurement belong to Riemannian manifold, referred to as the intrinsic CRB for distributed Bayesian estimator. The intrinsic CRBs are mainly obtained by (i) utilising the fact that the product of Riemannian manifolds is also a Riemannian manifold; (ii) extracting the covariance and information matrices at time-step k from the augmented matrices up to time step k; and (iii) finally, applying the Kullback-Liebler average to combine FIMs from neighbouring agents. From our analysis, it has been proved that the intrinsic CRB, and accordingly the MSE, of the distributed Bayesian estimator depends on the recursive FIMs and the curvature of the system manifolds. Thereafter, we developed a class of distributed Bayesian estimator whose state and measurement are Gaussian, referred to as the distributed Riemannian Kalman filter (DRKF). The DRKF has been designed by following the similar procedure of the DUKF design for Euclidean systems. For verification purpose, a simulation is conducted for distributed attitude estimation using the distributed Riemannian filter. Finaly, the simulation result validates that the actual MSE of the estimation via the proposed filter is greater than the MSE of the correponding intrinsic CRBs.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

This thesis addresses issues in the coordination and Bayesian estimation processes in MASs.

In the coverage control problem, several control protocols are investigated to tackle the cooperative issues of the networked robots from the perspective of distributed optimisation. Firstly, in our analysis, the proposed gradient-based algorithm has been developed via distributed optimisation to allow distributed cooperation among robots to achieve optimal positions while maintaining the formation. Secondly, to reduce the communication burden and improve the timeliness, another coverage control protocol has been established to guarantee the finite-time convergence independent to the initial positions. A potential-field based obstacle avoidance has also been utilised to prevent collisions between an agent and other agents, and between an agent and static obstacles in an environment, while all agents are moving towards the optimal positions. Finally, numerical experiments demonstrate the performance of the proposed algorithms and validate our results.

Subsequently, a distributed unscented Kalman filter has been proposed for general communication scheme. The notion of unscented transformation has been employed to predict the state, measurement and communication data. In the posterior estimator, the Kalman and communication gains are optimised to produce an optimal posterior estimate value. By using this mechanism, the

consensus-based distributed Kalman filter in literature could be considered as a special case of our proposed algorithm. Furthermore, we have shown that the estimate error is exponentially bounded in mean-square with regards to some boundaries. Combined with the coverage control protocol in our results in Chapter 3, the proposed algorithm was used to estimate the density function and find the optimal deployment of robots. The numerical experiments suggest that that the proposed distributed algorithm has outperformed the modified-consensus observer in [21] by successfully estimating the unknown density function of an environment indicated, driving all agents to the optimal centroid positions and minimise the true objective function without oscillations. Moreover, the simulation results have indicated that the proposed algorithm needs less computation time than the existing modified-consensus observer.

Thereafter, we have first investigated the lower bounds one can expect when designing an estimator following the distributed Bayesian estimator procedure when the state and measurement belong to Riemannian manifold, referred to as the intrinsic CRB for distributed Bayesian estimator. We have shown that the intrinsic CRBs, and accordingly the MSE, of the distributed Bayesian estimator, depends on the recursive FIMs and the curvature of the system manifolds. We have also developed a class of distributed Bayesian estimator whose state and measurement are Gaussian, referred to as the distributed Riemannian Kalman filter (DRKF). A simulation has been carried out to perform distributed attitude estimation using the DRKF. In the simulation result, we have verified that the MSE of the DRKF is greater than the analytical MSE from the corresponding intrinsic CRBs.

6.2 Future Works

There are several possible extensions of the works presented in this thesis which are listed as follows:

1. The first possible future direction is the distributed coverage control with non-isotropic sensing unreliability function. In Chapter 3, we have developed algorithms to handle distributed implementation of coverage control problem.

These algorithms require the sensors to have equal abilities in detecting information from any direction, that is, isotropic. However, in reality, there are sensors with directional sensing ability. For instance, forward-facing light sensor of a mobile robot will have non-isotropic sensing unreliability function. Investigation about distributed implementation of such scenario should become an interesting future research topic because it can solve many more practical problems.

2. Secondly, another research direction could be to analyse the convergence behaviour of the DRKF algorithm. In Chapter 5, intrinsic CRBs for distributed Bayesian estimator have been developed to provide the most sensible performance one can expect when designing such estimator. However, this is not to guarantee the convergence analysis of such filtering algorithm. Therefore, it is worthwhile to analytically establish the convergence behaviour of the DRKF algorithm.

