Information fusion and cooperative control for target search and localization in multi-agent sensor networks

Hu, Jinwen

2013

Hu, J. (2013). Information fusion and cooperative control for target search and localization in multi-agent sensor networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/51879>

<https://doi.org/10.32657/10356/51879>

**Information Fusion and Cooperative Control for
Target Search and Localization in Multi-Agent
Sensor Networks**

Jinwen Hu

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirements for the degree of
Doctor of Philosophy

2013

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....

Date

.....

Jinwen Hu

Acknowledgement

I would like to give my sincere thanks to all those who have supported me consistently in my study and life.

First and foremost, I want to express my greatest gratitude and thankfulness to my supervisor, Prof. Xie Lihua. Without the professional guidance and continuous encouragement of him, I would not have archived the fulfillment of this thesis. It is his insightful guidance that brings me into such an interesting research field for my four years PhD study. It is his patient discussions and comments that helps me conquer all the tough difficulties in my research. It is his rigorous scholarship and generous humanity that teaches me how to become a well qualified researcher.

I am also thankful to my former supervisor, Prof. Zhang Cishen, for his valuable guidance and kindly help in the first two years of my PhD study. His expertized suggestions have been great motivations for my academic work.

Many of my lab colleagues in the Sensor Network Laboratory of NTU have been a good source of ideas and helps. Especially, I would like to thank Xu Jun, You Keyou, Xiao Nan and Liu Shuai for their constructive ideas and insightful comments during my research.

Finally, I owe a lot of thanks to my parents who have supported and believed in me for 30 years, and my wife Xu Zhao who has given me the most cares and consideration in my life since we got married. It is their consistent and unreserved love that supports me to go so far no matter how many times I fell down.

Abstract

Multi-agent sensor networks (MSNs) are comprised of multiple static or mobile agents capable of collecting, processing, storing and transferring information from one agent to another. They can play a critical role in several application domains such as landmine detection and identification, monitoring of endangered species, monitoring of urban environments, manufacturing plants, and civil infrastructure, and intruder and target detection systems. These networks are expected to operate cooperatively and reliably in cluttered dynamic environments with little human intervention. However, coordinating such large heterogeneous sensor networks is challenging and requires the development of novel methods of communication, motion control and planning, computation, proactive estimation and sensing, and power management. This thesis summarizes research results on information fusion and cooperative control for target search and localization in MSNs.

First, the distributed target estimation problem for linear time-varying systems in MSNs is addressed. A diffusion Kalman filtering algorithm based on the covariance intersection method is proposed, where local estimates are fused by incorporating the covariance information of local Kalman filters. Our algorithm leads to a stable estimate for each agent regardless of whether or not the system is uniformly observable locally with the measurements of itself and its neighbors as long as the system is uniformly observable with the measurements of all agents and the communication is sufficiently fast compared to the sampling. Simulation results validate the effectiveness of the proposed distributed Kalman filtering algorithm.

Second, we study the cooperative control for target localization and pursuit in the ground MSNs. Conventional target tracking methods always require an explicit system observation model of the target positions, which, however, would fail if such model is not available. Thus, a distributed target localization and pursuit scheme is proposed based on discrete measurements of the energy intensity field produced by static or mobile targets. The accurate observation model of such field is not available except some critical bounds. By our control strategy, all agents are categorized into two groups: the leaders, responsible for the target pursuit, and the followers, responsible for the formation and connectivity maintenance. The influence of the system parameters on the convergence of leaders to the local maximum points is analyzed. Finally, the proposed scheme is demonstrated by simulation.

Next, we study the cooperative search for multiple stationary ground targets by a group of unmanned aerial vehicles (UAVs) with limited sensing and communication capabilities, where targets and UAVs are moving in planes. The whole surveillance region is partitioned into cells where each cell is associated with a probability of target existence within the cell, which constitutes a probability map for the whole region. Each agent keeps an individual probability map and updates the map individually with measurements according to Bayesian rule. A nonlinear transformation of the probability map is introduced to simplify the computation by linearizing the Bayesian update. A consensus-like distributed fusion scheme is proposed for multi-agent map fusion. It is proven that all the individual probability maps converge to the same one that reflects the true existence or nonexistence of targets within each cell. Coverage and topology control algorithms are designed for the path planning of mobile agents. Moreover, the performance of the fusion scheme for asynchronous implementations of sampling and communication is analyzed. Finally, the effectiveness of the proposed algorithms is illustrated via simulations.

Further, we consider the vision-based cooperative search for multiple mobile ground targets by a group of UAVs with limited sensing and communication capabilities and moving in a three dimensional space. The airborne camera on each UAV has a lim-

ited field of view and its target discriminability varies as a function of altitude. First, a general target detection probability model is built based on the physical imaging process of a camera. Based on the previous results, we propose a generalized distributed probability map updating model which includes the fusion of measurement information, information sharing among neighboring agents, information decaying and transmission due to environmental changes such as the target movement. Furthermore, we formulate the target search problem by multiple agents as a cooperative coverage control problem by optimizing the collective coverage area and the detection performance. The proposed map updating model and the cooperative control scheme are distributed, i.e., allowing that each agent only communicates with its neighbors within its communication range. Finally, the effectiveness of the proposed algorithms is illustrated by simulation.

Finally, we investigate the adaptive sensing for three-dimensional target tracking in MSNs based on measurements from the time-difference-of-arrival (TDOA) sensors. An iterated filtering algorithm combined with the Gauss-Newton method is applied to estimate the target location. By minimizing the determinant of the estimation error covariance matrix, an optimal adaptive sensing strategy is developed. A gradient-based control law for each agent is proposed and a set of stationary points for local optimum geometric configurations of the agents is given. The optimal sensing strategy is further compared with other sensing strategies using different optimization criteria such as the Cramer-Rao lower bound. Possible modifications of the proposed optimal sensing strategy are also discussed. Finally, the proposed sensing strategy is demonstrated and compared with other sensing strategies by simulation, which shows that our method can provide good performance with even only one TDOA measurement at each time.

Symbols and Acronyms

Algebraic Operators

A^T	Transpose of matrix A
A^{-1}	Inverse of matrix A
I	Identity matrix of appropriate dimension
k	Discrete time step
i	Agent index number
N	Total number of agents
$\mu_{i,k}$	Position of agent i at time step k
δ_{ij}	Kronecker delta function
$\mathbb{1}_{\{\bullet\}}$	Indicator function
$\mathbf{1}$	Column vector of appropriate dimension with all entries equal to one
R_c	Communication range of each agent
$E[X]$	Mathematical expectation of X
$tr(A)$	Trace of matrix A
$P(\mathcal{A})$	Probability of event $\mathcal{A}$
$\ \bullet\ $	Euclidean norm of a vector or matrix
$\lceil \bullet \rceil$	Ceiling function
$\lfloor \bullet \rfloor$	Floor function
$ a $	Absolute value of real number a
$\text{int}(\mathbb{W})$	Interior of a closed set $\mathbb{W}$
$\det(A)$	Determinant of matrix A

$\triangleq$ “is defined as”

Sets

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^p$	Set of p -dimensional real column vectors
$\mathbb{R}_{>0}$	Set of all positive real numbers
$\mathbb{R}_{\geq 0}$	Set of all nonnegative real numbers
$\emptyset$	Empty set
$\mathbb{V}_i$	Voronoi cell generated by agent i
$\mathcal{N}_i$	Set of communication neighbors of agent i
$\mathcal{N}_{V_i}$	Set of Voronoi neighbors of agent i

Acronyms

MSN - Multi-Agent Sensor Networks

UAV - Unmanned Aerial Vehicles

a.s. - almost surely

Contents

Acknowledgements	i
Abstract	iii
Symbols and Acronyms	vii
List of Contents	ix
List of Figures	xv
List of Algorithms	xix
1 Introduction	1
1.1 Motivations and Objectives	1
1.2 Organization and Main Contributions of the Thesis	3
2 Literature Review	7
2.1 Overview of Distributed Control and Estimation in MSNs	7
2.2 Distributed Estimation and Fusion in MSNs	8
2.3 Distributed Control in MSNs	11

2.3.1	Topology Control	12
2.3.2	Coverage Control	12
2.3.3	Cooperative Control	13
2.4	Target Search and Localization in MSNs	15
3	Diffusion Kalman Filtering Based on Covariance Intersection	19
3.1	Background	20
3.1.1	System Description	20
3.1.2	Diffusion Kalman Filtering	21
3.2	CI-Based Adaptive Diffusion Matrix	23
3.3	CI-Based Diffusion Kalman Filtering Algorithm	25
3.3.1	Partial Local Uniform Observability	27
3.3.2	No Local Uniform Observability	31
3.4	Simulation	39
3.4.1	Simulation Environment	39
3.4.2	Simulation Results	41
3.5	Conclusions	43
4	Energy-Based Multiple Target Localization and Pursuit in Mobile Sensor Networks	47
4.1	Basic Assumptions and Definitions	48
4.1.1	Sensing Model	48
4.1.2	Assumptions	49

4.2	Continuous-Time Formulation	50
4.3	Discrete-Time Formulation	53
4.4	Target Pursuit Scheme	59
4.4.1	Pursuit Scheme	59
4.4.2	Convergence Analysis	61
4.5	Simulation	64
4.5.1	Simulation Environment and Parameters	64
4.5.2	Simulation Results	65
4.6	Conclusions	66
5	Multi-Agent Information Fusion and Cooperative Control in Target Search	71
5.1	Basic Definitions and Assumptions	72
5.2	Uncooperative Probability Map Update	73
5.3	Cooperative Probability Map Update	79
5.4	Path Planning and Asynchronous Implementations	85
5.4.1	Coverage Control	85
5.4.2	Connectivity Maintenance	89
5.4.3	Asynchronous Sampling and Communication	91
5.5	Simulation	94
5.5.1	Simulation Environment	94
5.5.2	Simulation Results	96
5.6	Conclusions	100

6	Vision-Based Multi-Agent Cooperative Target Search	103
6.1	Basic Definitions and Assumptions	104
6.2	Probability Map Update	105
6.2.1	Bayesian Update and Consensus-Based Information Fusion . .	105
6.2.2	Time-Varying Detection Probability	106
6.2.3	Environment-Based Probability Map	112
6.3	Cooperative Coverage Control	116
6.4	Simulation	122
6.4.1	Simulation Environment	122
6.4.2	Simulation Results	123
6.5	Conclusions	124
7	Adaptive Sensing in Multi-Agent Target Tracking	131
7.1	Basic Definitions and Assumptions	132
7.2	Iterative Estimation	133
7.2.1	Prediction Step:	133
7.2.2	Correction Step:	134
7.3	Cooperative Control for Adaptive Sensing	135
7.4	Discussions	141
7.5	Simulation	143
7.5.1	Simulation Environment	143
7.5.2	Simulation Results	144

7.6	Conclusions	144
8	Development of A Multi-agent Simulator and Testing Results	153
8.1	Design of the Simulation Platform	154
8.2	Experiments	156
8.3	Conclusions	159
9	Conclusions and Future Work	161
9.1	Conclusions	161
9.2	Future Work	163
	Appendix A Elementary Concepts and Notations	165
	Author's Publications	167
	Bibliography	169

List of Figures

3.1	Comparison between the DKF and CI-DKF algorithms.	22
3.2	Implementation of consensus protocol embedded in the Kalman filter for agent i	32
3.3	Simulation setup of Scenario I: full local uniform observability. . . .	42
3.4	Simulation setup of Scenario II: partial local uniform observability. .	42
3.5	Simulation setup of Scenario III: no local uniform observability. . . .	43
3.6	Results of Scenario I: full local uniform observability.	44
3.7	Results of Scenario II: partial local uniform observability.	45
3.8	Results of Scenario III: no local uniform observability.	46
4.1	Energy-based multiple target localization and pursuit.	48
4.2	Simulation setup.	67
4.3	The dynamic process of target pursuit.	68
4.4	The localization error for the three targets.	69
4.5	The boundness of the convergence region in terms of the spatial gra- dient.	69
5.1	Target search by multiple UAVs.	73

5.2	Surveillance region and target positions.	95
5.3	The convergence of the probability map of agent 1.	97
5.4	The results in Scenario I.	98
5.5	The results in Scenario II.	99
5.6	The results in Scenario III.	99
5.7	The results in Scenario IV.	100
5.8	The results in Scenario V.	101
5.9	The results in Scenario VI.	102
6.1	Target search by multiple UAVs.	104
6.2	Target image in the picture taken by an airborne camera.	107
6.3	The convergence of the probability map of an agent in Scenario I. . .	125
6.4	Snapshots of UAVs in Scenario I.	126
6.5	The convergence of the probability map of an agent in Scenario II. . .	127
6.6	Snapshots of UAVs in Scenario II.	128
6.7	Weight average $\boldsymbol{\eta}_k$	129
7.1	Snapshots of agents, target and estimate trajectories.	145
7.2	The estimation performance with 2 agents.	146
7.3	The estimation performance with 3 agents.	147
7.4	The estimation performance with 4 agents.	148
7.5	The estimation performance with 6 agents.	149
7.6	Convergence of geometric configuration.	150

8.1 The platform architecture. 154

8.2 Sensors on UAVs. 155

8.3 UAV model. 156

8.4 Simulating the real environment. 157

8.5 Virtual environment. 158

8.6 Environment for vision-based target search. 159

8.7 Starting positions of UAVs and roads. 160

8.8 Moving trajectories of the two UAVs. 160

List of Algorithms

3.1 CI-based diffusion Kalman filtering for partial local uniform observability 28

3.2 Information diffusion scheme 35

3.3 CI-based diffusion Kalman filtering for no uniform local observability . 36

4.1 Pursuit Scheme 60

Chapter 1

Introduction

1.1 Motivations and Objectives

Multi-agent sensor networks (MSNs) are comprised of multiple static or mobile agents capable of collecting, processing, storing and transferring information from one agent to another. Each agent can be equipped with inexpensive sensing devices to monitor a variety of environment conditions such as temperature, pressure, humidity, acoustic intensity, acceleration, magnetism, lighting conditions, and so on. These agents are able to autonomously form a network through which sensing results can be propagated. Since the agents have some computing capabilities, data can be processed as they flow through the network. The flexibility of installation and configuration of the sensing agents has resulted in wide applications of MSNs in various areas such as security, telecommunications and manufacturing, and has also motivated an enormous amount of research activities in the field of MSNs.

Recent developments in autonomous vehicle technologies are producing advanced surveillance systems, where the mobile agents are characterized by a high degree of functionality and reconfigurability. MSNs can play a critical role in several application domains such as landmine detection and identification, monitoring of endangered species, monitoring of urban environments, manufacturing plants, and civil

infrastructure, and intruder and target detection systems [1–3]. These networks are expected to operate cooperatively and reliably in cluttered dynamic environments with little human intervention. However, coordinating such large heterogeneous sensor networks is challenging and requires the development of novel methods of communication, motion control and planning, computation, proactive estimation and sensing, and power management.

Motivated by these new trends and challenges of MSNs, research work has been done to achieve the following objectives:

- To develop distributed estimation and fusion methods that allow agents in a network to incorporate local sensory information efficiently to produce local estimates and further improve the local estimates by fusing the estimates from neighbors. Most existing fusion methods are not suitable for the real-time estimation of time-varying system, since the global correlation of each local estimate is unknown by an individual agent especially when the network topology is time-varying. Moreover, in large distributed MSNs, due to the limited sensing and communication ranges of each agent, the local estimator for each agent may not be stable in the lack of sufficient measurement information (i.e., the detectability condition is not satisfied for the local system of each agent) and the stability performance has not been rigorously studied in this case. Thus, new estimation and fusion methods should be designed such that a stable estimate can be obtained for each agent when the detectability is lost. Though consensus-based methods [4–6] and delayed Kalman filter [7] can make the detectability condition satisfied for each agent by gathering the measurement information from all agents in the network, the communication and computation load may be pretty high, which is unnecessary when a global optimal estimate is not desired. Hence, it is meaningful to seek a more energy efficient but suboptimal fusion method that can lead to acceptable estimation performance but with much less energy cost.

- To develop distributed cooperative control methods for MSNs to fulfill given tasks such as the optimal coverage over the surveillance region in terms of sensing and detection, search and localization of multiple mobile targets and reconstruction of an unknown environmental model by adaptive sensing. This problem may become more complex when the observation model can not be explicitly obtained. For instance, the received signal is the energy intensity produced by multiple sources. On the other hand, instead of driving the agents into fixed topological or geometrical configurations as in many conventional robot control methods [8–12], the control strategy to be developed should be adapted to the time-varying sensing information obtained by each agent. For instance, a group of agents should adaptively adjust their communication topology and optimize their sampling positions in a real-time manner based on their detection results of the environment as time evolves. In such case, the estimation performance may also be affected by the movement of agents in addition to the obtained sensing information. Thus, the interactive influence between the path planning control and the information fusion should be analyzed.

1.2 Organization and Main Contributions of the Thesis

The thesis consists of seven chapters and one appendix. The contents and contributions of each chapter are summarized as follows.

Chapter 2: A literature review on historical developments and state-of-the-art technologies of distributed control and estimation in MSNs is given in this chapter.

Chapter 3: This chapter is concerned with distributed Kalman filtering for linear time-varying systems in MSNs. Motivated by the work of [13] which shows that the diffusion of the estimates of local Kalman filters can improve the estimation

performance of the whole network, a diffusion Kalman filtering algorithm based on covariance intersection is proposed which allows each agent to obtain a stable estimate by sharing information only with its neighbors. In addition, the diffusion scheme is generalized into more comprehensive cases, where the entire system and network topology can be time-varying, and the system may be unobservable via each agent together with its neighbors. Different from the algorithm proposed in [13] which fuses the estimates of local Kalman filters by a convex combination regardless of the error covariance information, our estimates are fused by the covariance intersection algorithm which incorporates the error covariance information as an important factor for stability assurance. Further, in the case that all the agents are locally unobservable, i.e., the estimates of all the agents are unstable merely based on the local measurement information, a consensus-based information diffusion scheme is designed to achieve the local observability within a finite time duration. It is proven that under certain connectivity condition on the time-varying network topology, the covariance of the estimation error of each agent by our method is bounded if the uniform observability condition for the system is satisfied under global measurements.

Chapter 4: In this chapter, a distributed target localization and pursuit scheme is proposed based on discrete measurements of the energy intensity field produced by mobile targets. The accurate observation model of such field is not available except some critical bounds. By our control strategy, all agents are categorized into two groups: the leaders, responsible for the target pursuit, and the followers, responsible for the formation and connectivity maintenance. The influence of the system parameters on the convergence of leaders to the local maximum points is analyzed. Finally, the proposed scheme is demonstrated by simulation.

Chapter 5: This chapter addresses the cooperative search for multiple stationary ground targets by a group of unmanned aerial vehicles (UAVs) with limited sensing and communication capabilities and two-dimensional dynamic motion models of the UAVs and the targets. In our setting, target existence within each cell is modeled

as the Bernoulli distribution. By observing a cell, each agent obtains a 0 (no target detected) or 1 (target detected) detection result with fixed detection and false-alarm probabilities. In addition, each agent can only communicate with the agents that are within its communication range and keeps an individual probability map for the whole region. First, by taking a transformation of the original probability map, the nonlinear Bayesian update is converted to a linear update of the transformed map, which reduces computational complexity. Second, for a group of connected agents sharing information, a consensus-like distributed scheme is proposed to fuse the individual probability maps for each agent, and its performance is analyzed for both synchronous and asynchronous implementations of sampling and communication. Finally, a path planning algorithm is designed for autonomous target search by a group of agents, which includes the coverage and topological control of the networked agents.

Chapter 6: This chapter deals with the vision-based cooperative search for multiple mobile ground targets by a group of UAVs with limited sensing and communication capabilities and a three-dimensional discrete-time motion model. First, a model of detection probability is proposed as a function of the height of an agent above the ground for vision-based target search. Based on the probability map updating model proposed in Chapter 5, the model is then generalized by considering the information decaying and transmission between cells due to environmental changes such as the target movement. The influence of the time-varying detection probability on the update of probability maps due to the three-dimensional UAV motion model is also analyzed. Then, a coverage optimization problem is formulated to balance the coverage area and detection performance. The proposed map updating model and cooperative control scheme are distributed, i.e., each agent only communicates with the agents within its communication range.

Chapter 7: This chapter investigates the adaptive sensing in three-dimensional target tracking in MSNs based on measurements from TDOA sensors. The main contribution is that our method can be applied with an arbitrary number of agents

that produce at least one measurement at each time. This decreases the cost of agent deployment and alleviates the request for a sufficient number of agents to make the Fisher information matrix invertible by using the CRLB. First, the framework of Kalman filtering is applied to iteratively estimate the target location which incorporates the Gauss-Newton method. Then, an optimal adaptive sensing strategy is given by minimizing the determinant of the estimation error covariance matrix, and a gradient-based control law is derived for each agent to reach a local optimum. A set of stationary points for local optimum geometric configurations of the agents is given. Furthermore, the connections between different optimization criteria are discussed including the CRLB. The proposed adaptive sensing strategy still can provide good tracking performance with only two agents, i.e., with only one TDOA measurement at each time, for estimation of a three-dimensional target location.

Chapter 8: In this chapter, a multi-purpose three dimensional simulation platform is designed for MSNs based on Unreal Engine, LabView, Matlab and OM-Net++. The platform is used to testify the performance of the proposed target search, information fusion and cooperative control algorithms in this thesis.

Chapter 9: This chapter concludes the whole thesis and presents some potential future work on the research topics of the thesis.

Chapter 2

Literature Review

2.1 Overview of Distributed Control and Estimation in MSNs

The continuous advancements in wireless communication and robotics technologies make it possible to implement multi-agent sensor networks (MSNs) in a variety of scenarios. In recent years, MSNs have been studied for applications in traffic control, battlefield surveillance, habitat monitoring and target localization [14–18]. Compared with the stationary sensor networks, robotic sensor networks have more advantages such as the adaptation to environmental changes and reconfigurability for better sensing performance. Extensive studies have been carried out on robotic sensor networks [19–21]. In the near future, large number of robots will coordinate their actions through ad-hoc communication networks and perform challenging tasks including manipulation in hazardous environments, exploration, environmental monitoring for intruder detection and interception.

As a consequence of the growing interest in MSNs, research on cooperative control has increased tremendously over the last few years. Key aspects of distributed cooperative control include formation [22–24], flocking [19, 25, 26], self-configuration

[27], swarm aggregation [28], gradient climbing [10], deployment and task allocation [11, 29–31], rendezvous [32–35], vehicle routing [36], and consensus [37, 38].

Estimation and fusion has been another major problem in MSNs and received a lot of attentions in the field of environmental monitoring and target tracking, etc [39–46]. There are two basic information fusion architectures, namely centralized and distributed architectures, depending on whether raw data are used directly for fusion or not. In the centralized fusion, all agents send their raw sensing data to a fusion center, which leads to the global optimal estimate. In the distributed fusion, each agent can only communicate with its neighbors and obtains an estimate which generally is suboptimal given the local information. However, compared with the centralized fusion, the distributed fusion has a less communication burden and higher survivability, and is more flexible and reliable for large-scale mobile multi-agent networks.

These advancements and potentials have triggered the present research interests in the two areas. This thesis is mainly focused on the application of MSNs in target search and localization, and focus on two main technical issues: the distributed cooperative control and the distributed estimation and fusion for target search and localization.

2.2 Distributed Estimation and Fusion in MSNs

Distributed estimation and fusion is an important topic in MSNs which has been studied for many applications such as environmental monitoring, surveillance, target tracking, etc [39–46]. In these applications, sensors installed on the agents can obtain measurements in a parallel manner. The communication and computing capabilities of each agent enable acquiring information from neighbors and processing it individually to get an estimate of a concerned physical quantity. The distributed processing greatly alleviates the computation load as taken by the fusion center in

the centralized fashion. Furthermore, the redundancy in a distributed sensor network by deploying a large number of sensors over the surveillance region makes the entire system more robust to the failures of some sensors.

Recently, the distributed Kalman filtering has received great attentions. In [47], a decentralized hierarchical structure is proposed, in which the estimates of local Kalman filters are calculated individually by each agent and then combined by a fusion center. A distributed Kalman filtering method using weighted averaging is proposed in [48], which requires the global information of the state covariances. Consensus-based distributed Kalman filters are proposed in [4–6], where local measurements are exchanged among neighbors in order for each agent to obtain the global optimal estimate given by a centralized estimation. However, communications between agents are required to be sufficiently fast such that consensus can be reached between two consecutive Kalman filter updates. In [13], a distributed Kalman filtering (DKF) algorithm is designed, where agents exchange their measurements as well as their pre-estimates. Measurements of neighboring agents are integrated to perform the local Kalman filtering and get a pre-estimate of the state, after which the pre-estimates of neighboring agents are fused locally by a convex combination to refine the pre-estimates. Different from the consensus approach, this new strategy does not rely on the consensus between two consecutive Kalman filter updates, which improves the efficiency of incorporating new measurement information. This diffusion strategy is of more practical use when dealing with dynamic systems, where new measurements must be processed in a timely manner instead of running consensus. In [49], the DKF algorithm is further developed with adaptive weights by optimizing a locally defined cost function. However, this adaptive rule cannot guarantee the stability of estimation error dynamics for each agent. Moreover, the estimation error covariance information is not taken into consideration in the choice of combination weights in [13] and [49], which can play an important role in improving the estimation performance.