3. The third possible future work would be the DRKF for any geodesically-complete Riemannian manifolds. Although in Chapter 5 we have designed the DRKF and its lower bounds, a neccessary condition of this algorithm is that the state and measurement should be in the interior of the maximal domain. Investigation on how to relax this condition such that all states initialised at any point in a complete Riemannian manifold converge to the consensus value could become another future work. In the practical implementation, for example, this relaxation might reduce the MSE of rotational attitude estimator algorithm in a spacecraft with any initial estimate values.

4. Finally, it would be interesting to see the practical implementation of the proposed algorithms. In this thesis, we have proposed several algorithms in coordination and estimation of multi-agent systems and analyse their theoretical performance. However, since implementation sometimes requires additional tuned parameter sensor network, experiments that implement the distributed coverage control, DUKF, and DRKF, to real robotic sensor network might become a direction of future works.

Bibliography

[1] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987*, vol. 21, no. 4, pp. 25–34, 1987.

[2] Z. Qu, J. Chunyu, and J. Wang, "Nonlinear cooperative control for consensus of nonlinear and heterogeneous systems," *Proceedings of the IEEE Conference on Decision and Control*, no. 1, pp. 2301–2308, 2007.

[3] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, Mar 2006.

[4] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems*, vol. 27, no. 2, pp. 71–82, Apr 2007.

[5] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.

[6] Z. Li and Z. Duan, *Cooperative control of multi-agent systems: A consensus region approach*. CRC Press, Dec 2017.

[7] M. Deghat, B. D. O. Anderson, and Z. Lin, "Combined flocking and distance-based shape control of multi-agent formations," *IEEE Transactions on Automatic Control*, vol. 61, no. 7, pp. 1824–1837, Jul 2016.

[8] M. Akbari, B. Gharesifard, and T. Linder, "Distributed online convex optimization on time-varying directed graphs," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 417–428, 2017.

[9] J. Wang and N. Elia, "A control perspective for centralized and distributed convex optimization," *Proceedings of the IEEE Conference on Decision and Control*, no. 3, pp. 3800–3805, 2011.

[10] B. Gharesifard and J. Cortes, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, Mar 2014.

[11] Z. Li, Z. Ding, J. Sun, and Z. Li, "Distributed adaptive convex optimization on directed graphs via continuous-time algorithms," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1434–1441, May 2018.

[12] R. Olfati-Saber, "Kalman-Consensus Filter : Optimality, stability, and performance," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Dec 2009, pp. 7036–7042.

[13] W. Li, G. Wei, F. Han, and Y. Liu, "Weighted average consensus-based unscented Kalman filtering," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 558–567, Feb 2016.

[14] G. Battistelli and L. Chisci, "Stability of consensus extended Kalman filter for distributed state estimation," *Automatica*, vol. 68, no. Supplement C, pp. 169–178, 2016.

[15] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, Sep 2010.

[16] I. Arasaratnam, S. Haykin, and T. R. Hurd, "Cubature Kalman filtering for continuous-discrete systems: theory and simulations," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 4977–4993, Oct 2010.

[17] R. Tron, B. Afsari, and R. Vidal, "Riemannian consensus for manifolds with bounded curvature," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 921–934, 2013.

[18] R. Tron and R. Vidal, "Distributed 3-D localization of camera sensor networks from 2-D image measurements," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3325–3340, 2014.

[19] M. Schwager, J.-J. Slotine, and D. Rus, "Decentralized, Adaptive Control for Coverage with Networked Robots." IEEE, Apr 2007, pp. 3289–3294.

[20] M. Schwager, J. McLurkin, J. J. E. Slotine, and D. Rus, "From Theory to Practice: Distributed Coverage Control Experiments with Groups of Robots," *Springer Tracts in Advanced Robotics*, vol. 54, no. 1, pp. 127–136, 2009.

[21] M. Schwager, M. P. Vitus, S. Powers, D. Rus, and C. J. Tomlin, "Robust adaptive coverage control for robotic sensor networks," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 462–476, 2017.