Most of the existing distributed fusion methods involving the covariance informa-

tion are based on the assumption that the correlation between the estimates of two neighboring agents is known to both the agents [39–42]. However, in a decentralized network, each agent only has information about its local topology and neighbors' estimates, and the cross-correlation of the estimates between two agents is usually unknown. In [50], a channel filter is proposed for distributed estimation where the global topology is unknown to each agent and the globally optimal estimate can be obtained for each agent if a tree connected and stationary topology is imposed on the network. However, the channel filter cannot be applied in the case of dynamic and multi-path topologies, which makes the system sensitive to failures of some agents. Similar work has been delivered in [51] where the correlated information of signal and sensor data between agents is taken into consideration. However, the covariance information between the correlated estimates is required to be known in order to optimally fuse the sensor data and estimates. This assumption is usually hard to be satisfied under time-varying topologies where the communication links may be randomly changed. In [52, 53], the so-called Covariance Intersection (CI) algorithm is proposed for fusion of multiple consistent estimates with unknown correlations, which uses a convex combination of the estimates and chooses the combination weights by minimizing the trace or determinant of an upper bound of the error covariance matrix. [52, 54–56] apply the CI algorithm in the state estimation together with Kalman filter, where the estimates of individual Kalman filters are fused locally by the CI algorithm. However, no conclusion has been made on how to guarantee the estimation stability. [57] addressed the problem of decomposing the centralized Kalman filter into local filters, each of which undertakes the estimation of only part of the whole system state for sparsely connected large-scale dynamical systems. The local filter at each agent runs the estimation distributively, through which the computation load is cut down and shared by the agents while maintaining the same performance as that provided by the centralized filter. The proposed method is also based on a fixed network topology.

The main problem considered is how to get a stable estimate (i.e. with bounded

error covariance) for each agent in a distributed network with unknown and time-varying topologies. As mentioned above, one typical method, e.g., the consensus-based filtering method [6], is to let each agent collect enough measurement information from other agents through multi-hop communications and then compute an estimate by standard Kalman filtering, where only the measurement information is exchanged among all agents. Another typical method, e.g., the covariance intersection (CI) based algorithm [52], is to let each agent first compute an estimate based on its local measurement information by standard Kalman filtering and then fuse the estimates of all its neighbors locally to get an improved estimate, where only the estimates and error covariance matrices are exchanged among all agents. The former method can easily guarantee the estimation stability by collecting enough measurement information. However, the measurement information cannot be promptly fused until all required information is obtained. The latter one can incorporate measurement information immediately after receiving estimates from the neighbors of each agent instead of waiting for enough information to be collected through multi-hop communications, and bring in new information by fusing the estimates from different agents. However, since the covariances between the estimates are usually difficult to obtain under multi-path and time-varying topologies, such method cannot guarantee the estimation stability. Therefore, it is required to design a distributed filtering method which balances the advantages and disadvantages of the previous two.

2.3 Distributed Control in MSNs

Two categories of control problems in MSNs have been studied in this thesis: topology control and coverage control. Topology control mainly aims at the control of communication topology, while coverage control mainly aims at the control of coverage area in terms of some sensing criteria. In many applications, the two problems are combined. For example, in the collaborative coverage over a closed surveillance region, agents may have to know the information of other agents and adaptively

tune their communication topology and geometric formation to achieve the optimal coverage, which requires that the network be connected.

2.3.1 Topology Control

The problem of topology control for multi-agent networks is to set the radio range for each agent so as to minimize the energy usage, while still ensuring that the communication graph of all agents remains connected and satisfies some desirable communication properties [58]. However, for MSNs the problem of topology control is not only to set the radio range but also to keep the network connectivity using minimum energy while the agents move. Due to the mobility and power restriction of each agent, designing and maintaining a highly connected network topology is considerably complex. In general, the algorithms of topology control are classified with centralized algorithms and distributed algorithms.

Centralized algorithms can achieve optimal configuration of agents or its approximation, which are more applicable to stationary networks due to the lack of adaptability to topological changes [59]. In contrast, distributed algorithms are more suitable for mobile sensor networks since the environment is inherently dynamic and they are more adaptive to topological changes though at the cost of possibly degraded performance [60]. The basic aims of distributed topology control algorithms for multi-agent networks are to maintain network connectivity, conserve energy and increase the fault tolerance.

2.3.2 Coverage Control

The coverage control is one of the basic problems in MSNs. It is concerned with deploying the agents of a multi-agent network in order to maximize the total area covered by the agents in terms of certain sensing performance or to approximate a special distribution, while guaranteeing some constrained conditions: network

connection, minimum energy consumption, etc. There are many applications of coverage control such as target tracking, environment monitoring, and urban search and rescue.

In [11], a distributed and adaptive gradient descent algorithm is proposed for agent deployment based on Voronoi diagram. In this algorithm, a mobile robot coordinates with only its one-hop neighbors, defined by the Delaunay triangulation, but a global objective can be achieved. The topology of the multi-agent system can adaptively change with respect to different environments and task specifications. In [12], a control strategy is presented that allows a group of mobile robots to position themselves to optimize the sensory information about the environment. The agents use sensed information to estimate a function indicating the relative importance of different areas in the environment and their estimate is then used to drive the network to a desirable placement configuration. A decentralized and adaptive control law is presented in [61] to drive a network of mobile agents to an optimal sensing configuration for sensing in their environment. The controller uses an adaptive control architecture to learn a parameterized model of the environment and makes the agents move to the estimated centroid of their Voronoi regions. In [62], a dynamic awareness model is proposed to control a multi-vehicle sensor network with intermittent communications. The state of awareness of each individual vehicle is updated by its own sensing model and sharing information with its neighbors. In [63], a three dimensional distributed control strategy is proposed to deploy hovering robots with downward facing cameras to collectively monitor an environment. A new optimization criterion is defined as the information obtained by each pixel of a camera.

2.3.3 Cooperative Control

Instead of being treated as two separate problems, the topology control and coverage control are usually considered as a combined cooperative control problem and have

been studied in many applications. In addition, compared with the centralized control algorithms, distributed control algorithms are more robust to accidental failures of agents and breaks of communication links. Gradient following by autonomous vehicle systems inspired by bacterial chemotaxis has been explored in [8, 9]. In [28], a swarming control method is proposed, in which individuals balance their own gradient descent with inter-agent attraction and repulsion forces. However, this method requires each agent to know the gradient at its location and know the relative position of each of the other agents. In [64], a virtual leader approach to gradient climbing is taken, where the virtual leader finally can reach the local maximum point instead of the true agent. A coordinated control strategy is developed for a group of autonomous vehicles to descend or climb an environmental gradient using measurements of the environment together with relative position measurements of the nearest neighbors [65]. The local gradient information is still assumed to be known in this strategy. In [10], a stable control strategy is proposed for groups of agents to move and reconfigure cooperatively in response to a sensed environment. The underlying coordination framework uses the virtual bodies and artificial potentials mentioned in [64]. This strategy aims to seek out local maxima or minima in the environment field. However, each agent is required to implement a centralized processing of the measurements from the whole network. In [66], a distributed and fault tolerant control algorithm is designed for the cooperation and redeployment of mobile sensor networks such that the covered area can be enlarged, which combines the virtual potential method and the Delaunay triangulation. In [19], a theoretical framework is put forward for the design and analysis of distributed flocking algorithms for multi-agent networks. A group of agents can finally achieve an expected formation and reach a given moving rendezvous point by the proposed flocking algorithms. The optimal sensor placement and motion coordination strategies are studied in [67] for target tracking with range sensors in MSNs. In [68], the problem of environmental modeling is addressed using a proportional-integral average consensus estimator to fuse the local data of each individual agent to estimate the environment model. A control law is proposed for mobile agents to move to maxi-

mize their sensory information relative to current uncertainties in the model. In [69], a distributed cooperative control method is proposed to let robots reach peaks of an unknown scalar field, which is similar to the energy intensity field of targets. However, such method relies on the parameterization of the unknown field and the basis functions for the parameterization are assumed to be known, which makes it unsuitable for time-variant energy intensity fields of moving targets. A distributed Kriged Kalman filter is developed in [70] to estimate a spatio-temporal field. Gradient control laws are developed to move the mobile agents to critical points of the sensory field. Some other related works are: cooperative task allocation [71–74] and cooperative formation control [75–78].

2.4 Target Search and Localization in MSNs

Target search and localization have been among the top issues in the application field of cooperative control and information processing since MSNs emerged [79]. Due to the mobility of robotic agents, an MSN can autonomously adjust its position or topology based on its sensing information of the targets to achieve certain coverage distribution and accomplish the job of target search and localization.

With direct measurements of target position such as the distance and bearing to the target, it is easy to build up the observation model and thus solve the problem by conventional tracking methods such as Kalman Filtering. There have been tremendous work in single or multiple target tracking using conventional tracking methods [80–85]. The optimal two-dimensional relative sensor-target geometry in target localization is studied in [86–88]. The Cramer-Rao lower bound (CRLB) has been a common utility function for searching for the optimal geometric configuration of sensors [86–92]. The optimal geometric configuration formed by sensors and target for bearing-only target localization is addressed with equal sensor to target ranges for all sensors in [87, 91] and with arbitrary sensor to target ranges in [88, 90]. The optimal geometric sensor configuration for range-based target localization is

addressed in [88, 92]. The control of mobile sensors to track mobile targets while maintaining the optimal geometry is further investigated by minimizing the CRLB of target position estimate. However, the application of the CRLB requires that the number of sensors be sufficient to guarantee the invertibility of the Fisher information matrix. Some other optimization criteria are also considered in [86], such as a criterion based on the estimation error covariance matrix which includes the prior estimation information. In [93], adaptive sensing for target tracking by mobile sensors is further addressed by minimizing the determinant of the estimation error covariance matrix. However, only a suboptimal sensing strategy is given, and the noises of different measurements are assumed to be independent, which is not adequate for the general case, such as for the time-difference-of-arrival (TDOA) measurements. For the estimation of target location, different sorts of methods have been proposed, such as the conventional extended Kalman filtering (EKF) method [80], iterated Kalman filtering (IKF) method [94], nonlinear least-squares method and maximum likelihood method [86]. A discussion on the connections between these methods can be found in [95]. Compared with the range or bearing based methods using proactive sensors, range or bearing free methods using reactive sensors can be more energy conservative and adapted to critical environments, one good example of which is through the received signal strength or measured energy intensity. There have been some energy-based localization algorithms for the search of single static target by one or more static agents [96–99]. For search of single static target using mobile robots, an adaptive source localization method is proposed in [20], which assumes that the distance to the target can be directly measured. The multi-target localization based on the sensory energy intensity field is always a tough issue since each measurement is in fact a combined influence of all the targets. Due to the nonlinearity of the energy intensity with respect to the distance, it is very difficult to decode each individual component and the mostly used method is the maximum likelihood approach [100].

With the fast development of high resolution imaging devices and processing

technologies, unmanned autonomous vehicles installed with cameras are increasingly employed in civil and military applications such as environmental monitoring, battlefield surveillance and map building. Several vision-based target search and localization strategies have been proposed [101–107]. A conventional method for vision-based target search in a closed region is to divide the whole surveillance region into cells, and associate each cell with a probability or a confidence level of target existence in the cell which constitutes a probability map for the whole region [104–107]. In [108], a statistical framework is proposed for predicting the amount of time an agent should spend in a cell to increase the target detection confidence in that cell. The probability associated with each cell is updated based on the detection result which is 0 (no target detected) or 1 (target detected). However, a central probability map is required to store information from all agents. An online planning and control method is proposed in [109] for cooperative search by a group of UAVs. Each agent keeps an individual probability map for the whole region, which is updated based on Dempster-Shafer theory. Path planning for each agent is implemented based on the information obtained through search. However, this method requires full connectivity, i.e., each agent needs to directly communicate with all the other agents. In [110], a co-evolutionary approach of search path planning is proposed under constrained information sharing for multi-UAV systems. The communication bandwidth constraint is considered and a distributed path planning scheme is developed aiming at minimizing the team entropy over target existence within each cell. However, only the latest observation results are exchanged for information sharing, and the fusion of probability map of each agent is not considered which actually contains historical target information. Recently, a decentralized search algorithm is developed in [111] which includes a two-step updating procedure for the probability maps. Each agent first obtains observations over the cells within its coverage area and updates its individual probability map by Bayesian rule. Then, each agent transmits its individual probability map to its neighbors for map fusion. This algorithm is fully decentralized and full connectivity is not required. However, in a network which is not fully connected, each agent can only obtain information directly from its neigh-

boring agents. The lack of information correlation makes the map fusion difficult and only an intuitive fusion method is given in [111]. In [112], target detection is considered as part of an integrated mission which consists of coverage control and data collection as parallel tasks for multi-agent networks. The proposed coverage control method aims to maximize the joint detection probability of random events under limited field-of-view constraints. The collected data are incorporated based on the Bayesian rule for estimation of target existence. However, each agent only shares measurement information with its neighbors and has no access to information from agents other than its neighbors, which makes it hard to obtain the knowledge of the overall surveillance region. Specifically, target search and pursuit by ground vehicles have been addressed in [101–103]. Cooperative pursuit strategies to detect, intercept and capture intelligent targets in cluttered environments are described in [101]. In [102], the authors show that a pursuer can detect an arbitrary fast evader using a randomized strategy. The evader can be captured by two pursuers solving a *lion and man* problem assuming that at least one pursuer is as fast as the evader. In [103], a decentralized motion coordination algorithm is developed for tracking tasks of dynamic targets. In [113], a vision-based landing algorithm is designed and applied on the real system of an autonomous helicopter for searching and landing on a target, where the helicopter needs to find the target and recognize its geometric shape.

Chapter 3

Diffusion Kalman Filtering Based on Covariance Intersection

This chapter addresses the distributed target estimation problem for linear time-varying systems in MSNs. Following the work of [13] which showed that the diffusion of the estimates of local Kalman filters can improve the estimation performance of the whole network, a (covariance and intersection) CI-based diffusion Kalman filtering algorithm (CI-DKF) is proposed which allows each agent to obtain a stable estimate by sharing information only with its neighbors, where the entire system and network topology can be time-varying, and the system may be unobservable via each agent together with its neighbors. Different from the Diffusion Kalman Filtering (DKF) algorithm proposed in [13] which fuses the estimates of local Kalman filters by a convex combination regardless of the error covariance information, our estimates are fused by the CI algorithm which incorporates the error covariance information as an important factor for stability assurance.

First, the system model is given and the DKF algorithm is recalled in Section 3.1. In Section 3.2, choice rules of adaptive weights of the CI-DKF algorithm are designed based on the CI method. Then, a CI-DKF algorithm is proposed for the case where local observability may be lost in Section 3.3. The effectiveness of the

CI-DKF algorithm is testified by simulation in Section 3.4. Section 3.5 gives the conclusions.

3.1 Background

3.1.1 System Description

Consider a set of N agents with limited communication ranges which are spatially distributed over a surveillance region. Agent i takes measurement $y_{i,k} \in \mathbb{R}^q$ of a common environment state $x_k \in \mathbb{R}^p$ independently at time k . The state-space model associated with the environment and the measurement of agent i are respectively of the form:

$$\begin{aligned} x_{k+1} &= F_k x_k + w_k, \\ y_{i,k} &= H_{i,k} x_k + v_{i,k}, \end{aligned} \tag{3.1}$$

where w_k is the process noise and $v_{i,k}$ the measurement noise of agent i at time k . The matrices F_k and $H_{i,k}$ are allowed to be time-varying but bounded (A matrix is bounded if each element of the matrix has a lower bound and an upper bound). w_k and $v_{i,k}$ are assumed to be zero-mean, uncorrelated and white with

$$\mathbb{E} \begin{bmatrix} w_k \\ v_{i,k} \end{bmatrix} \begin{bmatrix} w_l \\ v_{j,l} \end{bmatrix}^T = \begin{bmatrix} Q_k \delta_{kl} & 0 \\ 0 & R_{i,k} \delta_{kl} \delta_{ij} \end{bmatrix},$$

where Q_k and $R_{i,k}$ are assumed to be positive definite and bounded. Further, w_k and $v_{i,k}$ are uncorrelated with the initial state x_0 . If the measurement information is processed in a centralized manner, the augmented forms is applied:

$$y_k = \begin{bmatrix} y_{1,k} \\ \vdots \\ y_{N,k} \end{bmatrix}, \quad H_k = \begin{bmatrix} H_{1,k} \\ \vdots \\ H_{N,k} \end{bmatrix}, \quad v_k = \begin{bmatrix} v_{1,k} \\ \vdots \\ v_{N,k} \end{bmatrix}. \tag{3.2}$$

The estimate of x_k obtained by agent i based on local observations up to time l is denoted as $\hat{x}_{i,k|l}$. The estimation error is denoted as $\tilde{x}_{i,k|l} \triangleq x_k - \hat{x}_{i,k|l}$. $P_{i,k|l}$ is an estimate of the error covariance matrix $E \left[\tilde{x}_{i,k|l} \tilde{x}_{i,k|l}^T \right]$ kept by agent i . Specifically, the estimate of agent i is said to be uniformly stable if there exists a bounded positive definite matrix $\bar{P}_i$ such that $E \left[\tilde{x}_{i,k|k} \tilde{x}_{i,k|k}^T \right] \leq \bar{P}_i$ (i.e., $\bar{P}_i - E \left[\tilde{x}_{i,k|k} \tilde{x}_{i,k|k}^T \right]$ is positive semidefinite) for all k . The graph of the network and other elementary notations are defined in Appendix A. Let the set $\{i_{m_k}\}$ ($m_k = 1, \dots, d_{i,k}$) denote the indices of the neighbors of agent i at time k . Then, the local observation matrix for each i can be defined as:

$$H_{i,k}^{\text{loc}} \triangleq [H_{i_1,k}^T, H_{i_2,k}^T, \dots, H_{i_{d_{i,k}},k}^T]^T. \quad (3.3)$$

3.1.2 Diffusion Kalman Filtering

Recently, a diffusion Kalman filtering (DKF) algorithm was proposed in [13] as shown in Fig. 3.1(a). Since our algorithm adopts the same information processing procedure as the DKF algorithm, which includes an incremental update (standard Kalman filtering update) step and a diffusion update step, it will be illustrated following the introduction of the DKF algorithm by showing the differences between the two algorithms.

The objective of the DKF is for every agent i in the network to compute a stable estimate of the unknown state x_k , while sharing data only with its neighbors. In the DKF algorithm, at every time instant k , agent i sends the quantities $H_{i,k}^T R_{i,k}^{-1} H_{i,k}$ and $H_{i,k}^T R_{i,k}^{-1} y_{i,k}$ to its neighbors for incremental update and the intermediate estimate $\psi_{i,k}$ for diffusion update. Having received the intermediate estimates from its neighbors, agent i combines the intermediate estimate $\psi_{j,k}$, $j \in \mathcal{N}_{i,k}$ in the diffusion update step by a $p \times p$ diffusion matrix $C_{i,j,k}$, $k, j \in \mathcal{V}$ as a weight which is subject to

$$\sum_{j \in \mathcal{N}_{i,k}} C_{i,j,k} = I, \quad C_{i,j,k} = 0 \text{ for } j \notin \mathcal{N}_{i,k}.$$

(a) The DKF algorithm.

(b) The CI-DKF algorithm.

Figure 3.1: Comparison between the DKF and CI-DKF algorithms.

On the basis of Kalman filtering which incorporates the real-time measurement information, the DKF algorithm improves the estimate of each agent by fusing the local estimates of its neighbors. A simplified convex combination is used for fusion of estimates with unknown correlations, which makes the algorithm applicable in a general distributed multi-agent network. Though an adaptive method of choosing combination weights is developed in [49], no conclusions have been made on how to guarantee the estimation stability for a general time-varying system. Furthermore, the assumptions in [13, 49] are restrictive since they require that F_k and $H_{i,k}^{\text{loc}}$ be time-invariant and $\{F, H_{i,k}^{\text{loc}}\}$ be detectable for each agent in the stability proof. This motivates us to develop a DKF algorithm with an effective diffusion strategy suitable for more general applications.

3.2 CI-Based Adaptive Diffusion Matrix

The diffusion matrices $C_{i,j,k}$ play an important role in the diffusion update which influence the performance of the whole network through assigning different weights to different local Kalman filter estimates at each iteration. In this section, the choice rules of diffusion matrices for fusion of estimates will be introduced first and then followed by the design of filtering algorithm. Heuristically speaking, a larger weight should be assigned to a local Kalman filter estimate with better estimation accuracy. Based on this idea, the CI algorithm is applied to incorporate the information of error covariance for choosing the diffusion matrices.

Suppose agent i is going to fuse the estimates $\hat{x}_{j,k|k-1}$, $j \in \mathcal{N}_{i,k}$ from its neighbors with the estimates of their error covariance matrices $P_{j,k|k-1} > 0$. The fusion based on the CI algorithm proposed in [52] is given by

$$\psi_{i,k} = \sum_{j \in \mathcal{N}_{i,k}} C_{i,j,k} \hat{x}_{j,k|k-1}, \quad (3.4)$$

where

$$\begin{aligned} C_{i,j,k} &= \beta_{i,j,k} \Lambda_{i,k} P_{j,k|k-1}^{-1}, \\ \Lambda_{i,k} &= \left(\sum_{j \in \mathcal{N}_{i,k}} \beta_{i,j,k} P_{j,k|k-1}^{-1} \right)^{-1}, \end{aligned} \quad (3.5)$$

and $\beta_{i,j,k}$ are subject to $0 \leq \beta_{i,j,k} \leq 1$ and $\sum_{j \in \mathcal{N}_{i,k}} \beta_{i,j,k} = 1$ which are chosen such that the trace or determinant of $\Lambda_{i,k}$ is minimized. However, such optimization is nonlinear and a high computation load is required for computing the optimal $\beta_{i,j,k}$. Several fast CI algorithms that produce suboptimal solutions have been proposed in terms of trace or determinant minimization [56, 114, 115]. For the sake of computational simplicity, the simplified algorithm proposed in [114] is used to calculate $\beta_{i,j,k}$ in (3.5). First, there always exists an agent $g_{i,k} \in \mathcal{N}_{i,k}$ at time k such that $\text{tr}(P_{g_{i,k},k|k-1}) = \min_{j \in \mathcal{N}_{i,k}} \text{tr}(P_{j,k|k-1})$. Then, $\beta_{i,j,k}$ is calculated as follows:

$$\beta_{i,j,k} = \begin{cases} \frac{1/\text{tr}(P_{j,k|k-1})}{\sum_{m \in \mathcal{N}_{i,k}} 1/\text{tr}(P_{m,k|k-1})}, & \text{if } \text{tr}(\Lambda_{i,k}) < \text{tr}(P_{g_{i,k},k|k-1}); \\ 1, & \text{if } \text{tr}(\Lambda_{i,k}) \geq \text{tr}(P_{g_{i,k},k|k-1}), j = g_{i,k}; \\ 0, & \text{otherwise.} \end{cases}$$

If $\text{tr}(\Lambda_{i,k})$ is ignored in the above equation for simplification of computation and let $\beta_{i,g_{i,k},k} = 1, \beta_{i,j,k} = 0 \forall j \neq g_{i,k}$, the following 0-1 weighting rule can be obtained:

$$\begin{aligned} C_{i,j,k} &= \begin{cases} I, & \text{if } j = g_{i,k}; \\ 0, & \text{otherwise,} \end{cases} \\ \Lambda_{i,k} &= P_{g_{i,k},k|k-1}. \end{aligned} \quad (3.6)$$

The weight choice rule (3.6) means that agent i takes the estimate of the neighbor which has the minimum trace of estimated prediction error covariance matrix as the best estimate. It is actually a special case of rule (3.5) and also provides a

suboptimal solution. However, it greatly simplifies the computation. In Section 3.4, performances of (3.5) and (3.6) will be compared by simulation.

It should be noted that $P_{j,k|k-1}$ does not have to be the true error covariance matrix of $\hat{x}_{j,k|k-1}$, and its influence on the estimation stability will be discussed later. By implementing the CI algorithm with rule (3.5) or (3.6), the following conclusion can be obtained.

Lemma 3.1 ([52]). *If $P_{j,k|k-1} > 0$ and $P_{j,k|k-1} \geq E \left[\tilde{x}_{j,k|k-1} \tilde{x}_{j,k|k-1}^T \right]$ for $j \in \mathcal{N}_{i,k}$, it holds that $\Lambda_{i,k} \geq E \left[\tilde{x}_{i,k|k-1} \tilde{x}_{i,k|k-1}^T \right]$.*

3.3 CI-Based Diffusion Kalman Filtering Algorithm

In [13], the stability of the DKF algorithm is proven under the assumption that the local Kalman filter for each agent is stable based on its local measurement information, i.e. the measurements from its neighbors which include the measurement of itself according to the definition of neighbors. This assumption is restrictive in general. Our aim is to seek methods such that each agent can obtain a stable estimate even when the estimation based only on local measurements is unstable.

Before designing the algorithm, first recall the definition of uniform observability of linear time-varying systems [116]. Consider the system with time-varying matrix F_k and global measurement matrix H_k defined in (3.2) and let the observability Gramian be given by

$$\overline{W}_{k+\delta,k} = \sum_{t=k}^{k+\delta} \Phi_{t,k}^T H_t^T H_t \Phi_{t,k} \quad (3.7)$$

for some integer $\delta > 0$, where $\Phi_{k,k} = I$ and

$$\Phi_{t,k} = F_{t-1} \cdots F_k$$

for $t > i$. The matrices F_k and H_k are said to satisfy the uniform observability condition and the entire system is said to be uniformly observable, if there are real numbers $\underline{\eta}, \bar{\eta} > 0$ and an integer $\delta > 0$, such that

$$\underline{\eta}I \leq \overline{W}_{k+\delta,k} \leq \bar{\eta}I. \quad (3.8)$$

In the same way, the local uniform observability can be defined for each agent. Consider agent i with time-varying matrix F_k and local measurement matrix $H_{i,k}^{\text{loc}}$ defined in (3.3), and let the local observability Gramian be given by

$$W_{k+\delta_i,k}^k = \sum_{t=k}^{k+\delta_i} \Phi_{t,k}^T (H_{i,t}^{\text{loc}})^T H_{i,t}^{\text{loc}} \Phi_{t,k} \quad (3.9)$$

for some integer $\delta_i > 0$. The matrices F_k and $H_{i,k}^{\text{loc}}$ are said to satisfy the uniform observability condition and the local subsystem of agent i is said to be uniformly observable, if there are real numbers $\underline{\eta}_i, \bar{\eta}_i > 0$ and an integer $\delta_i > 0$, such that

$$\underline{\eta}_i I \leq W_{k+\delta_i,k}^k \leq \bar{\eta}_i I. \quad (3.10)$$

In a network, (3.10) may not hold for all agents. First, denote by Ω the set of agents for which (3.10) holds, i.e., $i \in \Omega$ if F_k and $H_{i,k}^{\text{loc}}$ satisfy the local uniform observability condition, and $i \notin \Omega$ otherwise. In the sequel, the problem will be discussed under two scenarios, partial local uniform observability ($\Omega \neq \emptyset$ and at least one agent i can determine $i \in \Omega$) and no local uniform observability ($\Omega = \emptyset$ or each agent i cannot determine $i \in \Omega$).

Remark 3.1. *In the case of partial local uniform observability, there exists at least one agent in the network which can obtain stable estimates merely based on the measurement information from its neighbors without the need of acquiring information from other agents. However, in the case of no local uniform observability, i.e., none of the agents satisfies the local uniform observability condition, there may not exist an agent which can give a stable estimate.*

Note that the uniform observability condition may be weakened. For example, for linear time-invariant systems, the observability condition can be replaced by detectability condition. A system is detectable if and only if all of its unobservable modes are stable. For a detectable system, it can be decomposed into two subsystems, one with all observable modes which may not be a stable system and the other with all unobservable modes which is a stable system. Then, it is needed only to consider the estimation stability of the subsystem with all observable modes since the estimation error covariance matrix for the stable subsystem is always bounded no matter what initial estimate is made.

3.3.1 Partial Local Uniform Observability

In this case, each agent $i \in \Omega$ can give a stable estimate only based on the measurement information from its neighbors. Our main task is to let all the other agents $j \notin \Omega$ obtain stable estimates by diffusion of local estimates. To this end, the CI-based diffusion Kalman filtering (CI-DKF) algorithm is proposed for the case of partial local uniform observability as shown in Fig. 3.1(b) and Algorithm 3.1. The CI-DKF algorithm requires that at every instant k , each agent communicate to its neighbors the quantities $H_{i,k}^T R_{i,k}^{-1} H_{i,k}$, $H_{i,k}^T R_{i,k}^{-1} y_{i,k}$, $\hat{x}_{i,k|k-1}$ and $P_{i,k|k-1}$ by one message. After receiving the messages from neighbors, each agent first fuses the estimates $\hat{x}_{j,k|k-1}$ for $j \in \mathcal{N}_{i,k}$ in the diffusion update step based on the CI algorithm introduced in Section 3.2. Then, the measurement information is incorporated by Kalman filtering in the incremental update step. Different from the DKF algorithm, the CI-DKF algorithm (Algorithm 3.1) does not require each agent to communicate an intermediate estimate, which reduces the communication load as each message contains significant overhead information. However, the estimation error covariance matrix is communicated in the CI-DKF algorithm in order to get a stable estimate for each agent. The following theorem shows that the estimation stability can be obtained by implementing the CI-DKF algorithm in the case of partial local uniform observability.

Algorithm 3.1: CI-based diffusion Kalman filtering for partial local uniform observability

Start with $\hat{x}_{i,0|-1} = 0$, $P_{i,0|-1} = \Pi_0 > 0$ and $i = 0$ for all i :

Step 1: Diffusion Update:

Take measurement and communicate $H_{i,k}^T R_{i,k}^{-1} H_{i,k}$, $H_{i,k}^T R_{i,k}^{-1} y_{i,k}$, $\hat{x}_{i,k|k-1}$ and $P_{i,k|k-1}$ to neighbors;

Calculate $\Lambda_{i,k}$ and the diffusion matrix $C_{i,j,k}$ by (3.5) or (3.6);

If $i \in \Omega$ & $P_{i,k|k-1} \geq \Lambda_{i,k}$,

$$\hat{x}_{i,k|k-1} \leftarrow \sum_{j \in \mathcal{N}_{i,k}} C_{i,j,k} \hat{x}_{j,k|k-1}$$

$$P_{i,k|k-1} \leftarrow \Lambda_{i,k}$$

else if $k \notin \Omega$,

$$\hat{x}_{i,k|k-1} \leftarrow \sum_{j \in \mathcal{N}_{i,k}} C_{i,j,k} \hat{x}_{j,k|k-1}$$

$$P_{i,k|k-1} \leftarrow \Lambda_{i,k}$$

end if

Step 2: Incremental Update:

$$S_{i,k} = \sum_{j \in \mathcal{N}_{i,k}} H_{j,k}^T R_{j,k}^{-1} H_{j,k}$$

$$q_{i,k} = \sum_{j \in \mathcal{N}_{i,k}} H_{j,k}^T R_{j,k}^{-1} y_{j,k}$$

$$P_{i,k|k}^{-1} = P_{i,k|k-1}^{-1} + S_{i,k}$$

$$\hat{x}_{i,k|k} = \hat{x}_{i,k|k-1} + P_{i,k|k} [q_{i,k} - S_{i,k} \hat{x}_{i,k|k-1}]$$

$$\hat{x}_{i,k+1|k} = F_k \hat{x}_{i,k|k}$$

$$P_{i,k+1|k} = F_k P_{i,k|k} F_k^T + Q_k$$

$$k \leftarrow k + 1.$$

Theorem 3.1. *With $P_{j,0|-1} > 0$ and $P_{j,0|-1} \geq \mathbb{E} [\tilde{x}_{j,0|-1} \tilde{x}_{j,0|-1}^T]$ for $j \in \mathcal{V}$, if there exists at least one agent $i \in \Omega$ and the network is connected all the time, then the estimates of all agents are uniformly stable under Algorithm 3.1.*

Proof. First, consider agent $i \in \Omega$ which satisfies the local uniform observability condition. $P_{i,k|k-1}$ is bounded if only the incremental update is executed for agent i based on the measurement information from its neighbors with bounded matrices F_k and Q_k , i.e., there exists a matrix such that $P_{i,k|k-1} \leq \bar{P}_i$ for all k . Since the diffusion update for $i \in \Omega$ replaces $P_{i,k|k-1}$ with the matrix $\Lambda_{i,k} \leq P_{i,k|k-1}$, $P_{i,k|k-1}$ is bounded during each iteration of Algorithm 3.1 with the diffusion update step. Thus, for all k we have

$$P_{i,k|k} = \left(P_{i,k|k-1}^{-1} + S_{i,k} \right)^{-1} \leq P_{i,k|k-1} \leq \bar{P}_i.$$

Now, consider agent $j_1 \in \mathcal{N}_{i,k}$. According to Lemma 3.1, after the replacement in diffusion update, i.e., $P_{j,k|k-1} = \Lambda_{j,k}$ for $j \in \mathcal{V}$, we have

$$\begin{aligned} \text{tr} (P_{j_1,k|k-1}) &= \text{tr} (\Lambda_{j_1,k}) \leq \text{tr} (F_{i-1} P_{i,k-1|k-1} F_{i-1}^T + Q_{i-1}) \\ &\leq \text{tr} (F_{i-1} F_{i-1}^T) \text{tr} (P_{i,k-1|k-1}) + \text{tr} (Q_{i-1}) \\ &\leq \text{tr} (F_{i-1} F_{i-1}^T) \text{tr} (\bar{P}_i) + \text{tr} (Q_{i-1}). \end{aligned}$$

Since F_k and Q_k are bounded, there exist finite real numbers π_F and π_Q such that $\text{tr} (F_k^T F_k) \leq \pi_F$ and $\text{tr} (Q_k) \leq \pi_Q$ for all k . Thus, we can get

$$P_{j_1,k|k-1} \leq [\text{tr} (F_{i-1}^T F_{i-1}) \text{tr} (\bar{P}_i) + \text{tr} (Q_{i-1})] I \leq [\pi_F \text{tr} (\bar{P}_i) + \pi_Q] I$$

and

$$P_{j_1,k|k} = \left(P_{j_1,k|k-1}^{-1} + S_{i,k} \right)^{-1} \leq P_{j_1,k|k-1} \leq [\pi_F \text{tr} (\bar{P}_i) + \pi_Q] I \triangleq \bar{P}_{j_1}.$$

Similarly, for agent $j_2 \in \mathcal{N}_{j_1, k}$, we have

$$\text{tr} (P_{j_2, k|k-1}) = \text{tr} (\Lambda_{j_2, k}) \leq \text{tr} (F_{i-1}^T F_{i-1}) \text{tr} (\bar{P}_{j_1}) + \text{tr} (Q_{i-1})$$

which implies

$$P_{j_2, k|k-1} \leq [\pi_F \text{tr} (\bar{P}_i) + \pi_Q] I$$

and

$$P_{j_2, k|k} \leq P_{j_2, k|k-1} \leq [\pi_F \text{tr} (\bar{P}_i) + \pi_Q] I \triangleq \bar{P}_{j_2}.$$

In the same way, an upper bound $\bar{P}_j$ of the estimated error covariance matrix $P_{j, k|k}$ can be found for any other agent $j \in \mathcal{V}$.

On the other hand, if $P_{j, k|k-1} \geq \mathbb{E} [\tilde{x}_{j, k|k-1} \tilde{x}_{j, k|k-1}^T]$ holds for $j \in \mathcal{N}_{i, k}$ before the diffusion update during the k -th iteration, the CI algorithm guarantees that $\Lambda_{i, k} \geq \mathbb{E} [\tilde{x}_{j, k|k-1} \tilde{x}_{j, k|k-1}^T]$ according to Lemma 3.1, which implies that $P_{j, k|k-1} \geq \mathbb{E} [\tilde{x}_{j, k|k-1} \tilde{x}_{j, k|k-1}^T]$ still holds after the diffusion update within the same iteration. Following that, the incremental update in Algorithm 3.1 gives

$$\begin{aligned} P_{i, k|k}^{-1} &= P_{i, k|k-1}^{-1} + S_{i, k} \\ \hat{x}_{i, k|k} &= \hat{x}_{i, k|k-1} + P_{i, k|k} [q_{i, k} - S_{i, k} \hat{x}_{i, k|k-1}] \\ P_{i, k+1|k} &= F_k P_{i, k|k} F_k^T + Q_k \end{aligned} \tag{3.11}$$

Then, we have

$$\begin{aligned} \tilde{x}_{i, k|k} &= (I - P_{i, k|k} S_{i, k}) \tilde{x}_{i, k|k-1} - P_{i, k|k} (q_{i, k} - S_{i, k} x_k) \\ &= P_{i, k|k} P_{i, k|k-1}^{-1} \tilde{x}_{i, k|k-1} - P_{i, k|k} \sum_{j \in \mathcal{N}_{i, k}} H_{j, k}^T R_{j, k}^{-1} v_{j, k}. \end{aligned}$$

$\tilde{x}_{i,k|k-1}$ and $v_{j,k}$ are uncorrelated, which implies

$$\begin{aligned} \mathbb{E} [\tilde{x}_{i,k|k} \tilde{x}_{i,k|k}^T] &= P_{i,k|k} P_{i,k|k-1}^{-1} \mathbb{E} [\tilde{x}_{i,k|k-1} \tilde{x}_{i,k|k-1}^T] P_{i,k|k-1}^{-1} P_{i,k|k} \\ &\quad + P_{i,k|k} \sum_{j \in \mathcal{N}_{i,k}} H_{j,k}^T R_{j,k}^{-1} H_{j,k} P_{i,k|k} \\ &\leq P_{i,k|k} P_{i,k|k-1}^{-1} P_{i,k|k} + P_{i,k|k} S_{i,k} P_{i,k|k} = P_{i,k|k}. \end{aligned} \quad (3.12)$$

We can further get

$$\mathbb{E} [\tilde{x}_{i,k+1|k} \tilde{x}_{i,k+1|k}^T] = F_k \mathbb{E} [\tilde{x}_{i,k|k} \tilde{x}_{i,k|k}^T] F_k^T + Q_k \leq F_k P_{i,k|k} F_k^T + Q_k = P_{i,k+1|k}.$$

with the initial condition $\mathbb{E} [\tilde{x}_{i,0|-1} \tilde{x}_{i,0|-1}^T] \leq P_{i,0|-1}$ for $j \in \mathcal{V}$, $P_{j,k|k} \geq \mathbb{E} [\tilde{x}_{j,k|k} \tilde{x}_{j,k|k}^T]$ holds through all iterations. Consequently, $\mathbb{E} [\tilde{x}_{j,k|k} \tilde{x}_{j,k|k}^T] \leq \bar{P}_j$ holds for all k , which implies Theorem 3.1 holds. $\square$

3.3.2 No Local Uniform Observability

In the case of no local uniform observability, one choice to make the DI-DKF algorithm applicable is to build the local uniform observability for some agents in the network. The key to building the observability is to get enough measurement information of the state. In Algorithm 3.1, the local measurement information of agent i is contained in $S_{i,k}$ and $q_{i,k}$, and the global information matrix and information vector which include the measurement information of all agents are respectively:

$$\begin{aligned} \bar{S}_k &= \sum_{i=1}^N H_{i,k}^T R_{i,k}^{-1} H_{i,k} = H_k^T R_k^{-1} H_k \\ \bar{q}_k &= \sum_{i=1}^N H_{i,k}^T R_{i,k}^{-1} y_{i,k} = H_k^T R_k^{-1} y_k \end{aligned} \quad (3.13)$$

To obtain a stable estimate, each agent should obtain sufficient measurement information through one-hop and/or multi-hop communications. However, communicating and storing raw measurements information from each agent would bring heavy

communication and storage burdens for agents. In recent years, many consensus approaches have been proposed for distributed information diffusion [5, 6, 37, 38, 117]. By these approaches, the information matrix $S_{i,k}$ and the information vector $q_{i,k}$ based on the information collected by agent i are updated iteratively with the initial values respectively as $H_{i,k}^T R_{i,k}^{-1} H_{i,k}$ and $H_{i,k}^T R_{i,k}^{-1} y_{i,k}$ during each sampling interval, and converge to $\bar{S}_k/N$ and $\bar{q}_k/N$ respectively in an asymptotic manner as the number of communication cycles within each sampling interval goes to infinity. During each communication cycle, only the updated $S_{i,k}$ and $q_{i,k}$ are sent and stored in the memory instead of the raw measurements information from other agents, which helps the agents lower the communication burden and save much storage space. The finite-time consensus for continuous-time systems is shown to be achievable in [77]. However, there has been no effective method to achieve the finite-time consensus for discrete-time systems. A consensus protocol is used for distributed Kalman filtering in [5], where each agent implements a consensus protocol to gather measurement information between two successive Kalman filter updates as shown in Fig. 3.2. However, by this approach, $S_{i,k}$ and $q_{i,k}$ may not have achieved consensus, i.e. converge to $\bar{S}_k/N$ and $\bar{q}_k/N$ before the next Kalman filter update, and the error caused by treating $S_{i,k}$ and $q_{i,k}$ as $\bar{S}_k$ and $\bar{q}_k$ respectively may destroy the estimation stability.

Figure 3.2: Implementation of consensus protocol embedded in the Kalman filter for agent i .

In this chapter, the distributed consensus protocol proposed in [117] is adopted for information diffusion between two successive Kalman filter updates (Fig. 3.2),

but scale the updated information by multiplying an appropriate coefficient at the end of the consensus implementation. In [117], each agent aims to let $S_{i,k}$ and $q_{i,k}$ reach a consensus by the following protocol:

$$\begin{aligned} S_{i,k}(l+1) &= \left(1 - \frac{d_{i,k}(l+1) - 1}{N}\right) S_{i,k}(l) + \frac{1}{N} \sum_{\substack{j \in \mathcal{N}_{i,k}(l+1) \\ j \neq i}} S_{j,k}(l) \\ q_{i,k}(l+1) &= \left(1 - \frac{d_{i,k}(l+1) - 1}{N}\right) q_{i,k}(l) + \frac{1}{N} \sum_{\substack{j \in \mathcal{N}_{i,k}(l+1) \\ j \neq i}} q_{j,k}(l) \end{aligned} \quad (3.14)$$

where k refers to the k -th sampling interval and l the l -th communication cycle, $d_{i,k}(l+1)$ is the number of neighbors of agent i including itself at the time instant of the $(l+1)$ -th communication cycle, and $S_{i,k}(0) = H_{i,k}^T R_{i,k}^{-1} H_{i,k}$, $q_{i,k}(0) = H_{i,k}^T R_{i,k}^{-1} y_{i,k}$. Then, we can get

$$\begin{aligned} S_{i,k}(l) &= \sum_{j=1}^N \rho_{i,j,k}(l) S_{i,k}(0) \\ q_{i,k}(l) &= \sum_{j=1}^N \rho_{i,j,k}(l) q_{i,k}(0) \end{aligned} \quad (3.15)$$

where $\rho_{i,i,k}(0) = 1$ and $\rho_{i,j,k}(0) = 0$ for $j \neq i$. From (3.14), it is easy to get

$$\rho_{i,j,k}(l+1) = \left(1 - \frac{d_{i,k}(l+1) - 1}{N}\right) \rho_{i,j,k}(l) + \sum_{\substack{m \in \mathcal{N}_{i,k}(l+1) \\ m \neq k}} \frac{\rho_{m,l,k}(l)}{N} \quad (3.16)$$

and $\rho_{i,j,k}$ are the weights satisfying

$$0 \leq \rho_{i,j,k}(l) \leq 1, \quad \rho_{i,j,k}(l) = \rho_{j,k,k}(l), \quad \sum_{j=1}^N \rho_{i,j,k}(l) = 1.$$

Since $\rho_{i,j,k}(l)$ for $j \in \mathcal{N}_{i,k}(l)$ can be obtained by agent i at time $l+1$ through communications with its neighbors, $\rho_{i,j,k}(l)$ ($i, j \in \mathcal{V}$) can be computed iteratively according to (3.16). It has been proven [117] that $\lim_{j \rightarrow \infty} \rho_{i,j,k}(l) = 1/N$ for $i, j \in \mathcal{V}$, thus $S_{i,k}(l) \rightarrow \bar{S}_k/N$ and $q_{i,k}(l) \rightarrow \bar{q}_k/N$ as $l \rightarrow \infty$.

Lemma 3.2. *If the network is connected all the time, then there exists a time $0 < t_o \leq N - 1$ such that for all k and $j \geq t_o$,*

$$\rho_{i,j,k}(l) \geq \left(\frac{1}{N}\right)^{t_o} \geq \left(\frac{1}{N}\right)^{N-1}$$

where $i, j \in \mathcal{V}$.

Proof. Let $t_{i,j,k}$ denote the first time at which $\rho_{i,j,k}(j)$ becomes non-zero. Since the information is transmitted in a diffusion way and the network is connected all the time, $t_{i,j,k}$ must exist and is less than N , i.e., $\max_{i,j} t_{i,j,k} \leq N - 1$ for all k . Moreover, by the definition of $t_{i,j,k}$, we have $\rho_{i,j,k}(t_{i,j,k}) = (1/N)^{t_{i,j,k}}$ for $i \neq j$. Then, for $t \geq t_{i,j,k}$ and $i \neq j$, we have

$$\rho_{i,j,k}(t) \geq \left(\frac{1}{N}\right)^{t_{i,j,k}} \prod_{l=t_{i,j,k}+1}^t \left(1 - \frac{d_{k,i}(l) - 1}{N}\right) \geq \left(\frac{1}{N}\right)^t.$$

For $i = j$, we have

$$\rho_{i,i,k}(t) \geq \prod_{l=1}^t \left(1 - \frac{d_{k,i}(l) - 1}{N}\right) \geq \left(\frac{1}{N}\right)^t.$$

Hence, by letting $t_o = \max_{i,j \in \mathcal{V}} t_{i,j,k}$, we have $\rho_{i,j,k}(t_o) \geq (1/N)^{t_o} \geq (1/N)^{N-1}$ for all i and j , which implies $\rho_{i,j,k}(l) \geq (1/N)^{N-1}$ for $l \geq t_o$ according to (3.14). $\square$

Remark 3.2. *Lemma 3.2 shows that one can always find a finite time length t_o for running the consensus protocol during each communication cycle such that the scaled measurement information from all agents can be collected by each agent, though the collected information from each agent is scaled by weight $\rho_{i,j,k}(t_o)$ rather than the original. In addition, it shows that the choice of $t_o = N - 1$ guarantees that $\rho_{i,j,k}(t_o) > 0$ for all i and j .*

Summarizing the results above, an information diffusion scheme (Algorithm 3.2) is designed, where t_o is a given time length for running the consensus protocol

during each communication cycle. Then, we can define a set Υ , where $i \in \Upsilon$ if $\min_{j \in \mathcal{V}} \rho_{i,j,k}(t_o) > 0$ for all k , and $i \notin \Upsilon$ otherwise. With Algorithm 3.2 embedded, the CI-DKF algorithm for the case of no local observability is shown in Algorithm 3.3 which includes an information diffusion step, a diffusion update step and an incremental update step. At every time instant k , each agent i first takes measurement and collects measurement information from other agents to get $S_{i,k}$ and $q_{i,k}$ by implementing the information diffusion scheme in the information diffusion step. In the meantime, each agent i obtains the prediction estimates $\hat{x}_{i,k|k-1}$ and $P_{i,k|k-1}$ from its neighbors, which is followed by the same steps as in Algorithm 3.1 for diffusion and incremental updates.

Algorithm 3.2: Information diffusion scheme

Start with $S_{i,k}(0) = H_{i,k}^T R_{i,k}^{-1} H_{i,k}$, $q_{i,k}(0) = H_{i,k}^T R_{i,k}^{-1} y_{i,k}$, $\rho_{i,i,k}(0) = 1$, $\rho_{i,j,k}(0) = 0$ ($j \in \mathcal{V}$, $j \neq i$) and $j = 1$ for all i :

Repeat the following steps until $j > t_o$:

Calculate $S_{i,k}(l)$ and $q_{i,k}(l)$ by (3.14);

Calculate $\rho_{i,j,k}(l)$ ($j \in \mathcal{V}$) by (3.16);

$l \leftarrow l + 1$

end

$$S_{i,k} \leftarrow \frac{1}{\max_{j \in \mathcal{V}} \rho_{i,j,k}(t_o)} S_{i,k}(t_o)$$

$$q_{i,k} \leftarrow \frac{1}{\max_{j \in \mathcal{V}} \rho_{i,j,k}(t_o)} q_{i,k}(t_o).$$

Theorem 3.2. With $P_{j,0|-1} > 0$ and $P_{j,0|-1} \geq \mathbb{E} [\tilde{x}_{j,0|-1} \tilde{x}_{j,0|-1}^T]$ for $j \in \mathcal{V}$, if there exists at least one agent $i \in \Upsilon$ and the network is connected all the time, then the estimates of all agents are uniformly stable under Algorithm 3.3 if F_k and H_k satisfy the uniform observability condition.