[22] Y. Wang and C. (Brad) Yu, "Translation and attitude synchronization for multiple rigid bodies using dual quaternions," *Journal of the Franklin Institute*, vol. 354, no. 8, pp. 3594–3616, 2017.

[23] Z. Li, Z. Ding, J. Sun, and Z. Li, "Distributed adaptive convex optimization on directed graphs via continuous-time algorithms," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1434–1441, 2018.

[24] H. Kaptui Sipowa, J. W. McMahon, and T. Deka, "Distributed unscented-information Kalman filter (UIKF) for cooperative localization in spacecraft formation flying," in *AIAA Scitech 2020 Forum*, no. January. Reston, Virginia: American Institute of Aeronautics and Astronautics, Jan 2020, pp. 1–17.

[25] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, 2008.

[26] Z. Li, Z. Duan, G. Chen, and L. Huang, "Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, vol. 57, no. 1, pp. 213–224, 2010.

[27] C. Wang, Z. Zuo, Z. Lin, and Z. Ding, "Consensus control of a class of lipschitz nonlinear systems with input delay," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 11, pp. 2730–2738, 2015.

[28] A. Nedic and J. Liu, "On Convergence Rate of Weighted-Averaging Dynamics for Consensus Problems," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 766–781, 2017.

[29] G. Battistelli and L. Chisci, "KullbackâĂŞLeibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, pp. 707–718, Mar 2014.

[30] Z. Ding, "Consensus output regulation of a class of heterogeneous nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2648–2653, 2013.

[31] ——, "Adaptive consensus output regulation of a class of nonlinear systems with unknown high-frequency gain," *Automatica*, vol. 51, pp. 348–355, 2015.

[32] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[33] G. Wen, Z. Duan, G. Chen, and W. Yu, "Consensus tracking of multi-agent systems with lipschitz-type node dynamics and switching topologies," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 2, pp. 499–511, 2014.

[34] Y. Mei, Y. H. Lu, Y. C. Hu, and C. S. Lee, "Deployment strategy for mobile robots with energy and timing constraints," *Proceedings - IEEE*

International Conference on Robotics and Automation, vol. 2005, no. 3, pp. 2816–2821, 2005.

[35] S. P. Bhat and D. S. Bernstein, "Finite-time stability of continuous autonomous systems," *SIAM Journal on Control and Optimization*, vol. 38, no. 3, pp. 751–766, Jan 2000.

[36] F. Xiao, L. Wang, J. Chen, and Y. Gao, "Finite-time formation control for multi-agent systems," *Automatica*, vol. 45, no. 11, pp. 2605–2611, 2009.

[37] S. Khoo, L. Xie, and Z. Man, "Robust finite-time consensus tracking algorithm for multirobot systems," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 219–228, 2009.

[38] H. Du, C. Yang, and R. Jia, "Finite-time formation control of multiple mobile robots," *6th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2016*, pp. 416–421, 2016.

[39] J. Wang, H. Liang, Z. Sun, S. Zhang, and M. Liu, "Finite-time control for spacecraft formation with dual-number-based description," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 3, pp. 950–962, 2012.

[40] Z. Zuo and L. Tie, "A new class of finite-time nonlinear consensus protocols for multi-agent systems," *International Journal of Control*, vol. 87, no. 2, pp. 363–370, 2014.

[41] Z. Zuo, "Nonsingular fixed-time consensus tracking for second-order multi-agent networks," *Automatica*, vol. 54, pp. 305–309, 2015.

[42] C. Wang, H. Tnunay, Z. Zuo, B. Lennox, and Z. Ding, "Fixed-Time Formation Control of Multirobot Systems: Design and Experiments," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 8, 2019.

[43] A. Okabe and A. Suzuki, "Locational optimization problems solved through Voronoi diagrams," *European Journal of Operational Research*, vol. 98, no. 3, pp. 445–456, May 1997.

[44] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed., ser. Wiley series in probability and statistics. New York: Wiley, Jan 1995, vol. 26, no. 1.

[45] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, "Distributed Algorithms for Environment Partitioning in Mobile Robotic Networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1834–1848, Aug 2011.

[46] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage Control for Mobile Sensing Networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, Apr 2004.

[47] S. G. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multirobot Control Using Time-Varying Density Functions," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 489–493, Apr 2015.