Proof. With $P_{i,k|k-1} \geq \mathbb{E} [\tilde{x}_{i,k|k-1} \tilde{x}_{i,k|k-1}^T]$ before the diffusion update, it still holds after the diffusion update according to Lemma 3.1. Following that, the incremental

Algorithm 3.3: CI-based diffusion Kalman filtering for no uniform local observability

Start with $x_{i,0|-1} = 0$, $P_{i,0|-1} = \Pi_0 > 0$ and $i = 0$ for all i :

Step 1: Information Diffusion:

Take measurement, communicate $\hat{x}_{i,k|k-1}$ and $P_{i,k|k-1}$ to neighbors and implement Algorithm 3.2

to get $S_{i,k}$, $q_{i,k}$ and $\rho_{i,j,k}$ ($j \in \mathcal{V}$);

Step 2: Diffusion Update:

Calculate $\Lambda_{i,k}$ and the diffusion matrix $C_{i,j,k}$ by (3.5) or (3.6);

If $i \in \Upsilon$ & $P_{i,k|k-1} \geq \Lambda_{i,k}$

$$\hat{x}_{i,k|k-1} \leftarrow \sum_{j \in \mathcal{N}_{i,k}} C_{i,j,k} \hat{x}_{j,k|k-1}$$

$$P_{i,k|k-1} \leftarrow \Lambda_{i,k}$$

else if $i \notin \Upsilon$,

$$\hat{x}_{i,k|k-1} \leftarrow \sum_{j \in \mathcal{N}_{i,k}} C_{i,j,k} \hat{x}_{j,k|k-1}$$

$$P_{i,k|k-1} \leftarrow \Lambda_{i,k}$$

end if

Step 3: Incremental Update:

$$P_{i,k|k}^{-1} = P_{i,k|k-1}^{-1} + S_{i,k}$$

$$\hat{x}_{i,k|k} = \hat{x}_{i,k|k-1} + P_{i,k|k} [q_{i,k} - S_{i,k} \hat{x}_{i,k|k-1}]$$

$$\hat{x}_{i,k+1|k} = F_k \hat{x}_{i,k|k}$$

$$P_{i,k+1|k} = F_k P_{i,k|k} F_k^T + Q_k$$

$$k \leftarrow k + 1.$$

update in Algorithm 3.3 gives

$$\begin{aligned} P_{i,k|k}^{-1} &= P_{i,k|k-1}^{-1} + S_{i,k} \\ \hat{x}_{i,k|k} &= \hat{x}_{i,k|k-1} + P_{i,k|k} [q_{i,k} - S_{i,k} \hat{x}_{i,k|k-1}] . \\ P_{i,k+1|k} &= F_k P_{i,k|k} F_k^T + Q_k \end{aligned} \tag{3.17}$$

Then, we have

$$\begin{aligned} \tilde{x}_{i,k|k} &= (I - P_{i,k|k} S_{i,k}) \tilde{x}_{i,k|k-1} - P_{i,k|k} (q_{i,k} - S_{i,k} x_k) \\ &= P_{i,k|k} P_{i,k|k-1}^{-1} \tilde{x}_{i,k|k-1} - P_{i,k|k} \sum_{j=1}^N \frac{\rho_{i,j,k}(t_o)}{\max_{j \in \mathcal{V}} \rho_{i,j,k}(t_o)} H_{j,k}^T R_{j,k}^{-1} v_{j,k}. \end{aligned}$$

Since $\tilde{x}_{i,k|k-1}$ and $v_{j,k}$ are uncorrelated, it is implied that

$$\begin{aligned}
\mathbb{E} [\tilde{x}_{i,k|k} \tilde{x}_{i,k|k}^T] &= P_{i,k|k} P_{i,k|k-1}^{-1} \mathbb{E} [\tilde{x}_{i,k|k-1} \tilde{x}_{i,k|k-1}^T] P_{i,k|k-1}^{-1} P_{i,k|k} \\
&\quad + P_{i,k|k} \sum_{j=1}^N \left[\frac{\rho_{i,j,k}(t_o)}{\max_{j \in \mathcal{V}} \rho_{i,j,k}(t_o)} \right]^2 H_{j,k}^T R_{j,k}^{-1} H_{j,k} P_{i,k|k} \\
&\leq P_{i,k|k} P_{i,k|k-1}^{-1} P_{i,k|k} + P_{i,k|k} \sum_{j=1}^N \frac{\rho_{i,j,k}(t_o)}{\max_{j \in \mathcal{V}} \rho_{i,j,k}(t_o)} H_{j,k}^T R_{j,k}^{-1} H_{j,k} P_{i,k|k} \\
&= P_{i,k|k} P_{i,k|k-1}^{-1} P_{i,k|k} + P_{i,k|k} S_{i,k} P_{i,k|k} = P_{i,k|k}.
\end{aligned} \tag{3.18}$$

We can further get

$$\mathbb{E} [\tilde{x}_{i,k+1|k} \tilde{x}_{i,k+1|k}^T] = F_k \mathbb{E} [\tilde{x}_{i,k|k} \tilde{x}_{i,k|k}^T] F_k^T + Q_k \leq F_k P_{i,k|k} F_k^T + Q_k = P_{i,k+1|k}.$$

Therefore, with the initial condition $\mathbb{E} [\tilde{x}_{k,0|-1} \tilde{x}_{k,0|-1}^T] \leq P_{k,0|-1}$, $\mathbb{E} [\tilde{x}_{i,k|k-1} \tilde{x}_{i,k|k-1}^T] \leq P_{i,k|k-1}$ holds for all k .

On the other hand, for $i \in \Upsilon$, the diffusion update replaces $P_{i,k|k-1}$ with the matrix $\Lambda_{i,k} \leq P_{i,k|k-1}$, which means $P_{i,k|k-1}$ must be bounded by the one without diffusion update. Hence, we ignore the diffusion update and write the algebraic Riccati iteration of $P_{i,k|k-1}$ by only considering the Kalman filter update as

$$\begin{aligned}
P_{i,k+1|k} &= F_k \left[P_{i,k|k-1}^{-1} + S_{i,k} \right]^{-1} F_k^T + Q_k \\
&= F_k \left[P_{i,k|k-1}^{-1} + \sum_{j=1}^N \frac{\rho_{i,j,k}(t_o)}{\max_{j \in \mathcal{V}} \rho_{i,j,k}(t_o)} H_{j,k}^T R_{j,k}^{-1} H_{j,k} \right]^{-1} F_k^T + Q_k \\
&= F_k \left[P_{i,k|k-1}^{-1} + H_k^T \hat{R}_k^{-1} H_k \right]^{-1} F_k^T + Q_k,
\end{aligned} \tag{3.19}$$

where

$$\hat{R}_k \triangleq \max_{j \in \mathcal{V}} \rho_{i,j,k}(t_o) \text{diag} \left\{ \frac{1}{\rho_{i,1,k}(t_o)} R_{1,i}, \dots, \frac{1}{\rho_{i,N,k}(t_o)} R_{1,i} \right\}$$

and $\hat{R}_k$ satisfies

$$R_k \leq \hat{R}_k \leq \frac{\max_{j \in \mathcal{V}} \rho_{i,j,k}(t_o)}{\min_{j \in \mathcal{V}} \rho_{i,j,k}(t_o)} R_k \leq \frac{1}{\min_{j \in \mathcal{V}} \rho_{i,j,k}(t_o)} R_k.$$

According to Lemma 3.2, we have $\min_{j \in \mathcal{V}} \rho_{i,j,k}(t_o) \geq (1/N)^{N-1}$, which implies $\hat{R}_k \leq N^{N-1} R_k$. Hence, (3.19) can be seen as a centralized algebraic Riccati iteration for the entire system with bounded noise covariance matrices $\hat{R}_k$ and Q_k . According to the conclusions in [116], there exists a positive definite matrix $\bar{P}_i$ such that $P_{i,k|k-1} \leq \bar{P}_i$ for all k if the bounded matrices F_k and H_k satisfy the uniform observability condition. Then, it follows that

$$\mathbb{E} [\tilde{x}_{i,k|k} \tilde{x}_{i,k|k}^T] \leq P_{i,k|k} = \left(P_{i,k|k-1}^{-1} + S_{i,k} \right)^{-1} \leq P_{i,k|k-1} \leq \bar{P}_i.$$

which implies that Theorem 3.2 holds. $\square$

Remark 3.3. Based on the definition of Υ , each agent $i \in \Upsilon$ can obtain the measurement information from all agents within each sampling interval. For a general case with time-varying topologies, it is hard to require an agent to determine its identity if not all agents are the members of Υ . In this case, the only method is to set a large enough t_o such that each agent can obtain the measurement information from all agents as illustrated in Remark 3.2, which implies that all agents are members of Υ . However, for time-invariant topologies, it is still possible to have a smaller t_o , under which not all agents have to be members of Υ . In this case, each agent i can determine its identity only by checking if $\min_{j \in \mathcal{V}} \rho_{i,j,k}(t_o) > 0$ in the first communication cycle because $\min_{j \in \mathcal{V}} \rho_{i,j,k}(t_o)$ is a fixed value for all k under a fixed topology. For example, in a connected network with a time-invariant topology, there always exists a tree that connects all agents. Then, the height (or an upper bound of it) of the tree can be set as t_o , which guaranties that there exists at least one agent (e.g., the root agent of the tree) $i \in \Upsilon$ with $\min_{j \in \mathcal{V}} \rho_{i,j,k}(t_o) > 0$ instead of all agents. Up to now, our selection of t_o is merely for the sake of estimation stability. It should be noted that a sufficiently large t_o can not only help more agents become uniformly

observable, but also lead to an estimate that is close to the optimal one given by the centralized method since $S_{i,k}(t_o) \rightarrow \bar{S}_k$ and $q_{i,k}(t_o) \rightarrow \bar{q}_k$ as $t_o \rightarrow \infty$ under Algorithm 3.2. Hence, users should make a trade-off between the communication energy consumption and the estimation performance for a suitable t_o . The influence of t_o on the estimation performance will be shown by simulation in Section 3.4.

3.4 Simulation

3.4.1 Simulation Environment

A time-invariant unstable system model is considered for the ease of simulation, though the proposed algorithm is not restricted to it. A stationary sensor network is estimating the dynamic energy intensity of two stationary sources, the positions of which are known. Each agent has a sensing range $R_s = 20\text{m}$. Two exponential functions are used to denote the energy intensity of two sources spreading over a surveillance region and the system model is given by:

$$F = \begin{bmatrix} 1 & 0.005 \\ 0 & 1 \end{bmatrix}, \quad G = I, \quad Q = 5I, \quad R_i = 20,$$

$$H_i = \begin{bmatrix} e^{-\lambda(s_i - \mu_1)^2} \mathbb{1}_{\{\|s_i - \mu_1\| \leq R_s\}} & e^{-\lambda(s_i - \mu_2)^2} \mathbb{1}_{\{\|s_i - \mu_2\| \leq R_s\}} \end{bmatrix},$$

where s_i is the position of agent i , μ_1 and μ_2 are the positions of the two sources, $\lambda = 0.02$ is the attenuation factor, and $\mathbb{1}_{\{\|s_i - \mu_1\| \leq R_s\}}$ is the indication function defined as

$$\mathbb{1}_{\{\|s_i - \mu_1\| \leq R_s\}} = \begin{cases} 1, & \text{if } \|s_i - \mu_1\| \leq R_s; \\ 0, & \text{otherwise.} \end{cases}$$

In the following simulations, different values will be set for the source positions and communication range R_c to get different local observability for each agent.

The same performance index adopted in [13], i.e., the mean-square deviation

(MSD), is used to evaluate the algorithm performance. The MSD for agent i at time k is defined as

$$\text{MSD}_{i,k} \triangleq \text{E} \left[\tilde{x}_{i,k|k}^T \tilde{x}_{i,k|k} \right].$$

Then, the MSD of the whole network is calculated as

$$\text{MSD}_k \triangleq \frac{1}{N} \sum_{i=1}^N \text{MSD}_{i,k}.$$

The results are averaged over 100 independent experiments. Besides, to show that $\text{E} \left[\tilde{x}_{i,k|k} \tilde{x}_{i,k|k}^T \right]$ is bounded by $P_{i,k|k}$, the following averaged trace is defined:

$$\text{Tr}_k \triangleq \frac{1}{N} \sum_{i=1}^N \text{tr} (P_{i,k|k}).$$

We first implement simulations in two scenarios to compare the performance of six different algorithms: DKF algorithm with adaptive weights, no diffusion algorithm (i.e. each agent only implements the incremental update), CI algorithm, CI-DKF algorithm for partial local observability (Algorithm 3.1) with weight choice rule (3.5) and (3.6), and the centralized algorithm (i.e., each agent can obtain the original measurement information of all agents) which provides the optimal estimate. In Scenario I (Fig. 3.3), $N = 25$ agents are uniformly deployed over a $50 \times 50\text{m}^2$ square region and the positions of the two sources are fixed at $\mu_1 = [20 \ 30]^T$ and $\mu_2 = [30 \ 20]^T$ respectively. The communication range is set as $R_c = 20\text{m}$ such that each agent can obtain enough measurement information to become uniformly observable. In Scenario II (Fig. 3.4), the communication range is changed to $R_c = 15\text{m}$ such that not all agents can collect enough measurement information from their neighbors and thus become unobservable. In addition, ten more agents are added randomly within $[-10, 0] \times [-10, 0]$ which are not uniformly observable since they are too far away from the sources and cannot get enough measurement information from their neighbors as well.

Then, in Scenario III, the performance of the CI-DKF algorithm is tested for

no local observability (Algorithm 3.3) with weight choice rule (3.5). In this case, each agent needs to collect the measurement information using the Algorithm 3.2. After obtaining enough measurement information to become uniformly observable for some agents, the condition will be the same with that in Scenario I or II. Hence, the above algorithms are not compared in Scenario III. Instead, we examine the influence of t_o on the estimation performance of Algorithm 3.3. In this simulation, again $N = 25$ agents are deployed uniformly over a $50 \times 50\text{m}^2$ square region, but place the sources at $\mu_1 = [0 \ 50]^T$ and $\mu_2 = [50 \ 0]^T$ respectively, and set $R_c = 15\text{m}$ such that no agent is uniformly observable as shown in Fig. 3.5. In this case, the smallest time length that makes only one agent (i.e., the agent at the center of the region) be the member of Υ is $t_o = 2$. To show the influence of t_o on the algorithm performance, several different values are selected for testing, i.e., $t_o = 2$, $t_o = 15$ and $t_o = 30$. The results are all compared with the centralized algorithm.

3.4.2 Simulation Results

In Scenario I, all the six algorithms can provide stable estimation as shown in Fig. 3.6. However, the results of our algorithm are closest to the optimal estimate given by the centralized algorithm which illustrates that the diffusion and incorporation of error covariance information can both improve the estimation performance. In Scenario II (Fig. 3.7), the results of the DKF algorithm with adaptive weights and the no diffusion algorithm both diverge. Although the CI algorithm still provides a stable estimate, its result is much worse than that of the centralized algorithm while the results of our algorithm are close to the optimal estimate. The comparison between the algorithms with diffusion and the algorithm without diffusion in the two scenarios illustrates that diffusion can improve the network average performance. Another point to note is that the algorithm 3.1 with rule (3.5) and (3.6) has nearly the same performance. This suggests us to use rule (3.6) in real applications due to its simplified computation.

In Scenario III, Fig. 3.8 shows that Algorithm 3.3 can produce a stable estimate for each agent with only 2 consensus steps, and a sufficiently large t_o can lead to a near optimal result. It is also shown that getting the optimal estimate requires much more communications than getting a stable estimate. In all, our algorithm provides users with an adjustable parameter, through which they can obtain a better trade-off between their needs of performance and the communication cost.

Figure 3.3: Simulation setup of Scenario I: full local uniform observability.

Figure 3.4: Simulation setup of Scenario II: partial local uniform observability.

Figure 3.5: Simulation setup of Scenario III: no local uniform observability.

3.5 Conclusions

In this chapter, a CI-based diffusion Kalman filtering (CI-DKF) algorithm is proposed by incorporating the covariance information. The CI-DKF algorithm can be applied in the case of lacking local observability. A consensus-based information diffusion scheme is embedded when no single agent can observe the state. Simulation shows that the CI-DKF algorithm has better performance than that by the original DKF and those by local Kalman filters.

(a) DKF algorithm with adaptive weights.

(b) No diffusion algorithm.

(c) CI algorithm.

(d) Algorithm 3.1 with rule (5).

(e) Algorithm 3.1 with rule (6).

(f) Centralized algorithm.

Figure 3.6: Results of Scenario I: full local uniform observability.

(a) DKF algorithm with adaptive weights.

(b) No diffusion algorithm.

(c) CI algorithm.

(d) Algorithm 3.1 with rule (5).

(e) Algorithm 3.1 with rule (6).

(f) Centralized algorithm.

Figure 3.7: Results of Scenario II: partial local uniform observability.

(a) Algorithm 3.3 with $t_o = 2$. (b) Algorithm 3.3 with $t_o = 15$. (c) Algorithm 3.3 with $t_o = 30$.

(d) Centralized algorithm.

Figure 3.8: Results of Scenario III: no local uniform observability.

Chapter 4

Energy-Based Multiple Target Localization and Pursuit in Mobile Sensor Networks

In this chapter, an energy-based distributed scheme is designed for the localization and pursuit of multiple mobile targets using mobile robotic agents. Its application scenario (as shown in Fig. 4.1) is that when multiple targets, which produce a measurable energy intensity field, intrude into or move inside a monitored region, agents can autonomously converge to the targets based on sampled information. Agents which are initially far away from the targets can only sense the energy intensity field such as an acoustic or a temperature field. When coming close to the targets, they may use other short-length sensors like sonar or optical camera to better observe the targets. Our objective is to make agents move closely to the targets based on the discrete measurements of the energy intensity field. The potential application areas of our work include robots surveillance, intruder detection and interception, emergency rescue, etc.

In Section 4.1, the sensing model and some basic assumptions are introduced. A continuous-time formulation of the distributed control strategy is presented in

Figure 4.1: Energy-based multiple target localization and pursuit.

Section 4.2. A discrete-time control strategy and an approximation of spatial gradient are given in Section 4.3. Section 4.4 presents a target pursuit scheme and its convergence analysis. Simulation results are provided in Section 4.5. Section 4.6 presents the conclusions.

4.1 Basic Assumptions and Definitions

4.1.1 Sensing Model

Consider the spatial and temporal energy intensity field $\varphi : \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}_{>0}$ over a two dimensional space, produced by mobile targets. Generally, for a single target, the energy intensity in the field attenuates as the distance to the target increases. Hence, by assuming that the target energy intensity attenuates isotropically along each direction, the field should be a function of the distance to the target at a fixed

time t , which can be modeled as

$$\varphi(t, q) = A\phi(\|q - p(t)\|), \quad (4.1)$$

where $q \in \mathbb{R}^2$ is a point in the two dimensional space, $p(t)$ is the position of the target and A is the energy intensity of the target. $\phi(d) : [0, +\infty) \rightarrow \mathbb{R}_{>0}$ is a strictly monotonically decreasing function subject to $\phi(0) = 1$ and $\phi(d) \rightarrow 0$ as $d \rightarrow +\infty$, which represents the attenuating part of φ . Then, the composite field of M targets is obtained by summing up the individual field produced by each target:

$$\varphi(t, q) = \sum_{m=1}^M \varphi_m(t, q) = \sum_{m=1}^M A_m \phi(\|q - p_m(t)\|), \quad (4.2)$$

where the subscript $m = 1, 2, \dots, M$ refers to the m -th target. In this thesis, the field model is not limited to a specific deterministic attenuation function but only require some critical bounds such that our method can be applied to the most general practical case.

The position of the i -th agent is: $\mu_i(t) = [x_i(t), y_i(t)]^T \in \mathbb{R}^2$ ($i = 1, 2, \dots, N$), where $x_i(t)$ and $y_i(t)$ represent the geographical coordinates of agent i . The measurement of the energy intensity field of agent i at time t is $\bar{\varphi}_i(t) = \varphi(t, \mu_i(t))$ where the measurement noise is not considered. In real systems, the influence of noise can be suppressed by installing multiple sensors on each agent and let them sample simultaneously so as to get some kind of average of the measurements and thus filter the noise. The graph of the network and other elementary notations are defined in Appendix A.

4.1.2 Assumptions

Each agent is assumed to have access to its position through, for example, the GPS. Though the number of targets is not assumed to be known exactly, users should deploy a sufficient number of agents to pursue the targets. A_m ($m = 1, 2, \dots, M$)

are constants which do not change with time. $\phi(\|\bar{q}\|)$, where $\|\cdot\|$ stands for 2 norm, is assumed to be at least second order differentiable with respect to $\bar{q} \in \mathbb{R}^2$, which implies $\frac{\partial \phi}{\partial \bar{q}}(0) = 0$. In addition, $\left\| \frac{\partial \phi}{\partial \bar{q}}(\|\bar{q}\|) \right\|$ and $\left\| \frac{\partial^2 \phi}{\partial \bar{q} \partial \bar{q}^T}(\|\bar{q}\|) \right\|$ are bounded.

The mobile agents are assumed to move and sample synchronously. The time duration for communication and computation can be omitted compared with the sampling interval. The N agents can generate a partial Delaunay triangulation within the constraint of communication range R_c and calculate their Voronoi neighbors $\mathcal{N}_{V_{i,k}}$ [118].

4.2 Continuous-Time Formulation

A first-order dynamic model of agent i is assumed and given by

$$\dot{\mu}_i(t) = u_i(t), \quad u_i(t) \in \mathbb{R}^2.$$

For simplicity of notation, time t will be omitted when no confusion is caused. By the potential field method with a user defined potential energy function U_i for agent i , the control law u_i should be chosen such that $\dot{U}_i < 0$. In a multi-agent system, neighboring agents should maintain certain formations for collision avoidance or communication connectivity. Meanwhile, in a target pursuit problem, an agent must be attracted by a force from each target in the potential field in order to catch them. Hence, U_i must include a potential energy U_{A_i} depending on the relative distance between agent i and each one of its neighbors, and a potential energy U_{B_i} depending on the relative distance between agent i and each target, i.e.,

$$U_i = U_{A_i} + U_{B_i}.$$

Though the potential field method for agent coordination is not new and there are many possible choices for U_{A_i} , the design of U_{B_i} has rarely been discussed before for an unknown set of targets merely based on the intensity measurements. In this

chapter, we respectively define U_{A_i} which is an extension of the energy function proposed in [66], and U_{B_i} using an energy intensity bound as

$$U_{A_i} \triangleq \sum_{j \in \mathcal{N}_{V_i, k}} \left(\frac{k_1}{\|\mu_i - \mu_j\|} + \frac{k_2}{R_c - \|\mu_i - \mu_j\|} \right),$$

$$U_{B_i} \triangleq k_r (MA_{\max} - \bar{\varphi}_i),$$

where k_1 , k_2 and k_r are positive gain parameters and $A_{\max}$ is a given energy intensity bound which satisfies $A_{\max} \geq A_m, \forall m = 1, 2, \dots, M$. The above defined cost function U_{A_i} differs from that defined in [66] because it includes the distance limitation for neighboring agents as a requirement of the network connectivity. Then, we get

$$U_i = U_{A_i} + U_{B_i} = \sum_{j \in \mathcal{N}_{V_i, k}} \left(\frac{k_1}{\|\mu_i - \mu_j\|} + \frac{k_2}{R_c - \|\mu_i - \mu_j\|} \right) + k_r (MA_{\max} - \bar{\varphi}_i). \quad (4.3)$$

It is easy to see that when agent i is running out of the communication range R_c or very close to any one of its neighbors, U_{A_i} will increase greatly, which leads to the increasing of U_i . Meanwhile, the further the agent i from the targets, the larger the U_{B_i} . Since the true target energy intensities are unknown, $A_{\max}$ is used to make U_i nonnegative. Its influence on the localization or tracking result will be discussed later.

Now, our task is to find u_i that makes $\dot{U}_i$ negative. From (4.3), we get:

$$\dot{U}_i = \dot{U}_{A_i} + \dot{U}_{B_i} = \frac{\partial U_{A_i}}{\partial \mu_i} \dot{\mu}_i + \frac{\partial U_{B_i}}{\partial \mu_i} \dot{\mu}_i = -(F_{1_i} + F_{2_i})^T \dot{\mu}_i - k_r \frac{\partial \varphi}{\partial t}(t, \mu_i), \quad (4.4)$$

where

$$F_{1_i} = \sum_{j \in \mathcal{N}_{V_i, k}} \left(\frac{k_1}{\|\mu_i - \mu_j\|^2} - \frac{k_2}{(R_c - \|\mu_i - \mu_j\|)^2} \right) \frac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|},$$

$$F_{2_i} = k_r \frac{\partial \varphi}{\partial q}(t, \mu_i).$$

F_{1_i} and F_{2_i} stand for the formation control force among neighbors and the gradient climbing force towards the local maximum points of the potential field respectively. By choosing the control input as

$$u_i = K (F_{1_i} + F_{2_i}),$$

where $K > 0$ is a scalar gain parameter, we can get

$$\dot{U}_i = -K (F_{1_i} + F_{2_i})^T (F_{1_i} + F_{2_i}) - k_r \frac{\partial \varphi}{\partial t} (t, \mu_i). \quad (4.5)$$

If the targets are stationary, i.e., $\frac{\partial \varphi}{\partial t} \equiv 0$, then we have $\dot{U}_i \leq 0$ and $\dot{U}_i = 0$ only when $F_{1_i} + F_{2_i} = 0$. In this case, it has been proven by [66] that with possible topological change, the N agents will converge to stationary deployment. If the targets are mobile, $\dot{U}_i$ also depends on the unknown term $\frac{\partial \varphi}{\partial t} (t, \mu_i)$. Thus, if $\frac{\partial \varphi}{\partial t} (t, \mu_i) < 0$, the system will never converge to a stationary deployment and the agents only converge into neighborhoods of their local maximum points. The convergence property of such control law will be discussed in Section 4.4.

In a network of N agents, each local maximum point only needs one agent to pursue and it is not necessary to add a gradient climbing force in control inputs of all agents. Hence, the N agents are divided into two groups, leaders and followers. Target pursuit is only taken as a task by the leaders, while the followers only need to follow the leaders by maintaining certain formation. Each agent i determines itself as a leader or follower according to the following rule:

$$Type_{i,k}(t) = \begin{cases} Leader, & \text{if } \bar{\varphi}_i(t) > \bar{\varphi}_j(t) \ \forall j \in \mathcal{N}_{V_{i,k}}; \\ Leader, & \text{if } \bar{\varphi}_i(t) = \bar{\varphi}_j(t) \ \& \ i < j \ \forall j \in \mathcal{N}_{V_{i,k}}; \\ Follower, & \text{otherwise.} \end{cases} \quad (4.6)$$

The type of an agent may change depending upon the measurements of its neighbors.

Then, two different control laws are chosen for the leaders and followers:

$$u_i = \begin{cases} K(F_{1_i} + F_{2_i}), & \text{if } Type_{i,k} = Leader; \\ KF_{1_i}, & \text{otherwise.} \end{cases}$$

The gain parameter k_r should be large enough to make the gradient climbing force dominant (i.e., $u_i \approx KF_{2_i}$) when leaders are not too far from or close to any one of their neighbors. Besides, the optimal network formation size d_0 (i.e., the optimal distance between neighboring agents) can also be set by choosing appropriate k_1 and k_2 . When two neighboring agents are separated by the optimal distance d_0 , their interactive potential force should be zero. Hence, by letting $\frac{k_1}{d_0^2} - \frac{k_2}{(R_c - d_0)^2} = 0$ we have

$$\frac{k_1}{k_2} = \frac{d_0^2}{(R_c - d_0)^2}. \quad (4.7)$$

Therefore, with a desired d_0 , proper parameters k_1 and k_2 can be determined by (4.7) to set the control law.

Remark 4.1. *In the real world, the measurements are taken at discrete time instants based on which the unknown spatial gradient should be estimated in order to calculate the control law. Thus, we have to consider a discrete-time control law and analyze the convergence performance based on discrete time measurements.*

4.3 Discrete-Time Formulation

In the discrete-time case, the agent motion model is given by:

$$\mu_{i,k+1} = \mu_{i,k} + u_{i,k},$$

where $k \in \mathbb{R}$ is the shorthand for kT and T is the sampling interval. The velocity of i -th agent during the k -th sampling interval can thus be expressed as

$$v_{r_{i,k}} = \frac{1}{T} (\mu_{i,k} - \mu_{i,k-1}) = \frac{1}{T} u_{i,k-1}.$$

We use the control law obtained in the continuous-time case by changing the continuous time index into the discrete time step, i.e.,

$$u_{i,k} = \begin{cases} K (F_{1_{i,k}} + F_{2_{i,k}}), & \text{if } Type_{i,k} = Leader; \\ KF_{1_{i,k}}, & \text{otherwise.} \end{cases} \quad (4.8)$$

Since the control input of leaders includes the spatial gradient $\frac{\partial \varphi}{\partial q}(k, q)$ of the unknown field, we need to estimate it by discrete measurements. By the first order Taylor expansion, we have

$$\bar{\varphi}_{i,k-1} = \bar{\varphi}_{i,k} - \frac{\partial \varphi^T}{\partial q}(k, \mu_{i,k}) [\mu_{i,k} - \mu_{i,k-1}] - \frac{\partial \varphi}{\partial t}(k, \mu_{i,k}) T + \rho_1(t_1, q_1), \quad (4.9)$$

where $t_1 \in [k-1, k]$, $q_1 \in [x_{i,k-1}, x_{i,k}] \times [y_{i,k-1}, y_{i,k}]$ and $\rho_1(t_1, q_1)$ is the combined second and higher order approximation error. Similarly, we have

$$\bar{\varphi}_i(k-2) = \bar{\varphi}_{i,k} - \frac{\partial \varphi^T}{\partial q}(k, \mu_{i,k}) [\mu_{i,k} - \mu_{i,k-2}] - 2 \frac{\partial \varphi}{\partial t}(k, \mu_{i,k}) T + \rho_2(t_2, q_2), \quad (4.10)$$

where $t_2 \in [k-2, k]$, $q_2 \in [x_{i,k-2}, x_{i,k}] \times [y_{i,k-2}, y_{i,k}]$. If the field varies slowly and the sampling rate is high enough, i.e., $\frac{\partial \varphi}{\partial t} \rightarrow 0$ and $T \rightarrow 0$, then by (4.9) and (4.10), we get

$$\Delta \Phi_{i,k} = \Delta P_{i,k} \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) + E_{i,k} \approx \Delta P_{i,k} \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}),$$

where

$$\begin{aligned}\Delta\Phi_{i,k} &= \begin{bmatrix} \bar{\varphi}_{i,k} - \bar{\varphi}_{i,k-1} \\ \bar{\varphi}_{i,k} - \bar{\varphi}_i(k-2) \end{bmatrix}, \\ \Delta P_{i,k} &= \begin{bmatrix} [\mu_{i,k} - \mu_{i,k-1}]^T \\ [\mu_{i,k} - \mu_{i,k-2}]^T \end{bmatrix}, \\ E_{i,k} &= \begin{bmatrix} \frac{\partial \varphi}{\partial t}(k, \mu_{i,k})T - \rho_1(t_1, q_1) \\ 2\frac{\partial \varphi}{\partial t}(k, \mu_{i,k})T - \rho_2(t_2, q_2) \end{bmatrix}.\end{aligned}\tag{4.11}$$

Hence, if $\Delta P_{i,k}$ is invertible, an estimate of the spatial gradient is

$$\frac{\partial \hat{\varphi}}{\partial q}(k, \mu_{i,k}) = \Delta P_{i,k}^{-1} \Delta \Phi_{i,k}.\tag{4.12}$$

The error of such approximation is given by

$$e_{i,k} = \frac{\partial \hat{\varphi}}{\partial q}(k, \mu_{i,k}) - \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) = \Delta P_{i,k}^{-1} E_{i,k}.\tag{4.13}$$

To make the approximation error bounded, the matrix $\Delta P_{i,k}$ should first be made invertible. From (4.11), we get

$$\Delta P_{i,k} = [u_{i,k-1}, u_{i,k-1} + u_{i,k-2}]^T.$$

Since $\Delta P_{i,k}$ is defined by $u_{i,k-2}$ and $u_{i,k-1}$, for the ease of implementation, we set

$$\|u_{i,k-2}\| = \|u_{i,k-1}\| = T v_r\tag{4.14}$$

such that $\Delta P_{i,k}$ only depends on the directions of $u_{i,k-2}$ and $u_{i,k-1}$, where $v_r \in \mathbb{R}_{>0}$ is a user defined moving speed of agents. By denoting the angle from $u_{i,k-2}$ to $u_{i,k-1}$ as α ($\alpha \in [-\pi, \pi)$) and with simple calculations, we can work out the determinant of $\Delta P_{i,k}$ as:

$$|\Delta P_{i,k}| = T^2 v_r^2 \sin \alpha.$$

Hence, $u_{i,k-2}$ and $u_{i,k-1}$ should be non-collinear to make $\Delta P_{i,k}$ invertible.

Second, since $\|e_{i,k}\| \leq \|\Delta P_{i,k}^{-1}\| \|E_{i,k}\|$ according to (4.13), we need to minimize $\|\Delta P_{i,k}^{-1}\|$ in order to minimize $\|e_{i,k}\|$. Hence, we should find the optimal α that minimizes $\|\Delta P_{i,k}^{-1}\|$, i.e., solve the following optimization problem:

$$\alpha_{opt} = \arg \min_{\alpha \in [-\pi, \pi)} \|\Delta P_{i,k}^{-1}\|, \quad (4.15)$$

where α_{opt} is the optimal solution. It can be worked out that

$$\|\Delta P_{i,k}^{-1}\| = \frac{1}{v_r T} \sqrt{\frac{3 + 2 \cos \alpha + \sqrt{8 \cos^2 \alpha + 12 \cos \alpha + 5}}{2(1 - \cos^2 \alpha)}}.$$

Then, by solving (4.15), we can get

$$\alpha_{opt} = \pm \frac{2\pi}{3},$$

where '+' denotes the clockwise direction and '-' the counterclockwise direction.

In real implementations, we can fix the turning direction of the agent to be either one of them for spatial gradient approximation, and this leads to the minimum of $\|\Delta P_{i,k}^{-1}\|$ as

$$\|\Delta P_{i,k}^{-1}\| = \frac{\sqrt{2}}{v_r T}. \quad (4.16)$$

Next, we will derive a bound of the approximation error. First, consider the case of single target and give the following result.

Theorem 4.1. *For the energy intensity field $\varphi(t, q)$ defined by (4.1), given the bounds $\bar{p} \geq \|\dot{p}\|$, $A_{\max} \geq A$, $G_1 \geq \left\| \frac{d\phi}{d\bar{q}}(\|\bar{q}\|) \right\|$ and $G_2 \geq \left\| \frac{d^2\phi}{d\bar{q}d\bar{q}^T}(\|\bar{q}\|) \right\| \forall \bar{q} \in \mathbb{R}^2$, $\|e_{i,k}\|$ is bounded by*

$$\|e_{i,k}\| \leq \bar{e}_1,$$

where $\bar{e}_1 = \frac{\sqrt{2}A_{\max}}{\eta_1} [\sqrt{5}G_1 + \xi_1 \bar{p}G_2T]$, $\xi_1 = \frac{1}{2} \sqrt{(1 + \eta_1)^4 + (2 + \sqrt{2}\eta_1)^4}$, $\eta_1 = \frac{v_r}{\bar{p}}$.

Proof. From (4.1), we can get

$$\frac{\partial^2 \varphi}{\partial t^2} = \dot{p}^T \frac{\partial^2 \varphi}{\partial p \partial p^T} \dot{p}, \quad \frac{\partial^2 \varphi}{\partial t \partial q} = - \frac{\partial^2 \varphi}{\partial p \partial p^T} \dot{p}, \quad \frac{\partial^2 \varphi}{\partial q \partial q^T} = \frac{\partial^2 \varphi}{\partial p \partial p^T}.$$

Hence, at arbitrary time k , we have

$$\begin{aligned} \rho_1(t_1, q_1) &= \frac{T^2}{2} \frac{\partial^2 \varphi}{\partial t^2}(t_1, q_1) + T \frac{\partial^2 \varphi^T}{\partial t \partial q}(t_1, q_1) [\mu_{i,k} - \mu_{i,k-1}] \\ &\quad + \frac{1}{2} [\mu_{i,k} - \mu_{i,k-1}]^T \frac{\partial^2 \varphi}{\partial q \partial q^T}(t_1, q_1) [\mu_{i,k} - \mu_{i,k-1}] \\ &= \frac{T^2}{2} \left(\dot{p}^T \frac{\partial^2 \varphi}{\partial q \partial q^T}(t_1, q_1) \dot{p} + 2 \dot{p}^T \frac{\partial^2 \varphi^T}{\partial q \partial q^T}(t_1, q_1) v_{r_{i,k}} \right. \\ &\quad \left. + v_{r_{i,k}}^T \frac{\partial^2 \varphi}{\partial q \partial q^T}(t_1, q_1) v_{r_{i,k}} \right). \end{aligned}$$

Thus,

$$\begin{aligned} \rho_1(t_1, q_1) &\leq \frac{T^2 \|\dot{p}\|^2}{2} \left(1 + 2 \frac{\|v_{r_{i,k}}\|}{\|\dot{p}\|} + \frac{\|v_{r_{i,k}}\|^2}{\|\dot{p}\|^2} \right) \left\| \frac{\partial^2 \varphi}{\partial q \partial q^T}(t_1, q_1) \right\| \\ &\leq \frac{(1 + \eta)^2 \bar{p}^2 A_{\max}}{2} G_2 T^2. \end{aligned}$$

With (4.14), in the same way, we can get

$$\|\rho_2(t_2, q_2)\| \leq \frac{(2 + \sqrt{2}\eta)^2 \bar{p}^2 A_{\max}}{2} G_2 T^2.$$

Therefore,

$$\begin{aligned} \|E_{i,k}\| &\leq \left\| \begin{bmatrix} \frac{\partial \varphi}{\partial t}(k, \mu_{i,k}) T \\ 2 \frac{\partial \varphi}{\partial t}(k, \mu_{i,k}) T \end{bmatrix} \right\| + \left\| \begin{bmatrix} \rho_1(t_1, q_1) \\ \rho_2(t_2, q_2) \end{bmatrix} \right\| \\ &\leq \sqrt{5} A_{\max} \bar{p} G_1 T + \xi_1 \bar{p}^2 A_{\max} G_2 T^2 \triangleq \bar{E}_1, \end{aligned}$$

where $E_{i,k}$ is defined in (4.11). From (4.16), we get

$$\|e_{i,k}\| \leq \|\Delta P_{i,k}^{-1}\| \|E_{i,k}\| \leq \|\Delta P_{i,k}^{-1}\| \bar{E}_1 = \frac{\sqrt{2}A_{\max}}{\eta_1} \left[\sqrt{5}G_1 + \xi_1 \bar{p} G_2 T \right] = \bar{e}_1.$$

□

For the case of M targets, a similar result can be obtained as follows.

Theorem 4.2. *For the energy intensity field $\varphi(t, q)$ defined by (4.2), given the bounds $\bar{p} \geq \max_{m=1,2,\dots,M} \|\dot{p}_m\|$, $A_{\max} \geq \max_{m=1,2,\dots,M} A_m$, $G_1 \geq \left\| \frac{d\phi}{d\bar{q}}(\|\bar{q}\|) \right\|$ and $G_2 \geq \left\| \frac{d^2\phi}{d\bar{q}d\bar{q}^T}(\|\bar{q}\|) \right\| \forall \bar{q} \in \mathbb{R}^2$, $\|e_{i,k}\|$ is bounded by*

$$\|e_{i,k}\| \leq \bar{e}_2,$$

$$\text{where } \bar{e}_2 = \frac{\sqrt{2}MA_{\max}}{\eta_2} \left[\sqrt{5}G_1 + \xi_2 \bar{p} G_2 T \right], \quad \xi_2 = \frac{1}{2} \sqrt{(1 + \eta_2)^4 + (2 + \sqrt{2}\eta_2)^4}, \quad \eta_2 = \frac{v_r}{\bar{p}}.$$

Proof. From (4.2), we have

$$\begin{aligned} \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) &= \sum_{m=1}^M \frac{\partial \varphi_m}{\partial q}(k, \mu_{i,k}), \\ \frac{\partial \varphi}{\partial t}(k, \mu_{i,k}) &= - \sum_{m=1}^M \frac{\partial \varphi_m^T}{\partial q}(k, \mu_{i,k}) \dot{p}_m, \\ \frac{\partial^2 \varphi}{\partial q \partial q^T}(k, \mu_{i,k}) &= \sum_{m=1}^M \frac{\partial^2 \varphi_m}{\partial q \partial q^T}(k, \mu_{i,k}), \\ \frac{\partial^2 \varphi}{\partial t^2}(k, \mu_{i,k}) &= \sum_{m=1}^M \dot{p}_m^T \frac{\partial^2 \varphi_m}{\partial q \partial q^T}(k, \mu_{i,k}) \dot{p}_m, \\ \frac{\partial^2 \varphi}{\partial t \partial q}(k, \mu_{i,k}) &= - \sum_{m=1}^M \frac{\partial^2 \varphi_m}{\partial q \partial q^T}(k, \mu_{i,k}) \dot{p}_m \end{aligned}$$

which imply that

$$\begin{aligned} \left\| \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) \right\| &\leq MA_{\max} G_1, & \left\| \frac{\partial \varphi}{\partial t}(k, \mu_{i,k}) \right\| &\leq MA_{\max} \bar{p} G_1, \\ \left\| \frac{\partial^2 \varphi}{\partial q \partial q^T}(k, \mu_{i,k}) \right\| &\leq MA_{\max} G_2, & \left\| \frac{\partial^2 \varphi}{\partial t^2}(k, \mu_{i,k}) \right\| &\leq MA_{\max} \bar{p}^2 G_2, \\ \left\| \frac{\partial^2 \varphi}{\partial t \partial q}(k, \mu_{i,k}) \right\| &\leq MA_{\max} \bar{p} G_2. \end{aligned}$$

Along the same line of the proof as for the single target case, we can get

$$\|e_{i,k}\| \leq \frac{\sqrt{2}MA_{\max}}{\eta_2} \left[\sqrt{5}G_1 + \xi_2 \bar{p} G_2 T \right] = \bar{e}_2.$$

□

Remark 4.2. *Theorem 4.1 and Theorem 4.2 give a relation between the bound of approximation error and the parameter setting. Since the convergence of leaders to the local maximum points is greatly influenced by the approximation error, the relation between the convergence error and the approximation error as well as the parameter setting will be further explored in the following section.*

4.4 Target Pursuit Scheme

4.4.1 Pursuit Scheme

If the measurement value of a leader i at time $k + 1$ is larger than that sampled at time k , this moving direction is recognized as a valid direction for agent i , otherwise it is an invalid direction. Since each approximation requires a turning movement and matrix calculations that consume additional energy, we want to avoid such approximation if it is not necessarily needed. Hence, we can let the leader keep moving along the latest valid direction without making new approximation until its measurement value starts to decrease.

On the other hand, if the leader moves along the direction of $\frac{\partial \hat{\varphi}}{\partial q}$, it may be trapped around the point where $\frac{\partial \hat{\varphi}}{\partial q} = 0$ rather than $\frac{\partial \varphi}{\partial q} = 0$ by implementing the designed control law. Thus, before leader i judges that it has converged into a neighborhood of a local maximum point, it is required to move at a constant speed v_r instead of the time-variant input defined in (4.8), i.e.,

$$u_{i,k} = \begin{cases} T v_r \frac{F_{1i,k} + F_{2i,k}}{\|F_{1i,k} + F_{2i,k}\|}, & \text{if } \|F_{1i} + F_{2i}\| > 0; \\ 0, & \text{otherwise.} \end{cases} \quad (4.17)$$

Users can define different criteria for leaders to judge their convergence and the behavior of leaders will not be described in detail when such convergence is confirmed. For example, a leader confirms the convergence if the norm of its spatial gradient approximation by (4.12) is smaller than a given threshold. Based on the above idea, we design the Pursuit Scheme (Algorithm 4.1) for the leaders.

Algorithm 4.1: Pursuit Scheme

After getting the approximation $\frac{\partial \hat{\varphi}}{\partial q}(k, \mu_{i,k})$ by (4.12) at time k for leader i :

- 1 Move by the control law (4.17);
 - 2 Get new sample $\bar{\varphi}_{i,k+1}$;
 - 3 If $\bar{\varphi}_{i,k+1} > \bar{\varphi}_{i,k}$
 - 4 Update $\frac{\partial \hat{\varphi}}{\partial q}(k+1, \mu_{i,k+1})$ by

$$\frac{\partial \hat{\varphi}}{\partial q}(k+1, \mu_{i,k+1}) = \frac{\bar{\varphi}_{i,k+1} - \bar{\varphi}_{i,k}}{\|\mu_{i,k+1} - \mu_{i,k}\|^2} [\mu_{i,k+1} - \mu_{i,k}];$$
 - 5 $k \leftarrow k+1$ and repeat 1-3 steps;
 - else
 - 6 Turn $\frac{2\pi}{3}$ and let $\|u_{i,k+1}\| = T v_r$;
 - 7 Get new sample $\bar{\varphi}_{i,k+2}$;
 - 8 Update $\frac{\partial \hat{\varphi}}{\partial q}(k+2, \mu_i(k+2))$ by (4.12);
 - 9 $k \leftarrow k+2$ and repeat 1-3 steps;
 - end
-

Remark 4.3. *At the initial stage, leader i can choose an arbitrary direction for*

$u_i(0)$. In the case that users may not deploy too many agents but let most or all agents work as leaders, a distance parameter h_d called type selection threshold can be set to meet this objective. The type selection rule in (4.6) is only executed if the distance between two neighboring agents is less than h_d so as to avoid collision in case that they are pursuing the same target, otherwise all agents implement the leader's Pursuit Scheme.

Remark 4.4. In fact, the information of relative positions of neighbours is enough for the implementation of Algorithm 4.1. However, if only the relative positions of neighbours are known, each agent has to calculate the coordinates of its neighbours in its local coordinate system at each time. That is, in order to implement Algorithm 4.1 effectively, each agent should build a local coordinate system at the initial stage. Then, at each time, each agent calculates the coordinates of its neighbours including itself in such a local coordinate system based on the relative positions. The only difference by using the relative positions is that the computation load of each agent becomes higher due to the extra calculations for the absolute positions in its local coordinate system.

4.4.2 Convergence Analysis

As discussed above, the formation control force for leaders should be negligible during the target pursuit and plays a major role only when neighboring nodes are running out of communication constraint or going to collide. Therefore, in the following parts, our focus will be narrowed down to the general case that the formation control force is omitted for leaders, i.e., $F_{1_i}(t) = 0$ for leader i . Besides, we assume that the sampling frequency is very high and the targets move slowly, i.e., $T(v_r + \bar{p})$ is very small, so that the turning movement for the approximation can be ignored, which means the movement in the last two steps before a new approximation is ignored for leaders. This ignorance is acceptable because $\frac{\partial \hat{\varphi}}{\partial q}$ dominates the long-term moving direction of agent i , and the movement in a small time interval takes obvious effect on the convergence only when agent i has already come very close to a local

maximum point.

According to (4.4), $\dot{U}_i(t) < 0$ is equivalent to

$$\frac{\partial \varphi^T}{\partial q}(t, \mu_i) v_{r_{i,k}} + \frac{\partial \varphi}{\partial t}(t, \mu_i) > 0. \quad (4.18)$$

Since the convergence of agent i is implied by the decreasing of U_i , it keeps converging through a sequence of approximations until at a time instant k that $v_{r_{i,k}}$, due to the approximation of $\frac{\partial \hat{\varphi}}{\partial q}(k, \mu_{i,k})$, can not satisfy (4.18). Then, we can get the following convergence result for the single target case.

Theorem 4.3. *For the energy intensity field $\varphi(t, q)$ defined by (4.1), leader i finally converges into the region bounded by*

$$\left\| \frac{\partial \varphi}{\partial q}(t, q) \right\| \leq \bar{c}_1, \quad (4.19)$$

where $\bar{c}_1 = \frac{\eta_1}{\sqrt{\eta_1^2 - 1}} \bar{e}_1$.

Proof. For the single target case at time k , (4.18) evolves as

$$\frac{\partial \varphi^T}{\partial q}(k, \mu_{i,k}) (v_{r_{i,k}} - \dot{p}) > 0. \quad (4.20)$$

A sufficient condition for (4.20) is

$$\frac{\partial \varphi^T}{\partial q}(k, \mu_{i,k}) v_{r_{i,k}} > \bar{p} \left\| \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) \right\|. \quad (4.21)$$

By denoting the angle between $\frac{\partial \hat{\varphi}}{\partial q}(k, \mu_{i,k})$ and $\frac{\partial \varphi}{\partial q}(k, \mu_{i,k})$ as $\theta(k)$, we get a sufficient condition for (4.21) as $\cos \theta(k) > \frac{1}{\eta_1}$ for $\left\| \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) \right\| \neq 0$, which is a necessary condition for

$$\sqrt{1 - \frac{\|e_{i,k}\|^2}{\left\| \frac{\partial \varphi}{\partial q}(k, \mu_i(t)) \right\|^2}} > \frac{1}{\eta_1},$$

i.e.,

$$\frac{\eta_1}{\sqrt{\eta_1^2 - 1}} \|e_{i,k}\| < \left\| \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) \right\|. \quad (4.22)$$

A sufficient condition for (4.22) is

$$\bar{c}_1 = \frac{\eta_1}{\sqrt{\eta_1^2 - 1}} \bar{e}_1 < \left\| \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) \right\|$$

which implies that leader i only stops converging when it enters into the region bounded by

$$\left\| \frac{\partial \varphi}{\partial q}(t, q) \right\| \leq \bar{c}_1.$$

□

For the case of M targets, a similar result can be obtained.

Theorem 4.4. *For the energy intensity field $\varphi(t, q)$ defined by (4.2), the leaders finally converge into the region bounded by*

$$\left\| \frac{\partial \varphi}{\partial q}(t, q) \right\| \leq \bar{c}_2, \quad (4.23)$$

where $\bar{c}_2 = \sqrt{\bar{e}_2^2 + \left(\frac{MA_{\max}G_1}{\eta_2}\right)^2}$.