[48] S. Salhi, "Facility Location: A Survey of Applications and Methods," *Journal of the Operational Research Society*, vol. 47, no. 11, pp. 1421–1422, Nov 1996.

[49] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM - Control, Optimisation and Calculus of Variations*, vol. 11, no. 4, pp. 691–719, 2005.

[50] L. C. Pimenta, V. Kumar, R. C. Mesquita, and G. A. Pereira, "Sensing and coverage for a network of heterogeneous robots," *Proceedings of the IEEE Conference on Decision and Control*, no. 2, pp. 3947–3952, 2008.

[51] A. Gusrialdi, T. Hatanaka, and M. Fujita, "Coverage control for mobile networks with limited-range anisotropic sensors," *Proceedings of the IEEE Conference on Decision and Control*, pp. 4263–4268, 2008.

[52] Y. Kantaros, M. Thanou, and A. Tzes, "Distributed coverage control for concave areas by a heterogeneous Robot-Swarm with visibility sensing constraints," *Automatica*, vol. 53, pp. 195–207, 2015.

[53] H. F. Parapari, F. Abdollahi, and M. B. Menhaj, "Coverage control in non-convex environment considering unknown non-convex obstacles," *2014 2nd RSI/ISM International Conference on Robotics and Mechatronics, ICRoM 2014*, pp. 119–124, 2014.

[54] Y. Kantaros and M. M. Zavlanos, "Distributed communication-aware coverage control by mobile sensor networks," *Automatica*, vol. 63, pp. 209–220, 2016.

[55] M. Schwager, J. J. Slotine, and D. Rus, "Decentralized, adaptive control for coverage with networked robots," *Proceedings - IEEE International Conference on Robotics and Automation*, no. April, pp. 3289–3294, 2007.

[56] S. Martinez, "Distributed interpolation schemes for field estimation by mobile sensor networks," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 491–500, Mar 2010.

[57] M. Schwager, F. Bullo, D. Skelly, and D. Rus, "A ladybug exploration strategy for distributed adaptive coverage control," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2346–2353, 2008.

[58] M. Schwager, J. J. Slotine, and D. Rus, "Consensus learning for distributed coverage control," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1042–1048, 2008.

[59] J. Choi, S. Oh, and R. Horowitz, "Distributed learning and cooperative control for multi-agent systems," *Automatica*, vol. 45, no. 12, pp. 2802–2814, Dec 2009.

[60] M. Schwager, F. Bullo, D. Skelly, and D. Rus, "A ladybug exploration strategy for distributed adaptive coverage control." IEEE, May 2008, pp. 2346–2353.

[61] M. Schwager, J.-J. Slotine, and D. Rus, "Consensus learning for distributed coverage control." IEEE, May 2008, pp. 1042–1048.

[62] Y. Deng, Z. Wang, and L. Liu, "Unscented Kalman filter for spacecraft pose estimation using twistors," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 8, pp. 1844–1856, 2016.

[63] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 15–22, 2014.

[64] S. Holmes, G. Klein, and D. Murray, "An O(NÂš) square root unscented Kalman filter for visual simultaneous localization and mapping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1251–1263, Jul 2009.

[65] R. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 15, no. 2, pp. 175–184, Apr 1970.

[66] J. L. Crassidis and F. L. Markley, "Unscented filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 4, pp. 536–542, 2003.

[67] 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, Apr 1999.

[68] E. A. Wan and R. V. D. 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)*, 2000, pp. 153–158.

[69] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar 2004.

[70] M. Geist, O. Pietquin, and G. Fricout, "Tracking in reinforcement learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009.

[71] M. Geist and O. Pietquin, "Kalman temporal differences," *Journal of Artificial Intelligence Research*, vol. 39, pp. 483–532, 2010.

[72] L. Li and Y. Xia, "UKF-based nonlinear filtering over sensor networks with wireless fading channel," *Information Sciences*, vol. 316, pp. 132–147, 2015.

[73] M. Oussalah and J. D. Schutter, "Possibilistic Kalman filtering for radar 2D tracking," *Information Sciences*, vol. 130, no. 1, pp. 85–107, 2000.

[74] G. Hao, S.-l. Sun, and Y. Li, "Nonlinear weighted measurement fusion Unscented Kalman Filter with asymptotic optimality," *Information Sciences*, vol. 299, pp. 85–98, 2015.