Proof. For the case of M targets, (4.18) evolves as

$$\frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) v_{r_{i,k+1}} > \sum_{m=1}^M \frac{\partial \varphi_m}{\partial q}(k, \mu_{i,k}) \dot{p}_m. \quad (4.24)$$

For $\left\| \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) \right\| \neq 0$, using the same definition of $\theta(k)$ as above, we get a sufficient condition for (4.24) as

$$\left\| \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) \right\| v_r \cos \theta(k) > \sum_{m=1}^M \left\| \frac{\partial \varphi_m}{\partial q}(k, \mu_{i,k}) \right\| \bar{p}$$

which is a necessary condition for

$$v_r \sqrt{\left\| \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) \right\|^2 - \|e_{i,k}\|^2} > M A_{\max} G_1 \bar{p}. \quad (4.25)$$

We can further get a sufficient condition for (4.25) as

$$\bar{c}_2 = \sqrt{\bar{e}_2^2 + \left(\frac{M A_{\max} G_1}{\eta_2} \right)^2} < \left\| \frac{\partial \varphi}{\partial q}(k, \mu_{i,k}) \right\|.$$

Hence, the conclusion holds. $\square$

Remark 4.5. *Based on Theorem 4.3 and Theorem 4.4, one can design the error bound $\bar{e}$ in theory as small as possible such that $\bar{c}_1$ and $\bar{c}_2$ are minimized. However, due to the system limit, $\bar{e}$ can never reach zero, which means $\bar{c}_1$ and $\bar{c}_2$ cannot reach zero as well.*

4.5 Simulation

4.5.1 Simulation Environment and Parameters

We use Matlab to implement simulations for the pursuit of both stationary and mobile targets. The energy intensity model in (4.1) is chosen as a Gaussian function, i.e., $\varphi(t, q) = A e^{-\lambda [q - \mu(t)]^T [q - \mu(t)]}$, where $\lambda > 0$ is the attenuation parameter. In our simulation, we set $\lambda = 0.01$. Initially, 40 agents are uniformly aligned over grids with size of $15 \times 15 \text{m}^2$ in a $90 \times 90 \text{m}^2$ region (as shown in Fig. 4.2 where red triangles denote the targets and blue stars denote the agents). System parameters are defined as follows: communication range $R_c = 50 \text{m}$, agent speed $v_r = 2 \text{m/s}$ and gain parameters $k_1 = 40$, $k_2 = 90$, $k_r = 200$, optimal distance between agents $d_0 = 20 \text{m}$. The sampling rate is set as $\frac{1}{T} = 5 \text{Hz}$. The bounds of target speed and target energy are set as $\bar{p} = 0.3 \text{m/s}$ and $A_{\max} = 50$ respectively. In the simulation, three targets are deployed (as shown in Fig. 4.2), where targets 1 and 2 are mobile

and target 3 is stationary. Their initial positions are $(73,82)$, $(125,133)$ and $(101,104)$ and energy intensities $A_1 = 50$, $A_2 = 40$ and $A_3 = 30$ respectively. The moving speeds of target 1 and 2 are $\|\dot{p}\|_1 = \|\dot{p}\|_2 = 0.3\text{m/s}$.

4.5.2 Simulation Results

Fig. 4.3 shows some snapshots of the target pursuit, and it can be seen that the leaders can pursue and localize the targets while the followers are gathering around the leaders and maintaining formations (Red triangles: targets; blue stars: leaders; blue curves: tracks of leaders; green squares: followers). The localization error for each target, i.e., the distance from each target to its nearest leader, is shown in Fig. 4.4. It can be seen that the leaders converge to the three targets with small error, which validates the effectiveness of the proposed target pursuit algorithm. The oscillation of the leaders around the targets is reasonable in the discrete-time implementations, and this can be prevented in the real applications by setting a command for agents to stop the pursuit after they have converged into the neighborhood of the local maximum points.

We also testify the boundness of the convergence of leaders in terms of the spatial gradient according to Theorem 4.3 and Theorem 4.4. To show this, we record the norm of the true spatial gradient at the position of each leader at each time step, which is denoted as $\text{Gra}_{l,k}$ for the l -th leader at time k . Then, we define

$$\text{Gra}_{\max,k} \triangleq \max_l \text{Gra}_{l,k}.$$

Fig. 4.5 shows that after a very short period of time, $\text{Gra}_{l,k}$ is consistently bounded by $\bar{c}_1$ and $\bar{c}_2$, which implies that the leaders converged into the region bounded by (4.19) or (4.23). It can be seen that the bound $\bar{c}_2$ for the case of multiple targets is pretty conservative. This is because it applies the energy intensity bound for each target and ignores possible cancelation in the sum of the vector products in (4.24). Generally, if the M targets are far away from each other such that the interference

among the individual energy intensity fields of the targets is negligible, the bound $\bar{c}_1$ for the case of single target can be applied.

4.6 Conclusions

In this chapter, a distributed multiple targets localization and pursuit scheme has been proposed for wireless robotic sensor networks, based on the discrete measurements of the unknown target energy intensity field. Robots, that are categorized into two types: leaders and followers, are driven by different control strategies to accomplish different tasks. The leaders take charge of the target localization and pursuit while the followers play a role to maintain the team formation and network connectivity. The relation between convergence error and system parameter setting was analyzed and simulation has shown the effectiveness of the proposed scheme.

(a) The initial deployment of targets and agents.

(b) The energy intensity field produced by the three targets.

Figure 4.2: Simulation setup.

Figure 4.3: The dynamic process of target pursuit.

Figure 4.4: The localization error for the three targets.

Figure 4.5: The boundness of the convergence region in terms of the spatial gradient.

Chapter 5

Multi-Agent Information Fusion and Cooperative Control in Target Search

In this chapter, the cooperative search of multiple stationary ground targets by a group of UAVs is investigated which fly over a two-dimensional plane with limited sensing and communication capabilities. In our setting, target existence within each cell is modeled as the Bernoulli distribution. By observing a cell, each agent obtains a 0 (no target detected) or 1 (target detected) detection result with fixed detection and false-alarm probabilities. In addition, each agent can only communicate with the agents that are within its communication range and keeps an individual probability map for the whole region. Our goal is first to design a distributed update and fusion scheme for the individual probability maps such that they all converge to the same one which reflects the true existence or nonexistence of targets within each cell. Second, design a distributed coverage and topology control algorithm such that the network is robust to variations of communication topologies, while seeking optimal trajectories to explore the surveillance region based on the updated probability maps.

Section 5.1 describes the basic notations and assumptions used in this chapter. Section 5.2 presents the probability map update by measurements based on Bayesian rule for each individual agent without information sharing. The map fusion scheme for multi-agent collaboration is proposed in Section 5.3. In Section 5.4, coverage and topology control algorithms are presented for path planning, and asynchronous implementations of the proposed fusion scheme are discussed. Simulation results are shown in Section 5.5, and this chapter is concluded in Section 5.6.

5.1 Basic Definitions and Assumptions

The surveillance region $\mathbb{O}$ is assumed to be on a plane ground and has been uniformly divided into M cells of the same size. By a slight abuse of notation, each cell is identified with its center $g = [x, y]^T$, where x and y are the coordinates of its center. All UAVs are assumed to move on a fixed plane above the surveillance region and thus the position of each agent can be described by its projection onto $\mathbb{O}$, which is denoted as $\mu_{i,k} = [x_{i,k}, y_{i,k}]^T$ for agent i ($i = 1, 2, \dots, N$) at time k , where $x_{i,k}$ and $y_{i,k}$ are the planar coordinates of its projection. Each agent is assumed to have access to its own position at any time. Each cell in the surveillance region is associated with a probability or a confidence level of target existence within the cell, modeled as a Bernoulli distribution, i.e. $\theta_g = 1$ (a target is present) with probability $P_{i,k}(\theta_g = 1)$ and $\theta_g = 0$ (no target is present) with probability $1 - P_{i,k}(\theta_g = 1)$ for agent i and cell g at time k . Targets are assumed to be present from the beginning of the search process and remain stationary throughout.

Agent i independently takes measurements $Z_{i,g,k}$ over the cells within its sensing region $\mathbb{C}_{i,k}$ at time k with sensing radius R_s (as shown in Fig.5.1), where $\mathbb{C}_{i,k} \triangleq \{g \in \mathbb{O} : \|g - \mu_{i,k}\| \leq R_s\}$. A cell is assumed to be wholly within $\mathbb{C}_{i,k}$ if its center is within $\mathbb{C}_{i,k}$. Only two observation results are defined for each cell, $Z_{i,g,k} = 0$ or $Z_{i,g,k} = 1$. For all cells, $P(Z_{i,g,k} = 1 | \theta_g = 1) = p$ and $P(Z_{i,g,k} = 1 | \theta_g = 0) = q$ are constants which are assumed to be known beforehand as the detection proba-

bility and false alarm probability respectively. The graph of the network and other elementary notations are defined in Appendix A.

Figure 5.1: Target search by multiple UAVs.

5.2 Uncooperative Probability Map Update

In a group of UAVs, each UAV agent i keeps an individual probability map $\mathcal{P}_{i,g,k}$ of the whole region, where $\mathcal{P}_{i,g,k} \triangleq P_{i,k}(\theta_g = 1)$. Basically, each agent should be able to update its own map based only on the measurements taken by itself without cooperation or information sharing with other agents. This would make the system robust to any loss of communication connectivity between some agents. Thus, in this section, we start our discussion from the probability map update by measurements for each individual agent in the group without cooperation.

The commonly used method of updating the probability map by measurements

is based on the Bayesian rule [111, 112], which is given by:

$$\begin{aligned} \mathcal{P}_{i,g,k} &= \frac{P(Z_{i,g,k}|\theta_g = 1) \mathcal{P}_{i,g,k-1}}{P(Z_{i,g,k}|\theta_g = 1) \mathcal{P}_{i,g,k-1} + P(Z_{i,g,k}|\theta_g = 0) (1 - \mathcal{P}_{i,g,k-1})} \\ &= \begin{cases} \frac{p \mathcal{P}_{i,g,k-1}}{p \mathcal{P}_{i,g,k-1} + q (1 - \mathcal{P}_{i,g,k-1})}, & \text{if } Z_{i,g,k} = 1, \\ \frac{(1-p) \mathcal{P}_{i,g,k-1}}{(1-p) \mathcal{P}_{i,g,k-1} + (1-q) (1 - \mathcal{P}_{i,g,k-1})}, & \text{if } Z_{i,g,k} = 0, \\ \mathcal{P}_{i,g,k-1}, & \text{otherwise.} \end{cases} \end{aligned} \quad (5.1)$$

It is easy to see that if $\mathcal{P}_{i,g,0} = 1$ (or 0), then $\mathcal{P}_{i,g,k} = 1$ (or 0, respectively) for all $k > 0$. On the other hand, for the special cases that either one of the two detection parameters, p and q , is equal to 0 or 1, (5.1) can be further simplified and such cases will not be discussed in detail due to its easy calculation. For example, if $p = 0$, $\mathcal{P}_{i,g,k}$ will become 0 once agent i gets a measurement of 1, and will remain unchanged from then on regardless of future measurements. In the following parts, we mainly consider the case that $0 < \mathcal{P}_{i,g,0} < 1$, $0 < p < 1$ and $0 < q < 1$, in which (5.1) is equivalent to

$$\frac{1}{\mathcal{P}_{i,g,k}} - 1 = \begin{cases} \frac{q}{p} \left(\frac{1}{\mathcal{P}_{i,g,k-1}} - 1 \right), & \text{if } Z_{i,g,k} = 1, \\ \frac{1-q}{1-p} \left(\frac{1}{\mathcal{P}_{i,g,k-1}} - 1 \right), & \text{if } Z_{i,g,k} = 0, \\ \frac{1}{\mathcal{P}_{i,g,k-1}} - 1, & \text{otherwise.} \end{cases} \quad (5.2)$$

Now, introduce the following nonlinear transformation of $\mathcal{P}_{i,g,k}$:

$$\mathcal{Q}_{i,g,k} \triangleq \ln \left(\frac{1}{\mathcal{P}_{i,g,k}} - 1 \right). \quad (5.3)$$

Then, (5.2) is transformed into

$$\mathcal{Q}_{i,g,k} = \mathcal{Q}_{i,g,k-1} + v_{i,g,k}, \quad (5.4)$$

where

$$v_{i,g,k} \triangleq \begin{cases} \ln \frac{q}{p}, & \text{if } Z_{i,g,k} = 1, \\ \ln \frac{1-q}{1-p}, & \text{if } Z_{i,g,k} = 0, \\ 0, & \text{otherwise.} \end{cases} \quad (5.5)$$

Compared with the Bayesian update in (5.1) which is a nonlinear function of $\mathcal{P}_{i,g,k-1}$, the update in (5.4), which is a linear function of $\mathcal{Q}_{i,g,k}$ and only involves simple summations, is more efficient in calculation. In addition, (5.3) is a bijective transformation of $\mathcal{P}_{i,g,k} \in (0, 1)$, so it can be recovered uniquely from $\mathcal{Q}_{i,g,k}$ whenever it is needed. Thus, storing $\mathcal{Q}_{i,g,k}$ instead of $\mathcal{P}_{i,g,k}$ in the memory of agents can help simplify the calculations.

Remark 5.1. *The logarithmic transformation of the Bayesian update is not new and has been used before such as in [119, 120], where the well-known information entropy is studied. However, the update of entropy by measurements is not linear which involves more complex computations. Furthermore, the update of entropy by information sharing between agents is linear only if the probability maps of the agents are independent, which is obviously not applicable for our problem. In [111], a heuristic approach of logarithmic transformation was proposed to obtain a linear map update by information sharing between agents. However, no theoretic analysis was made on the convergence of probability maps using the approach. Hypothesis testing and some other applications using logarithmic transformation of the Bayesian rule can be found in [121–123]. Our main contribution is not only making the update of probability maps by measurements or information sharing become linear, but also giving quantitative analysis on the convergence property of the proposed updating method, as will be shown later in this section.*

It can be seen that the evolution of $\mathcal{Q}_{i,g,k}$ depends on the number of observations, which is denoted as $m_{i,g,k}$, taken over the cell g up to time k . In real implementations, it is desired that $\mathcal{P}_{i,g,k}$ converges to 0 if no target is present within the cell g and converges to 1 otherwise as $m_{i,g,k} \rightarrow +\infty$, so that the converged probability map

reflects the true distribution of the targets. This result has been shown in [112]. However, as discussed before, updating $\mathcal{Q}_{i,g,k}$ is preferable to updating $\mathcal{P}_{i,g,k}$. Hence, in this chapter, we will analyze the asymptotic property of $\mathcal{Q}_{i,g,k}$ and include the result in [112] as part of our conclusion. For simplicity of notation, "almost surely" is abbreviated as "a.s."

Proposition 5.1. *Given the initial prior probability map $0 < \mathcal{P}_{i,g,0} < 1$ for each agent, the detection probability p and the false alarm probability q which are subjected to $q \neq p$ and $0 < p, q < 1$, the following conclusions hold by implementing the map updating rule (5.4).*

1. *If a target is present within the cell g , $\mathcal{Q}_{i,g,k} \xrightarrow{\text{a.s.}} -\infty$ (i.e., $\mathcal{P}_{i,g,k} \xrightarrow{\text{a.s.}} 1$) and $\frac{\mathcal{Q}_{i,g,k}}{m_{i,g,k}} \xrightarrow{\text{a.s.}} p \ln \frac{q}{p} + (1-p) \ln \frac{1-q}{1-p}$ as $m_{i,g,k} \rightarrow +\infty$.*
2. *If no target is present within the cell g , $\mathcal{Q}_{i,g,k} \xrightarrow{\text{a.s.}} +\infty$ (i.e., $\mathcal{P}_{i,g,k} \xrightarrow{\text{a.s.}} 0$) and $\frac{\mathcal{Q}_{i,g,k}}{m_{i,g,k}} \xrightarrow{\text{a.s.}} q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p}$ as $m_{i,g,k} \rightarrow +\infty$.*

Before showing the proof of Proposition 5.1, we first prove the following lemma.

Lemma 5.1. *If $p \neq q$, it holds that*

$$p \ln \frac{q}{p} + (1-p) \ln \frac{1-q}{1-p} < 0$$

$$q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p} > 0.$$

Proof. Consider the continuous function $f(\omega) = p \ln \omega + (1-p) \ln (1-\omega)$, where $\omega \in (0, 1)$. We can get

$$f'(\omega) = \frac{p}{\omega} - \frac{1-p}{1-\omega} = \frac{p-\omega}{\omega(1-\omega)}$$

which implies $f'(\omega) > 0$ if $\omega < p$ and $f'(\omega) < 0$ if $\omega > p$. Hence, for $\omega \neq p$, we have $f(\omega) < f(p)$. With $p \neq q$, it follows that $f(q) < f(p)$, i.e.,

$$p \ln q + (1 - p) \ln (1 - q) < p \ln p + (1 - p) \ln (1 - p). \quad (5.6)$$

From (5.6), we directly get

$$p \ln \frac{q}{p} + (1 - p) \ln \frac{1 - q}{1 - p} < 0.$$

Consider the continuous function $g(\omega) = q \ln \omega + (1 - q) \ln (1 - \omega)$, where $\omega \in (0, 1)$. We can get

$$g'(\omega) = \frac{q}{\omega} - \frac{1 - q}{1 - \omega} = \frac{q - \omega}{\omega(1 - \omega)}$$

which implies $g'(\omega) > 0$ if $\omega < q$ and $g'(\omega) < 0$ if $\omega > q$. Hence, for $\omega \neq q$, we have $g(\omega) < g(q)$. With $p \neq q$, it follows that $g(p) < g(q)$, i.e.,

$$q \ln p + (1 - q) \ln (1 - p) < q \ln q + (1 - q) \ln (1 - q). \quad (5.7)$$

From (5.7), we directly get

$$q \ln \frac{q}{p} + (1 - q) \ln \frac{1 - q}{1 - p} > 0.$$

□

Now, the Proposition 5.1 is proven.

Proof. From (5.4), we get

$$\mathcal{Q}_{i,g,k} = \mathcal{Q}_{i,g,0} + a_{i,g,k} \ln \frac{q}{p} + (m_{i,g,k} - a_{i,g,k}) \ln \frac{1 - q}{1 - p}, \quad (5.8)$$

where $a_{i,g,k}$ is the number of positive measurements taken over cell g by agent i up to time k .

We first prove the conclusion for case 1). $Z_{i,g,k}$ is a random variable generated based on the existence of targets, and it follows that $P_{i,g}(Z_{i,g,k} = 1) = p$ and

$P_{i,g}(Z_{i,g,k} = 0) = 1 - p$ if a target is present within the cell g . Meanwhile, $Z_{i,g,k}$ for all $k \geq 0$ are i.i.d. random variables. Then, according to the Strong Law of Large Numbers, we have $\frac{a_{i,g,k}}{m_{i,g,k}} \xrightarrow{a.s.} p$ as $m_{i,g,k} \rightarrow +\infty$, which implies

$$\begin{aligned} \frac{\mathcal{Q}_{i,g,k}}{m_{i,g,k}} &= \frac{\mathcal{Q}_{i,g,0}}{m_{i,g,k}} + \frac{a_{i,g,k}}{m_{i,g,k}} \ln \frac{q}{p} + \left(1 - \frac{a_{i,g,k}}{m_{i,g,k}}\right) \ln \frac{1-q}{1-p} \\ &\xrightarrow{a.s.} p \ln \frac{q}{p} + (1-p) \ln \frac{1-q}{1-p} < 0 \end{aligned}$$

as $m_{i,g,k} \rightarrow +\infty$. Hence, $\mathcal{Q}_{i,g,k} \xrightarrow{a.s.} -\infty$ as $m_{i,g,k} \rightarrow +\infty$.

In the same way, the conclusion for case 2) can be proven. $\square$

Remark 5.2. *Proposition 5.1 explicates how the environment parameters p and q influence the convergence performance of the probability maps. It can be seen that $\mathcal{P}_{i,g,k}$ is able to converge as long as the detection probability and false alarm probability are not equal, which has been previously shown in [112]. However, compared to [112], our result is less computationally demanding. Further, we analyze the influence of the detection parameters p and q on the convergence speed, which are important factors for real applications. In our work, the introduction of $\mathcal{Q}_{i,g,k}$ which is a nonlinear bijective transformation of $\mathcal{P}_{i,g,k}$ reduces the calculation complexity by transforming the nonlinear Bayesian updating into a linear one. Moreover, since $\frac{\mathcal{Q}_{i,g,k}}{m_{i,g,k}} = \frac{\mathcal{Q}_{i,g,k}/k}{m_{i,g,k}/k}$, where $\frac{m_{i,g,k}}{k}$ represents the average sampling rate of agent i over cell g , our result shows that $\mathcal{Q}_{i,g,k}$ approaches $-\infty$ or $+\infty$ at a speed proportional to the average sampling rate.*

Remark 5.3. *A potential problem that may be caused by storing $\mathcal{Q}_{i,g,k}$ in the memory is the data overflow for an extremely large or small value of $\mathcal{Q}_{i,g,k}$. This problem can be solved by setting a bound $\mathbf{Q} > 0$ such that*

$$\mathcal{Q}_{i,g,k} = \max(\min(\mathcal{Q}_{i,g,k}, \mathbf{Q}), -\mathbf{Q})$$

which implies $\mathcal{P}_{i,g,k} \in [\frac{1}{1+e^{\mathbf{Q}}}, \frac{1}{1+e^{-\mathbf{Q}}}]$. Therefore, $\mathbf{Q}$ should be large enough such that $\frac{1}{1+e^{\mathbf{Q}}}$ and $\frac{1}{1+e^{-\mathbf{Q}}}$ are close enough to 0 and 1 respectively.

5.3 Cooperative Probability Map Update

When a group of UAVs are deployed, the surveillance region is usually partitioned into multiple non-overlapping areas as task regions for different agents, and each agent searches targets within its own task region. Such a partition may not be static, e.g, it may be the Voronoi partition computed based on current agent positions in a real-time manner. Although each agent only observes one part of the whole region, it can still update its probability map by information sharing with other UAVs. Generally, the correlation between the individual probability maps of any two agents is unknown since the global topology and positions of all the UAVs are difficult to obtain by each agent in a large distributed network. Therefore, fusion of information from different agents is usually difficult. In this chapter, we strive to find a practical fusion method such that the individual probability maps of all UAVs can converge to the same one that reveals the true existence of the targets.

As discussed in Section 5.2, it is more energy-efficient for UAVs to store $\mathcal{Q}_{i,g,k}$ instead of $\mathcal{P}_{i,g,k}$. Hence, in this section it is assumed that each UAV only stores the records of $\mathcal{Q}_{i,g,k}$ for all cells. For a group of UAVs, we let each agent i at time k first take measurements and update $\mathcal{Q}_{i,g,k}$ using Bayesian rule, i.e.

$$H_{i,g,k} = \mathcal{Q}_{i,g,k-1} + v_{i,g,k},$$

where $v_{i,g,k}$ is defined in (5.5). Then, each agent i transmits the updated map to its neighbors for map fusion, using the consensus protocol proposed in [5, 38, 117]:

$$\mathcal{Q}_{i,g,k} = \sum_{j=1}^N w_{i,j,k} H_{j,g,k}, \quad (5.9)$$

where $w_{i,i,k} = 1 - \frac{d_{i,k}-1}{N}$, $w_{i,j,k} = \frac{1}{N}$ for $j \in \mathcal{N}_{i,k}$ ($j \neq i$) and $w_{i,j,k} = 0$ for $j \notin \mathcal{N}_{i,k}$.

Define the following augmented notations for the entire system:

$$\begin{aligned}\Upsilon_{g,k} &\triangleq [\mathcal{Q}_{1,g,k}, \mathcal{Q}_{2,g,k}, \dots, \mathcal{Q}_{N,g,k}]^T \\ \Phi_{g,k} &\triangleq [v_{1,g,k}, v_{2,g,k}, \dots, v_{N,g,k}]^T.\end{aligned}$$

We also define an $N \times N$ matrix W_k where the entry at the i -th row and j -th column is $[W_k]_{i,j} = w_{i,j,k}$. Then, the global form of the updating rule (5.9) for all agents can be written as

$$\Upsilon_{g,k} = W_k (\Upsilon_{g,k-1} + \Phi_{g,k}). \quad (5.10)$$

Before discussing the performance of such an updating rule, we need to define the following notations:

$$m_{g,k} \triangleq \sum_{i=1}^N m_{i,g,k}, \quad \mathcal{Q}_{g,k} \triangleq \sum_{i=1}^N \mathcal{Q}_{i,g,k} \quad (5.11)$$

Theorem 5.1. *Given the initial prior probability map $0 < \mathcal{P}_{i,g,0} < 1$ for each agent, the detection probability p and the false alarm probability q which are subjected to $q \neq p$ and $0 < p, q < 1$, if the network topology $\mathcal{G}_k$ is connected at all times, the following conclusions hold by implementing the map updating rule (5.10).*

1. *If a target is present within the cell g , $\mathcal{Q}_{i,g,k} \xrightarrow{a.s.} -\infty$ (i.e., $\mathcal{P}_{i,g,k} \xrightarrow{a.s.} 1$) and $\frac{\mathcal{Q}_{i,g,k}}{m_{g,k}} \xrightarrow{a.s.} \frac{p}{N} \ln \frac{q}{p} + \frac{1-p}{N} \ln \frac{1-q}{1-p}$ for all agents as $m_{g,k} \rightarrow +\infty$.*
2. *If no target is present within the cell g , $\mathcal{Q}_{i,g,k} \xrightarrow{a.s.} +\infty$ (i.e., $\mathcal{P}_{i,g,k} \xrightarrow{a.s.} 0$) and $\frac{\mathcal{Q}_{i,g,k}}{m_{g,k}} \xrightarrow{a.s.} \frac{q}{N} \ln \frac{q}{p} + \frac{1-q}{N} \ln \frac{1-q}{1-p}$ for all agents as $m_{g,k} \rightarrow +\infty$.*

Proof. From (5.9) and (5.10), we get

$$\Upsilon_{g,k} = W_k (\Upsilon_{g,k-1} + \Phi_{g,k}) = \prod_{t=1}^k W_t \Upsilon_{g,0} + \sum_{l=1}^k \prod_{t=l}^k W_t \Phi_{g,l}. \quad (5.12)$$

Now we define the deviation of $\Upsilon_{g,k}$ from the globally averaged map $\frac{1}{N}\mathcal{Q}_{g,k}\mathbf{1}$ as

$$\mathbf{e}_{g,k} \triangleq \Upsilon_{g,k} - \frac{1}{N}\mathcal{Q}_{g,k}\mathbf{1} = \Upsilon_{g,k} - \frac{1}{N}(\mathbf{1}^T \Upsilon_{g,k})\mathbf{1}. \quad (5.13)$$

Then, we get

$$\begin{aligned} \|\mathbf{e}_{g,k}\| &= \|\Sigma_{1,k} + \Sigma_{2,k}\| \leq \|\Sigma_{1,k}\| + \|\Sigma_{2,k}\| \\ &\leq \|\Sigma_{1,k}\| + \sum_{l=1}^k \left\| \prod_{t=l}^k W_t \Phi_{g,l} - \frac{1}{N}(\mathbf{1}^T \Phi_{g,l})\mathbf{1} \right\|, \end{aligned}$$

where

$$\begin{aligned} \Sigma_{1,k} &= \prod_{t=1}^k W_t \Upsilon_{g,0} - \frac{1}{N}(\mathbf{1}^T \Upsilon_{g,0})\mathbf{1}, \\ \Sigma_{2,k} &= \sum_{l=1}^k \left(\prod_{t=l}^k W_t \Phi_{g,l} - \frac{1}{N}(\mathbf{1}^T \Phi_{g,l})\mathbf{1} \right). \end{aligned}$$

Since $\mathbf{1}^T W_k = \mathbf{1}^T$, $W_k \mathbf{1} = \mathbf{1}$ and $[W_k]_{i,j} \geq 0$ for all i and j , W_k is a doubly stochastic matrix. According to Theorem 9 in [37], there exists a positive number $\lambda < 1$ such that for any vector $\xi \in \mathbb{R}^N$ and any W_k associated with a switching and connected topology $\mathcal{G}_k$,

$$\left\| W_k \xi - \frac{1}{N}(\mathbf{1}^T \xi)\mathbf{1} \right\| \leq \lambda \left\| \xi - \frac{1}{N}(\mathbf{1}^T \xi)\mathbf{1} \right\|.$$

Therefore, there exists a positive number $\lambda < 1$ such that

$$\|\mathbf{e}_{g,k}\| \leq \lambda^k \left\| \Upsilon_{g,0} - \frac{1}{N}(\mathbf{1}^T \Upsilon_{g,0})\mathbf{1} \right\| + \sum_{l=1}^k \lambda^{k-l+1} \left\| \Phi_{g,l} - \frac{1}{N}(\mathbf{1}^T \Phi_{g,l})\mathbf{1} \right\|. \quad (5.14)$$

It is easy to get

$$\left\| \Phi_{g,l} - \frac{1}{N}(\mathbf{1}^T \Phi_{g,l})\mathbf{1} \right\| < \sqrt{N} \left| \ln \frac{q}{p} - \ln \frac{1-q}{1-p} \right| \quad (5.15)$$

which implies

$$\begin{aligned}\|\mathbf{e}_{g,k}\| &< \lambda^k \left\| \Upsilon_{g,0} - \frac{1}{N} (\mathbf{1}^\top \Upsilon_{g,0}) \mathbf{1} \right\| + \sqrt{N} \left| \ln \frac{q}{p} - \ln \frac{1-q}{1-p} \right| \sum_{l=1}^k \lambda^{k-l+1} \\ &= \lambda^k \left\| \Upsilon_{g,0} - \frac{1}{N} (\mathbf{1}^\top \Upsilon_{g,0}) \mathbf{1} \right\| + \sqrt{N} \left| \ln \frac{q}{p} - \ln \frac{1-q}{1-p} \right| \frac{\lambda(1-\lambda^k)}{1-\lambda}.\end{aligned}$$

Thus, we have

$$\lim_{m_{g,k} \rightarrow +\infty} \|\mathbf{e}_{g,k}\| = \lim_{k \rightarrow +\infty} \|\mathbf{e}_{g,k}\| < \left| \ln \frac{q}{p} - \ln \frac{1-q}{1-p} \right| \frac{\lambda\sqrt{N}}{1-\lambda}. \quad (5.16)$$

On the other hand, from (5.12) we get

$$\begin{aligned}\mathcal{Q}_{g,k} &= \mathbf{1}^\top \Upsilon_{g,k} = \mathbf{1}^\top \Upsilon_{g,0} + \sum_{l=1}^k \mathbf{1}^\top \Phi_{g,l} = \sum_{i=1}^N \mathcal{Q}_{i,g,0} + \sum_{l=1}^k \sum_{i=1}^N v_{i,g,l} \\ &= \sum_{i=1}^N \mathcal{Q}_{i,g,0} + a_{g,k} \ln \frac{q}{p} + (m_{g,k} - a_{g,k}) \ln \frac{1-q}{1-p},\end{aligned} \quad (5.17)$$

where $a_{g,k}$ is the total number of positive measurements obtained by all agents up to time k . By comparing (5.17) with (5.8), we find that (5.17) can be seen as a map update for single agent exploration without cooperation by treating $\sum_{i=1}^N \mathcal{Q}_{i,g,0}$ as the initial map. Thus, based on the result of Proposition 5.1, we conclude that as $m_{g,k} \rightarrow +\infty$, $\mathcal{Q}_{g,k} \xrightarrow{a.s.} -\infty$ and $\frac{\mathcal{Q}_{g,k}}{m_{g,k}} \xrightarrow{a.s.} p \ln \frac{q}{p} + (1-p) \ln \frac{1-q}{1-p}$ for case 1), and $\mathcal{Q}_{g,k} \xrightarrow{a.s.} -\infty$ and $\frac{\mathcal{Q}_{g,k}}{m_{g,k}} \xrightarrow{a.s.} q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p}$ for case 2). Combining (5.13) and (5.16), we can get $\Upsilon_{g,k} \xrightarrow{a.s.} -\infty$ and $\frac{1}{m_{g,k}} \Upsilon_{g,k} \xrightarrow{a.s.} \frac{1}{N} \left(p \ln \frac{q}{p} + (1-p) \ln \frac{1-q}{1-p} \right) \mathbf{1}$ elementwise for case 1), and $\Upsilon_{g,k} \xrightarrow{a.s.} +\infty$ and $\frac{1}{m_{g,k}} \Upsilon_{g,k} \xrightarrow{a.s.} \frac{1}{N} \left(q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p} \right) \mathbf{1}$ elementwise for case 2). $\square$

Remark 5.4. *The basic idea of Theorem 5.1 is to let each agent get information of the region which cannot be detected by itself but can be detected by some other agents through communication. In the case 1) of Theorem 5.1, we can further derive that $\frac{\mathcal{Q}_{i,g,k}/k}{m_{g,k}/(Nk)} = \frac{\mathcal{Q}_{i,g,k}}{m_{g,k}/N} \xrightarrow{a.s.} p \ln \frac{q}{p} + (1-p) \ln \frac{1-q}{1-p} \triangleq b_1$ as $m_{g,k} \rightarrow +\infty$, where $\frac{m_{g,k}}{Nk}$ is the global average sampling rate (“global” means it is averaged over the N agents).*

Hence, $\mathcal{Q}_{i,g,k}$ approaches $-\infty$ at a speed proportional to the global average sampling rate. Similarly, in the case 2) we have $\frac{\mathcal{Q}_{i,g,k}/k}{m_{g,k}/(Nk)} \xrightarrow{a.s.} q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p} \triangleq b_2$ as $m_{g,k} \rightarrow +\infty$. Generally, the global average sampling rate for a cell is related to the deployment of UAVs including the sensing range, path planning and the number of deployed agents, etc. However, for all cells with identical sensors, we have $\frac{\sum_{g \in \mathbb{O}} m_{g,k}}{Nk} \leq M \frac{A_S}{A_{\mathbb{O}}}$, where $A_S = \pi R_s^2$ is the area of the sensing disk of each agent and $A_{\mathbb{O}}$ the area of the whole surveillance region. Then, we can get

$$\begin{aligned} \frac{1}{MN} \sum_{g \in \mathbb{O}} \frac{\|\mathcal{Q}_{i,g,k}\|}{k} &\leq \frac{1}{MN} \sum_{g \in \mathbb{O}} \frac{\|\mathcal{Q}_{i,g,0}\|}{k} + \frac{\sum_{g \in \mathbb{O}} m_{i,g,k}}{MNk} \max(|b_1|, |b_2|) \\ &\leq \frac{1}{MN} \sum_{g \in \mathbb{O}} \frac{\|\mathcal{Q}_{i,g,0}\|}{k} + \frac{A_S}{A_{\mathbb{O}}} \max(|b_1|, |b_2|) \end{aligned}$$

which implies

$$\lim_{k \rightarrow +\infty} \frac{1}{MN} \sum_{g \in \mathbb{O}} \frac{\|\mathcal{Q}_{i,g,k}\|}{k} \leq \frac{A_S}{A_{\mathbb{O}}} \max(|b_1|, |b_2|).$$

Since $\lim_{k \rightarrow +\infty} \frac{1}{MN} \sum_{g \in \mathbb{O}} \frac{\|\mathcal{Q}_{i,g,k}\|}{k}$ denotes the asymptotically averaged divergence speed over all agents and all cells in the surveillance region, we can predict the performance of the whole network by evaluating its upper bound:

$$\overline{Q} \triangleq \frac{A_S}{A_{\mathbb{O}}} \max(|b_1|, |b_2|) \quad (5.18)$$

which is a function of sensing range, detection and false alarm probabilities. In Section 5.5, we will examine the influence of different parameters on the convergence of probability maps by simulations, which will be compared with the result associated with $\overline{Q}$. Further, according to the Chebyshev's inequality with $\mathcal{Q}_{i,g,0} = 0$ for all i and g , it can be derived for case 1) and case 2) respectively that $\forall \varepsilon > 0$,

$$\begin{aligned} P\left(\left\|\frac{\mathcal{Q}_{i,g,k}}{m_{g,k}} - b_1\right\| \geq \varepsilon\right) &\leq \frac{p(1-p)}{m_{g,k}\varepsilon^2} \left[\left(\ln \frac{q}{p}\right)^2 + \left(\ln \frac{1-q}{1-p}\right)^2 \right], \\ P\left(\left\|\frac{\mathcal{Q}_{i,g,k}}{m_{g,k}} - b_2\right\| \geq \varepsilon\right) &\leq \frac{q(1-q)}{m_{g,k}\varepsilon^2} \left[\left(\ln \frac{q}{p}\right)^2 + \left(\ln \frac{1-q}{1-p}\right)^2 \right]. \end{aligned}$$

Then, the minimum number of observations over cell g by the whole team of agents for case 1) to satisfy $P\left(\left\|\frac{Q_{i,g,k}}{m_{g,k}} - b_1\right\| \geq \varepsilon\right) \leq \epsilon$ ($\forall \epsilon > 0$) is given by

$$m_{g,k} \geq \frac{p(1-p)}{\epsilon \varepsilon^2} \left[\left(\ln \frac{q}{p} \right)^2 + \left(\ln \frac{1-q}{1-p} \right)^2 \right]$$

and to satisfy $P\left(\left\|\frac{Q_{i,g,k}}{m_{g,k}} - b_2\right\| \geq \varepsilon\right) \leq \epsilon$ for case 2) by

$$m_{g,k} \geq \frac{q(1-q)}{\epsilon \varepsilon^2} \left[\left(\ln \frac{q}{p} \right)^2 + \left(\ln \frac{1-q}{1-p} \right)^2 \right].$$

Remark 5.5. We assume in (5.10) that only the individual probability maps are exchanged among neighbors. If neighboring agents exchange their current measurements as well in the meantime, each agent can get more measurements at each time than in the case without exchanging the measurements, which is equivalent to increasing the sensing areas A_s of agents or their global average sampling rates $\frac{m_{g,k}}{Nk}$ over the covered cells. For example, in addition to the measurements taken over the cells within its own sensing range, an agent can also get the measurements taken over the cells out of its sensing range but within the sensing ranges of its neighbors. Hence, the sensing area of the agent can be considered as increased from a spatial point of view, and the average sampling rates over the detected grids considered as increased from a temporal point of view. According to Theorem 5.1, it implies that the convergence of the probability maps is faster by exchanging the measurements.

Remark 5.6. It is assumed that the detection and false alarm probabilities are the same respectively for all cells in the surveillance region. However, in real systems, such probabilities may vary from cell to cell depending upon the environmental conditions or landscape of the surveillance region. With cell-dependent detection and false alarm probabilities, we can still prove the conclusions of Theorem 5.1 because there is no interactive influence between any two cells in the information sharing. Therefore, the individual probability map of each agent still converges as long as the detection and false alarm probabilities of each grid satisfy the constraint posed in

Theorem 5.1.

Remark 5.7. *The communication cost for exchanging the measurements and probability maps is not considered here. However, it may have a huge impact on some power-constrained applications. Since the communication cost is proportional to the number of cells in the whole region, users need to trade off the cost against accuracy of target localization by selecting an adequate number of cells to partition the region.*

5.4 Path Planning and Asynchronous Implementations

5.4.1 Coverage Control

Path planning plays an important role in the control of UAVs for target search. In our case, each UAV observes the cells within its sensing region $\mathbb{C}_{i,k}$ which is a disk. Therefore, we formulate it as a coverage control problem, i.e., to find an optimal configuration of all agents that minimizes a given coverage performance cost function. We consider the following first-order discrete-time motion model of UAVs:

$$\mu_{i,k} = \mu_{i,k-1} + u_{i,k}. \quad (5.19)$$

Remark 5.8. *Note that it is not a dynamic model of UAVs (which is usually continuous-time) but a discrete-time motion model. It would be complicated to include dynamic models which can be non-linear and high-order differential equations. There is an inner loop controller that handles the UAV dynamics and controls the attitude of UAVs which is not discussed in this chapter as in many existing works [11, 30, 124]. It is pointed out in [124] that any control law for the discrete-time motion model can be implemented in the continuous-time networks. Since we do not want to limit our discussion onto any specific nonlinear UAV dynamic model, only the discrete-time motion model is used for future extensions.*

In [11], a distributed multi-agent coverage control method is proposed. The coverage performance cost function is defined as

$$\mathcal{H}(\mu_1, \dots, \mu_N) = \frac{1}{2} \sum_{i=1}^N \int_{\mathbb{V}_i} \|r - \mu_i\|^2 \phi(r) dr, \quad (5.20)$$

where $r \in \mathbb{O}$, $\mathbb{V}_i$ is the Voronoi region generated by μ_i , and $\phi(r)$ is a given density function. The mass and centroid of each region $\mathbb{V}_i$ are defined respectively as

$$A_i \triangleq \int_{\mathbb{V}_i} \phi(r) dr, \quad \text{CM}_i \triangleq \frac{\int_{\mathbb{V}_i} r \phi(r) dr}{A_i}. \quad (5.21)$$

It has been proven in [11] that among all configurations in $\mathbb{O}$ the configuration $\{\mu_1, \mu_2, \dots, \mu_N\} = \{\text{CM}_1, \text{CM}_2, \dots, \text{CM}_N\}$ is optimal for $\mathcal{H}$. Since

$$\frac{\partial \mathcal{H}}{\partial \mu_i} = A_i (\text{CM}_i - \mu_i),$$

a gradient-based control law to minimize $\mathcal{H}$ is derived as

$$u_{i,k} = -K_u (\text{CM}_{i,k} - \mu_{i,k-1}), \quad (5.22)$$

where K_u is a positive gain parameter. Note that the convergence of agent positions can be guaranteed with $K_u \leq 1$ according to the discussions in [11]. Since the mobility of agents is usually constrained by a maximum speed $\|u\|_{\max}$, the control law in (5.22) should be chosen by the following saturation rule:

$$u_{i,k} = \begin{cases} u_{i,k}, & \text{if } \|u_{i,k}\| \leq \|u\|_{\max}; \\ u_{i,k} \frac{\|u\|_{\max}}{\|u_{i,k}\|}, & \text{otherwise.} \end{cases} \quad (5.23)$$

We adopt the control law (5.23) for our problem, where the integrals in (5.21) are approximated by the discrete summations, i.e., $A_{i,k} = \sum_{\mathbb{V}_{i,k}} \phi(g) \varsigma^2$ and $\text{CM}_{i,k} = \frac{1}{A_{i,k}} \sum_{\mathbb{V}_{i,k}} g \phi(g) \varsigma^2$, where ς and ς^2 are respectively the edge length and the area of each cell. The only thing left which is not clarified is the choice of the density

function $\phi(g)$. In [11], a fixed $\phi(g)$ is used for all agents to assign different weights to different cells, and the coverage of mobile agents will concentrate more on the cells that have higher weights. For our target search problem, each cell g needs to be observed a sufficient number of times such that the existence of target within the cell is determined, i.e., $\mathcal{P}_{i,g,k}$ approaches 0 or 1. Thus, it can be said intuitively that the closer $\mathcal{P}_{i,g,k}$ is to the middle between 0 and 1 (correspondingly, $\|\mathcal{Q}_{i,g,k}\|$ approaches 0), the harder we can determine if a target exists within the cell. That is, the smaller $\|\mathcal{Q}_{i,g,k}\|$ is, the larger uncertainty agent i has about target existence within cell g , and the uncertainty goes to zero if $\|\mathcal{Q}_{i,g,k}\|$ approaches infinity. Hence, we define the following uncertainty for each cell:

$$\eta_{i,g,k} \triangleq e^{-K_\eta \|\mathcal{Q}_{i,g,k}\|}, \quad (5.24)$$

where K_η is a positive gain parameter. Generally, more attention should be paid to regions with higher uncertainties. Based on this intuition, we choose the uncertainty of each cell as the its weight of coverage, i.e., let

$$\phi_{i,g,k} = \eta_{i,g,k}. \quad (5.25)$$

Since $\phi_{i,g,k}$ is related to the probability map $\mathcal{P}_{i,g,k}$ which is time-varying, $\phi_{i,g,k}$ may differ from agent to agent in the initial stage of search and is updated in a real-time manner. However, as stated in Theorem 5.1, the probability maps of all agents will approach the same one as the number of measurements for each cell approaches infinity, which implies the convergence of $\phi_{i,g,k}$ to the same one for all agents, and thus the same result of (5.22) with a common fixed density function can still be obtained.

Remark 5.9. *The Voronoi partition $\{\mathbb{V}_{1,k}, \mathbb{V}_{2,k}, \dots, \mathbb{V}_{N,k}\}$ may not be globally implemented because each agent may not have access to the positions of all the other agents due to the limited communication range. Each agent can compute its own Voronoi region $\mathbb{V}_{i,k}$ only based on the positions of its neighbors in a localized manner,*

which may induce overlaps between different Voronoi regions, i.e., $\mathbb{V}_{i,k} \cap \mathbb{V}_{j,k} \neq \emptyset$ for some $j \notin \mathcal{N}_{i,k}$. Although this may vary the moving path as derived, it does not affect the final convergence of probability maps since the agents are always navigated to the places which have higher uncertainties according to (5.25), and the uncertainties as well as the densities will approach 0 for the cells with sufficient number of observations no matter along which paths the agents move. On the other hand, we have $\mathbb{V}_{i,k} \cap \mathbb{V}_{j,k} = \emptyset$ for all $j \in \mathcal{N}_{i,k}$, which means the collision between agents can always be avoided. Once two agents come into the communication range, there will be no overlap between their Voronoi regions. Since each agent only moves inside its own Voronoi region towards the centroid at each time, the agents will never collide with each other.

Remark 5.10. To avoid the case that agents stop at local optimum solutions, we can give a perturbation to an agent if the total uncertainty over its individual probability map is nonzero when its control input given by (5.22) approaches zero. For example, we may set the control law of an agent as a vector with appropriate length pointing towards the cell that takes the largest uncertainty over its whole individual probability map (we can select the cell with the smallest index number if more than one cell takes the largest uncertainty). However, the control input should be bounded such that the agent will not jump out of its current Voronoi region for collision avoidance, i.e., to make $\mu_{i,k+1} \in \mathbb{V}_{i,k}$ all the time.

Remark 5.11. The gain parameter K_u in the control law (5.22) is related to the discrete step size of each movement of agents within a sampling interval. In practice, a larger value for the gain parameter may result in faster movement of agents, but the moving path may deviate more greatly from the optimal one derived by the continuous gradient. Users should make a balance between the two factors based on their needs. The gain parameter K_η in the uncertainty function (5.24) is used for differentiating the weights of different cells. In theory, any positive value of K_η satisfies the need, and in practice, we just need to make it not too small or too large so that the uncertainty function does not attenuate too sharply or too slowly.

5.4.2 Connectivity Maintenance

The key idea of maintaining the network connectivity is to restrict the allowable motion of each agent, i.e., to find a set $\chi_{i,k}$ of the control inputs for each agent such that the network is still connected at time $k + 1$ by selecting $u_{i,k} \in \chi_{i,k}$ for each agent if the network is connected at time k . In [124], a distributed topology control algorithm has been proposed to maintain so-called “pairwise connectivity”, that is, each two agents that are within the communication range R_c at time k are still within range R_c at time $k + 1$. It is clear that this algorithm is quite conservative since the network connectivity can still be maintained after some pairs of agents are disconnected. Our work is based on the connectivity maintenance algorithm proposed in [124], but aims to relax the constraints and give agents more freedom to select proper control inputs.

Consider two agents i and j at time k which satisfy $\|\mu_{i,k} - \mu_{j,k}\| \leq R_c$. In [124], the pairwise connectivity constraint set of agent i with respect to agent j ($i \neq j$) is defined as

$$\Omega_{i,j,k} \triangleq \left\{ \xi \in \mathbb{O} : \left\| \xi - \frac{\mu_{i,k} + \mu_{j,k}}{2} \right\| \leq \frac{R_c}{2} \right\}.$$

The constrained control input set of agent i is given as

$$\chi_{i,k} = \left\{ u : \mu_{i,k} + u \in \bigcap_{j \in \mathcal{N}_{i,k} \setminus \{i\}} \Omega_{i,j,k} \right\}.$$

Then, by selecting $u_{i,k} \in \chi_{i,k}$ for each agent, the connected pairs of agents at time k are still connected at time $k + 1$. Compared with $\chi_{i,k}$ given in [124], a relaxed constrained control input set is given by the following proposition.

Proposition 5.2. *If the network is connected at time k , then the network is still*

connected at time $k + 1$ by selecting $u_{i,k} \in \bar{\chi}_{i,k}$ for each agent, where

$$\begin{aligned}\bar{\chi}_{i,k} &= \left\{ u : \mu_{i,k} + u \in \bigcap_{j \in \bar{\mathcal{N}}_{i,k}} \Omega_{i,j,k} \right\}, \\ \bar{\mathcal{N}}_{i,k} &\triangleq \left\{ j \in \mathcal{N}_{i,k} \setminus \{i\} : \min_{s \in \mathcal{N}_{i,k} \cap \mathcal{N}_{j,k}} s = i \text{ or } j \right\},\end{aligned}\tag{5.26}$$

and it holds that $\bar{\chi}_{i,k} \supset \chi_{i,k}$.

Proof. Since the network is connected at time k , $\bar{\mathcal{N}}_{i,k} \neq \emptyset$, i.e., $\bar{\chi}_{i,k}$ exists. Moreover, $\bar{\mathcal{N}}_{i,k} \subset \mathcal{N}_{i,k} \setminus \{i\}$ implies $\bar{\chi}_{i,k} \supset \chi_{i,k}$. For any agent i and its neighbor $j \in \mathcal{N}_{i,k} \setminus \{i\}$, if $u_{i,k} \in \{u : \mu_{i,k} + u \in \Omega_{i,j,k}\}$ and $u_{j,k} \in \{u : \mu_{j,k} + u \in \Omega_{j,i,k}\}$, then $j \in \mathcal{N}_{i,k+1}$ holds according to Lemma 4.2 in [124]. Further, according to the definition of $\bar{\mathcal{N}}_{i,k}$, we have that $j \in \bar{\mathcal{N}}_{i,k}$ implies $i \in \bar{\mathcal{N}}_{j,k}$, that is, $u_{i,k} \in \{u : \mu_{i,k} + u \in \Omega_{i,j,k}\}$ implies $u_{j,k} \in \{u : \mu_{j,k} + u \in \Omega_{j,i,k}\}$. Hence, for any agent $j \in \bar{\mathcal{N}}_{i,k}$, $j \in \mathcal{N}_{i,k+1}$ holds, i.e., the edge $\{i, j\}$ is maintained at time $k + 1$.