[75] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, Dec 2005, pp. 8179–8184.

[76] G. Liu and G. Tian, "Square-root sigma-point information consensus filters for distributed nonlinear estimation," *Sensors*, vol. 17, no. 4, p. 800, Apr 2017.

[77] S. Wang and W. Ren, "Fully distributed nonlinear state estimation using sensor networks," in *2017 American Control Conference (ACC)*, May 2017, pp. 2562–2567.

[78] Q. Tan, X. Dong, Q. Li, and Z. Ren, "Weighted average consensus-based cubature Kalman filtering for mobile sensor networks with switching topologies," in *2017 13th IEEE International Conference on Control Automation (ICCA)*, Jul 2017, pp. 271–276.

[79] Y. Niu and L. Sheng, "Distributed consensus-based unscented Kalman filtering with missing measurements," in *2017 36th Chinese Control Conference (CCC)*, Jul 2017, pp. 8993–8998.

[80] Li, Wang, and Zheng, "Adaptive consensus-based unscented information filter for tracking target with maneuver and colored noise," *Sensors*, vol. 19, no. 14, p. 3069, Jul 2019.

[81] P. Tichavsky, C. H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1386–1396, 1998.

[82] Y. Zheng, O. Ozdemir, R. Niu, and P. K. Varshney, "New conditional posterior Cramér-Rao lower bounds for nonlinear sequential Bayesian estimation," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5549–5556, 2012.

[83] S. Smith, "Optimization techniques on Riemannian manifolds," *Hamiltonian and Gradient Flows, Algorithms and Control*, vol. 3, pp. 113–136, 1995.

[84] N. Boumal, "On intrinsic Cramér-Rao bounds for Riemannian submanifolds and quotient manifolds," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1809–1821, Apr 2013.

[85] S. Bonnabel and A. Barrau, "An intrinsic Cramér-Rao bound on lie groups," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9389, no. 3, pp. 664–672, 2015.

[86] M. Šimandl, J. Královec, and P. Tichavský, "Filtering, predictive, and smoothing Cramér-Rao bounds for discrete-time nonlinear dynamic systems," *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*, vol. 37, pp. 697–710, 2007.

[87] L. Zuo, R. Niu, and P. K. Varshney, "Conditional posterior Cramér-Rao lower bounds for nonlinear recursive filtering," *2009 12th International Conference on Information Fusion, FUSION 2009*, no. 7, pp. 1528–1535, 2009.

[88] E. Saatci and A. Akan, "Posterior Cramér-Rao lower bounds for dual Kalman estimation," *Digital Signal Processing*, vol. 22, no. 1, pp. 47–53, Jan 2012.

[89] X. Zhong, A. Mohammadi, A. B. Premkumar, and A. Asif, "A distributed particle filtering approach for multiple acoustic source tracking using an

acoustic vector sensor network," *Signal Processing*, vol. 108, pp. 589–603, 2015.

[90] S. Hauberg, F. Lauze, and K. S. Pedersen, "Unscented Kalman filtering on Riemannian manifolds," *Journal of Mathematical Imaging and Vision*, vol. 46, no. 1, pp. 103–120, May 2013.

[91] H. M. T. Menegaz, J. Y. Ishihara, G. A. Borges, and A. N. Vargas, "A systematization of the Unscented Kalman filter theory," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2583–2598, Oct 2015.

[92] G. Bourmaud, R. Megret, A. Giremus, and Y. Berthoumieu, "Discrete extended Kalman filter on Lie groups," *European Signal Processing Conference*, pp. 1–5, 2013.

[93] A. Barrau and S. Bonnabel, "Intrinsic filtering on Lie groups with applications to attitude estimation," *IEEE Transactions on Automatic Control*, vol. 60, no. 2, pp. 436–449, 2015.

[94] C. Zhang, A. Taghvaei, and P. G. Mehta, "Feedback particle filter on riemannian manifolds and matrix lie groups," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2465–2480, 2018.

[95] N. Filipe, M. Kontitsis, and P. Tsiotras, "Extended Kalman filter for spacecraft pose estimation using dual quaternions," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 9, pp. 1625–1641, Sep 2015.

[96] D. Kang, C. Jang, and F. C. Park, "Unscented Kalman filtering for simultaneous estimation of attitude and gyroscope Bias," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 350–360, Feb 2019.

[97] M. Brossard, S. Bonnabel, and J.-P. Condomines, "Unscented Kalman filtering on Lie groups," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2017-Septe, no. 3. IEEE, Sep 2017, pp. 2485–2491.

[98] M. Brossard, S. Bonnabel, and A. Barrau, "Unscented Kalman filter on Lie groups for visual inertial odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct 2018, pp. 649–655.

[99] G. Loianno, M. Watterson, and V. Kumar, "Visual inertial odometry for quadrotors on SE(3)," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2016-June, no. 3. IEEE, May 2016, pp. 1544–1551.

[100] R. Agaev and P. Chebotarev, "On the spectra of nonsymmetric Laplacian matrices," *Linear Algebra and its Applications*, vol. 399, pp. 157–168, Apr 2005.

[101] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge: Cambridge University Press, Dec 1985.

[102] M. do Carmo, *Riemannian geometry*, ser. Mathematics. theory and applications. Boston: Birkhäuser Basel, Dec 1992.

[103] X. Pennec, "Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements," *Journal of Mathematical Imaging and Vision*, vol. 25, no. 1, pp. 127–154, 2006.

[104] B. Afsari, "Riemannian L^p center of mass: existence, uniqueness, and convexity," *Proceedings of the American Mathematical Society*, vol. 139, no. 02, pp. 655–655, Feb 2011.

[105] B. Afsari, R. Tron, and R. Vidal, "On the convergence of gradient descent for finding the Riemannian center of mass," *SIAM Journal on Control and Optimization*, vol. 51, no. 3, pp. 2230–2260, Jan 2013.

[106] B. Afsari, "Means and averaging on Riemannian manifolds," Ph.D. dissertation, University of Maryland, 2009.

[107] W. S. Kendall, "Probability, convexity, and harmonic maps with small image i: Uniqueness and fine existence," *Proceedings of the London Mathematical Society*, vol. s3-61, no. 2, pp. 371–406, 1990.

[108] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002.

[109] Z. Ding, *Nonlinear and adaptive control systems.* Institution of Engineering and Technology, Apr 2013.

[110] A. R. Abdulghany, "Generalization of parallel axis theorem for rotational inertia," *American Journal of Physics*, vol. 85, no. 10, pp. 791–795, Oct 2017.

[111] S. P. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge, UK; New York: Cambridge University Press, 2004.

[112] T.-J. Tarn and Y. Rasis, "Observers for nonlinear stochastic systems," *IEEE Transactions on Automatic Control*, vol. 21, no. 4, pp. 441–448, Aug 1976.

[113] E. A. Wan and R. V. D. Merwe, "Chapter 7: The unscented Kalman filter," *Kalman Filtering and Neural Networks*, 2001.

[114] K. Xiong, H. Y. Zhang, and C. W. Chan, "Performance evaluation of UKF-based nonlinear filtering," *Automatica*, vol. 42, no. 2, pp. 261–270, 2006.

[115] H. Tnunay, Z. Li, C. Wang, and Z. Ding, "Distributed collision-free coverage control of mobile robots with consensus-based approach," in *2017 13th IEEE International Conference on Control Automation (ICCA)*, Jul 2017, pp. 678–683.

[116] R. M. Sanner and J.-J. Slotine, "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 837–863, Nov 1992.

[117] N. Boumal, "Optimization and estimation on manifolds," *Information and Inference*, no. February 2014, 2014.

[118] S. Smith, "Covariance, subspace, and intrinsic Crameŕ-Rao bounds," *IEEE Transactions on Signal Processing*, vol. 53, no. 5, pp. 1610–1630, May 2005.

[119] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Communications on Pure and Applied Mathematics*, vol. 30, no. 5, pp. 509–541, 1977.

[120] B. Afsari, R. Tron, and R. Vidal, "On the convergence of gradient descent for finding the Riemannian center of mass," *SIAM Journal on Control and Optimization*, vol. 51, no. 3, pp. 2230–2260, Jan 2013.

[121] H. M. T. Menegaz, J. Y. Ishihara, and H. T. M. Kussaba, "Unscented Kalman filters for Riemannian state-space systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 4, pp. 1487–1502, Apr 2019.