Now, we consider the agent $j \in \mathcal{N}_{i,k} \setminus (\bar{\mathcal{N}}_{i,k} \cup \{i\})$. Under the control laws $u_{i,k} \in \bar{\chi}_{i,k}$ and $u_{j,k} \in \bar{\chi}_{j,k}$, $j \notin \mathcal{N}_{i,k+1}$ only if $\min(i, j) \neq s_1 \triangleq \min_{s \in \mathcal{N}_{i,k} \cap \mathcal{N}_{j,k}} s$, i.e., the edge $\{i, j\}$ is eliminated as a redundant one at time $k + 1$ only if there is a path $\{i, s_1\}$, $\{s_1, j\}$ which connects i and j at time k . Since the network scale is finite with N agents in total, any agent $j \in \mathcal{N}_{i,k} \setminus (\bar{\mathcal{N}}_{i,k} \cup \{i\})$ will be connected with agent i at time $k + 1$ even with elimination of the edge $\{i, j\}$ under the given control laws.

In all, for any agent i , the agents in $\mathcal{N}_{i,k}$ are still connected at time $k + 1$ under the given control laws, and thus the whole network is still connected. $\square$

Remark 5.12. Basically, the topology control method given by Proposition 5.2 aims to maintain a connected topology for the network while minimizing the constraints on the control inputs of agents. The physical meaning of (5.26) is that an edge between two agents is recognized as redundant and does not have to be maintained if they have a common neighbor with a smaller identity number than theirs. Different from the one proposed in [124], our method allows topological switching during the

group motion and the redundant edges for a connected graph are not necessarily maintained. Hence, our method gives more freedom for the agents to control their motion during target search. Meanwhile, since $\bar{\mathcal{N}}_{i,k}$ only involves $\mathcal{N}_{i,k} \cap \mathcal{N}_{j,k}$, agent i can compute the control input $u_{i,k}$ distributively merely based on the positions of its neighbors, which requires each agent to transmit its own position to its neighbors. Then, agent i determines $s \in \mathcal{N}_{i,k} \cap \mathcal{N}_{j,k}$, if $\|\mu_{s,k} - \mu_{j,k}\| \leq R_c$ and $s \in \mathcal{N}_{i,k}$.

5.4.3 Asynchronous Sampling and Communication

The asynchronous implementation of the control law (5.22) has been addressed by [11]. In this part, we mainly focus on the asynchronous implementation of the updating rule (5.10) proposed in Section 5.3, i.e., asynchronous sampling and communication.

For simplicity, we assume that all agents broadcast messages to their neighbors synchronously at a fixed communication frequency F_C and sample the environment synchronously at a fixed sampling frequency F_S . Further, for the ease of calculation, the larger one in the two frequencies is assumed to be an integral multiple of the smaller one. We only consider the impact of the synchronicity due to the difference between F_C and F_S . In the real applications, it is common to have different frequencies for sampling and communication in order to balance the energy consumption and information update. For example, the communication among multiple UAVs may be both time and energy consuming so that the a low communication frequency F_C is chosen for multi-agent collaboration, but a high sampling frequency F_S is set for fast information update for single agent exploration. In Section 5.3, the performance of the map updating rule (5.10) has been analyzed for the case of $F_C = F_S$. In this part, we will discuss the performance in the cases of $F_S = \kappa F_C$ and $F_C = \kappa F_S$, where $\kappa \geq 1$ is an integer. For simplicity of notation, we assume $\max(F_S, F_C) = 1$.

We first discuss the case of $F_S = \kappa F_C$, which means more than one measurement over the same cell can be obtained by a single agent within the communication

interval $\frac{1}{F_C}$. In this case, we use $k > 0$ to denote the sampling time index, and the communication among all agents only occurs at time instants $k = l\kappa$, where l is a positive integer. At time $k \neq l\kappa$, there is no communication and we set W_k as an identity matrix for the entire system. Then, at time $k = l\kappa$, the map updating rule (5.10) is reformulated as

$$\begin{aligned}\Upsilon_{g,k} &= \Upsilon_{g,l\kappa} = W_{l\kappa} \left(\Upsilon_{g,(l-1)\kappa} + \sum_{t=(l-1)\kappa+1}^{l\kappa} \Phi_{g,t} \right) \\ &= \prod_{h=1}^l W_{h\kappa} \Upsilon_{g,0} + \sum_{f=1}^l \prod_{h=f}^l W_{h\kappa} \sum_{t=(f-1)\kappa+1}^{f\kappa} \Phi_{g,t}.\end{aligned}\tag{5.27}$$

And at time $(l-1)\kappa < k < l\kappa$, (5.10) is reformulated as

$$\Upsilon_{g,k} = \Upsilon_{g,l\kappa} + \sum_{t=(l-1)\kappa+1}^k \Phi_{g,t}.\tag{5.28}$$

For the case of $F_C = \kappa F_S$, we use $k > 0$ to denote the communication time index, and each agent only takes measurements at time $k = (l-1)\kappa + 1$, where l is a positive integer. At time $k \neq (l-1)\kappa + 1$, no measurement is taken and we set $\Phi_{g,k}$ as a zero vector. Then, at time $k = l\kappa$, (5.10) is replaced by

$$\begin{aligned}\Upsilon_{g,k} &= \Upsilon_{g,l\kappa} = \prod_{t=(l-1)\kappa+1}^{l\kappa} W_t \left(\Upsilon_{g,(l-1)\kappa} + \Phi_{g,(l-1)\kappa+1} \right) \\ &= \prod_{t=1}^{l\kappa} W_t \Upsilon_{g,0} + \sum_{h=1}^l \prod_{t=(h-1)\kappa+1}^{l\kappa} W_t \Phi_{g,(h-1)\kappa+1}.\end{aligned}\tag{5.29}$$

And at time $(l-1)\kappa < k < l\kappa$, (5.10) is replaced by

$$\Upsilon_{g,k} = \prod_{t=(l-1)\kappa+1}^k W_t \left(\Upsilon_{g,(l-1)\kappa} + \Phi_{g,(l-1)\kappa+1} \right).\tag{5.30}$$

Corollary 5.1. *Given the initial prior probability map $0 < \mathcal{P}_{i,g,0} < 1$ for each agent, the detection probability p and the false alarm probability q which are subjected to*

$q \neq p$ and $0 < p, q < 1$, if the network topology $\mathcal{G}_k$ is connected at all times and $F_S = \kappa F_C$ or $F_C = \kappa F_S$, the conclusions of Theorem 5.1 hold by implementing the map updating rule (5.10).

Proof. We first prove the conclusions under $F_S = \kappa F_C$. According to the proof of Theorem 5.1, the key of the proof of Theorem 5.1 is to show that $\|\mathbf{e}_{g,k}\|$ as defined in (5.13) is bounded. From (5.27), for $k = l\kappa$ and some positive number $\lambda < 1$ we can get

$$\begin{aligned} \|\mathbf{e}_{g,k}\| &\leq \lambda^l \left\| \Upsilon_{g,0} - \frac{1}{N} (\mathbf{1}^T \Upsilon_{g,0}) \mathbf{1} \right\| + \sum_{f=1}^l \lambda^{l-f+1} \\ &\quad \times \left\| \sum_{t=(f-1)\kappa+1}^{f\kappa} \Phi_{g,t} - \frac{1}{N} \left(\mathbf{1}^T \sum_{t=(f-1)\kappa+1}^{f\kappa} \Phi_{g,t} \right) \mathbf{1} \right\| \\ &< \lambda^l \left\| \Upsilon_{g,0} - \frac{1}{N} (\mathbf{1}^T \Upsilon_{g,0}) \mathbf{1} \right\| \\ &\quad + \sum_{f=1}^l \lambda^{l-f+1} \kappa \sqrt{N} \left| \ln \frac{q}{p} - \ln \frac{1-q}{1-p} \right|. \end{aligned}$$

From (5.28), for $(l-1)\kappa < k < l\kappa$ we have

$$\begin{aligned} \|\mathbf{e}_{g,k}\| &= \|\mathbf{e}_{g,(l-1)\kappa} + \Xi_{g,k}\| \leq \|\mathbf{e}_{g,(l-1)\kappa}\| + \|\Xi_{g,k}\| \\ &\leq \|\mathbf{e}_{g,(l-1)\kappa}\| + (k - (l-1)\kappa) \sqrt{N} \left| \ln \frac{q}{p} - \ln \frac{1-q}{1-p} \right| \\ &< \|\mathbf{e}_{g,(l-1)\kappa}\| + \kappa \sqrt{N} \left| \ln \frac{q}{p} - \ln \frac{1-q}{1-p} \right|, \end{aligned}$$

where

$$\Xi_{g,k} = \sum_{t=(l-1)\kappa+1}^k \Phi_{g,t} - \frac{1}{N} \left(\mathbf{1}^T \sum_{t=(l-1)\kappa+1}^k \Phi_{g,t} \right) \mathbf{1}.$$

Hence, we have

$$\lim_{m_{g,k} \rightarrow +\infty} \|\mathbf{e}_{g,k}\| < \left| \ln \frac{q}{p} - \ln \frac{1-q}{1-p} \right| \frac{\kappa \sqrt{N}}{1-\lambda}.$$

Therefore, the conclusions of Theorem 5.1 still hold for the case of $F_S = \kappa F_C$.

Following the same procedures above, in the case of $F_C = \kappa F_S$, we can get for some positive number $\lambda < 1$,

$$\lim_{m_{g,k} \rightarrow +\infty} \|\mathbf{e}_{g,k}\| < \left| \ln \frac{q}{p} - \ln \frac{1-q}{1-p} \right| \frac{\sqrt{N}}{1-\lambda^\kappa},$$

implying that the conclusions of Theorem 5.1 hold. $\square$

Remark 5.13. *Although we assume that all agents have the same sampling frequency F_S and communication frequency F_C in the discussions above, the same conclusions can be obtained if the agents have different sampling and communication frequencies. Based on the conclusions in [25, 37], one can easily show that Theorem 5.1 still holds following the same procedure of proof above as long as the network is connected and there exist positive numbers $\bar{F}_S, \bar{F}_C$ such that the sampling frequency F_{S_i} , the communication frequency F_{C_i} of each agent i are subject to $F_{S_i} \geq \bar{F}_S$ and $F_{C_i} \geq \bar{F}_C$ respectively.*

5.5 Simulation

5.5.1 Simulation Environment

In this section, we test the performance of our proposed target search method in different scenarios and examine the influence of different parameters on the convergence of individual probability maps. Eight targets randomly appear at the beginning of simulations in a square surveillance region $([0, 25] \times [0, 25] \text{ m}^2)$ as shown in Fig. 5.2 (Red squares: targets). The agents are all randomly deployed initially within a small region so that they are connected at the beginning. In Scenario I, we use different number of agents to test its influence on the convergence performance, and keep the sensing range as $R_s = 4\text{m}$, the communication range as $R_c = 40\text{m}$ which is large enough to maintain all to all communications so that the influence of the communication control is ignored, the detection and false alarm probabilities

respectively as $p = 0.9$ and $q = 0.3$, and the ratio between sampling frequency and communication frequency as $\frac{F_S}{F_C} = 1\text{Hz}$. In Scenario II, we set different values of sensing range to test its influence on the convergence performance, and keep $N = 6$, $R_c = 40\text{m}$, $p = 0.9$, $q = 0.3$ and $\frac{F_S}{F_C} = 1$. In Scenario III, we set different detection probability to test its influence on the convergence performance, and keep $N = 6$, $R_s = 4\text{m}$, $R_c = 40\text{m}$, $q = 0.3$ and $\frac{F_S}{F_C} = 1$. In Scenario IV, we set different ratios between sampling frequency and communication frequency to test the influence of asynchronous sampling and communication on the convergence performance, and keep $N = 6$, $R_s = 4\text{m}$, $R_c = 40\text{m}$, $p = 0.9$ and $q = 0.3$. In Scenario V, we set different communication ranges to test the influence of the connectivity maintenance on the convergence performance, and keep $N = 6$, $R_s = 4\text{m}$, $p = 0.9$, $q = 0.3$ and $\frac{F_S}{F_C} = 1$. In Scenario VI, we further test the improvement of our modified connectivity maintenance scheme (5.26) over the one proposed in [124], where $N = 10$, $R_s = 3\text{m}$, $p = 0.9$, $q = 0.3$ and $\frac{F_S}{F_C} = 1$. In addition to the above mentioned parameters, we set $\mathcal{Q}_{i,g,0} = 0$ for all i and g , and $K_u = 1$, $K_\eta = 2$, $\|u\|_{\max} = 3\text{m/s}$, $\max(F_S, F_C) = 1\text{Hz}$.

Figure 5.2: Surveillance region and target positions.

Since the convergence of the individual probability map $\mathcal{P}_{i,g,k}$ of agent i implies that the uncertainty $\eta_{i,g,k}$ defined by (5.24) approaches 0 for each cell, we define the

following average uncertainty to evaluate the convergence performance of the whole network:

$$\boldsymbol{\eta}_k = \frac{1}{NM} \sum_{i=1}^N \sum_{g \in \mathbb{O}} \eta_{i,g,k}.$$

It is easy to find that the initial value of $\boldsymbol{\eta}_k$ is $\boldsymbol{\eta}_0 = \frac{1}{NM} \sum_{i=1}^N \sum_{g \in \mathbb{O}} e^{-K_\eta \|\mathcal{Q}_{i,g,0}\|} = 1$. From Scenario I to Scenario IV, the communication is large enough for all agents to have immediate consensus of the individual probability maps after each communication. However, in Scenario V, due to limited communication range, the deviation between each two individual probability maps exists at each time. Hence, in Scenario V we further define the following average uncertainty deviation to evaluate the consensus performance of the whole network with limited communication range:

$$D_{\eta_k} = \frac{1}{NM} \sum_{i=1}^N \sum_{g \in \mathbb{O}} \|\eta_{i,g,k} - \boldsymbol{\eta}_k\|.$$

Further, from (5.18), we can get a lower bound of $\eta_{i,g,k}$ associated with $\bar{Q}_k$ as

$$\bar{\eta}_k \triangleq e^{-K_\eta \bar{Q}_k k} \quad (5.31)$$

which is a function of sensing range R_s , detection and false alarm probabilities p and q . Hence, we compare the simulation results with $\bar{\eta}_k$ under fixed R_s , p and q . Due to the randomness of detection, the final results are averaged over 100 simulations for each case.

5.5.2 Simulation Results

We first give a sketch of the convergence process of the individual probability maps and only select one agent for illustration due to the limitation of space as shown in Fig. 5.3, where the probabilities converge to 1 for the cells within which targets truly exist and 0 for the cells within which no target exists.

Through all the scenarios, we can summarize that the larger the number of agents

Figure 5.3: The convergence of the probability map of agent 1.

(as shown in Fig. 5.4), the sensing range (as shown in Fig. 5.5), the sampling frequency (as shown in Fig. 5.6), the detection probability (as shown in Fig. 5.7) and the communication range (as shown in Fig. 5.8(a)), the faster the average uncertainty decreases to 0. Furthermore, the larger the number of agents and the sampling frequency, the closer the simulation results to the lower bound given by (5.31). Specifically, in Scenario V (as shown in Fig. 5.8(b)), the larger the communication range, the smaller the average deviation of the uncertainties which means the consensus performance is better. The convergence speed is higher with a larger communication range because the topology control by Proposition 5.2 gives more freedom for the agents to seek the optimal coverage control input. In Scenario VI (as shown in Fig. 5.9), the average uncertainty converges faster by our modified connectivity maintenance algorithm (5.26) (denoted by "Modified Alg") than by the original algorithm proposed in [124] (denoted by "Original Alg"). This is because the modified algorithm sets less constraints on the moving of agents.

Figure 5.4: The results in Scenario I.

Figure 5.5: The results in Scenario II.

Figure 5.6: The results in Scenario III.

Figure 5.7: The results in Scenario IV.

5.6 Conclusions

In this chapter, we have studied the cooperative control and information fusion in target search by a group of UAVs. By dividing the whole surveillance region into cells, each agent keeps an individual probability map about the target existence within each cell. The map is first updated by local measurements based on Bayesian rule, which was simplified to a linear update by introducing a nonlinear transformation of the probability map. A consensus-like distributed fusion scheme was further proposed for multi-agent map fusion. It has been proven that all the individual probability maps converge to the same one that reveals the true existence or nonexistence of targets within each cell. A path planning algorithm for coverage optimization and connectivity maintenance was designed and the performance of the fusion scheme for asynchronous implementations of sampling and communication was analyzed. The simulation results showed that the proposed algorithms can make the individual probability maps of all agents converge to the same one which reflects the true environment.

(a) Results of η_k . (b) Results of D_{η_k} .

Figure 5.8: The results in Scenario V.

Figure 5.9: The results in Scenario VI.

Chapter 6

Vision-Based Multi-Agent Cooperative Target Search

This chapter deals with the vision-based cooperative search for multiple ground mobile targets by a group of UAVs with limited sensing and communication capabilities which have a three-dimensional discrete-time motion model. Based on the probability map updating model proposed in Chapter 5, the model is generalized by considering the information decaying and transmission between cells due to environmental changes such as the target movement. Furthermore, a time-varying detection probability of an agent is considered as a function of its height above the ground. The target search is then formulated as a coverage optimization problem by trading off the the coverage area and detection performance.

Section 6.1 describes the basic notations and assumptions used in this chapter. Section 6.2 presents the probability map updates by measurements and information sharing with time-varying detection probabilities. In Section 6.3, a novel three-dimensional coverage control method is presented for target search. Simulation results are shown in Section 6.4, and the conclusions are drawn in Section 6.5.

6.1 Basic Definitions and Assumptions

The surveillance region $\mathbb{O} \in \mathbb{R}^2$ is assumed to be on a plane ground and has been uniformly divided into a set of cells of the same size. We assume that all UAVs use the same global Cartesian coordinate system and the position of each agent is denoted as $\mu_{i,k} = [c_{i,k}^T, h_{i,k}]^T \in \mathbb{R}^3$ for agent i ($i = 1, 2, \dots, N$) at time k (as shown in Fig. 6.1), where $c_{i,k} \in \mathbb{R}^2$ is the planar coordinate of its projection on $\mathbb{O}$, $h_{i,k} \in \mathbb{R}$ is the altitude of the agent above $\mathbb{O}$. Each agent is assumed to have access to its own position at any time. Each cell in the surveillance region is associated with a probability or confidence of target existence within the cell which is modeled using the Bernoulli distribution, i.e. $\theta_{g,k} = 1$ (a target is present) with probability $P_i(\theta_{g,k} = 1)$ and $\theta_{g,k} = 0$ (no target is present) with probability $1 - P_i(\theta_{g,k} = 1)$ for agent i and cell g at time k , where $g \in \mathbb{R}^2$ is the location of the cell center in $\mathbb{O}$. If more than one target are present within a cell, they are treated as one single target.

Figure 6.1: Target search by multiple UAVs.

In this chapter, we mainly discuss about vision-based detections and each agent carries an airborne camera facing downward to the surveillance region (as shown in Fig. 6.1). Each agent independently takes measurements $Z_{i,g,k}$ over the cells within its sensing region $\mathbb{C}_{i,k}$ at time k , where $\mathbb{C}_{i,k} \triangleq \{g \in \mathbb{O} : \|g - c_{i,k}\| \leq h_{i,k} \tan \varphi\}$. We assume that the size of each cell is sufficiently small comparing with the size of $\mathbb{C}_{i,k}$ so that we can ignore the boundary effect and roughly consider a cell as wholly

within $\mathbb{C}_{i,k}$ if its center is within $\mathbb{C}_{i,k}$. Only two observation results are defined for each cell, $Z_{i,g,k} = 0$ or $Z_{i,g,k} = 1$. For all cells, $P(Z_{i,g,k} = 1|\theta_{g,k} = 1) = p_{i,k}$ and $P(Z_{i,g,k} = 1|\theta_{g,k} = 0) = q_{i,k}$ are assumed to be known by agent i as the detection probability and false alarm probability respectively.

6.2 Probability Map Update

6.2.1 Bayesian Update and Consensus-Based Information Fusion

In Chapter 5, we proposed a cooperative control scheme for target search in multi-agent systems. In a group of UAVs, each UAV agent i keeps an individual probability map $\mathcal{P}_{i,g,k}$ of the whole region, where $\mathcal{P}_{i,g,k} \triangleq P_i(\theta_{g,k} = 1)$ and is updated by the Bayesain rule:

$$\begin{aligned} \mathcal{P}_{i,g,k} &= \frac{P(Z_{i,g,k}|\theta_{g,k} = 1) \mathcal{P}_{i,g,k-1}}{P(Z_{i,g,k}|\theta_{g,k} = 1) \mathcal{P}_{i,g,k-1} + P(Z_{i,g,k}|\theta_{g,k} = 0) (1 - \mathcal{P}_{i,g,k-1})} \\ &= \begin{cases} \frac{p_{i,k} \mathcal{P}_{i,g,k-1}}{p_{i,k} \mathcal{P}_{i,g,k-1} + q_{i,k} (1 - \mathcal{P}_{i,g,k-1})}, & \text{if } Z_{i,g,k} = 1; \\ \frac{(1 - p_{i,k}) \mathcal{P}_{i,g,k-1}}{(1 - p_{i,k}) \mathcal{P}_{i,g,k-1} + (1 - q_{i,k}) (1 - \mathcal{P}_{i,g,k-1})}, & \text{if } Z_{i,g,k} = 0; \\ \mathcal{P}_{i,g,k-1}, & \text{otherwise,} \end{cases} \end{aligned} \quad (6.1)$$

where $0 < \mathcal{P}_{i,g,0} < 1$ and $1 > p_{i,k}, q_{i,k} > 0$. For the cases with $p_{i,k} = 0$ or 1 or $q_{i,k} = 0$ or 1 , simplified conclusions can be obtained as shown in Chapter 5 and will not be considered in this chapter. By letting $\mathcal{Q}_{i,g,k} \triangleq \ln\left(\frac{1}{\mathcal{P}_{i,g,k}} - 1\right)$, we get the following transformation of (6.1),

$$\mathcal{Q}_{i,g,k} = \mathcal{Q}_{i,g,k-1} + v_{i,g,k}, \quad (6.2)$$

where

$$v_{i,g,k} \triangleq \begin{cases} \ln \frac{q_{i,k}}{p_{i,k}} & \text{if } Z_{i,g,k} = 1; \\ \ln \frac{1 - q_{i,k}}{1 - p_{i,k}} & \text{if } Z_{i,g,k} = 0; \\ 0 & \text{otherwise.} \end{cases} \quad (6.3)$$

Keeping $\mathcal{Q}_{i,g,k}$ as the updated term instead of $\mathcal{P}_{i,g,k}$ simplifies the nonlinear update in (6.1) into the linear one in (6.2). For a group of UAVs, we let each agent i at time k first take measurements and transmit the measurements to its neighbors. After receiving the measurements from all its neighbors, $\mathcal{Q}_{i,g,k}$ is updated as follows,

$$H_{i,g,k} = \mathcal{Q}_{i,g,k-1} + \sum_{j \in \mathcal{N}_{i,k}} v_{j,g,k}. \quad (6.4)$$

Then, each agent i transmits the updated $\mathcal{Q}_{i,g,k}$ of the whole region to its neighbors for map fusion, which is given by:

$$\mathcal{Q}_{i,g,k} = \sum_{j \in \mathcal{N}_{i,k}} w_{i,j,k} H_{j,g,k}, \quad (6.5)$$

where $w_{i,i,k} = 1 - \frac{d_{i,k} - 1}{N}$, $w_{i,j,k} = \frac{1}{N}$ for $j \in \mathcal{N}_{i,k}$ ($j \neq i$) and $w_{i,j,k} = 0$ for $j \notin \mathcal{N}_{i,k}$.

Then, a matrix composed of $w_{i,j,k}$ can be defined as

$$W_k \triangleq [w_{i,j,k}]_{N \times N} (i, j = 1, \dots, N)$$

which is a doubly stochastic matrix [125].

6.2.2 Time-Varying Detection Probability

In Chapter 5, we only considered a 2-dimensional control scheme assuming that all agents move on a fixed plane parallel to the ground plane. However, in the real world, UAVs such as helicopters can change their altitudes according to their task requirements so as to enlarge their sensing area (here we only consider cameras with

Figure 6.2: Target image in the picture taken by an airborne camera.

a fixed zooming level). Therefore, in this chapter, we will consider the influence of three-dimensional motion model of UAVs on the detection performance.

For vision-based detection, the detection probability relies on the picture resolutions. Fig. 6.2 shows the basic imaging scheme by an airborne camera similar to the one given in [63, 126]. Naturally speaking, the larger the image of a target in the picture (in terms of the number of occupied pixels) obtained by the UAV, the easier for the UAV to discriminate the target no matter what recognition method is used. Hence, we can model the target discriminability of a UAV as a function ρ proportional to the ratio between the size of a target image taken by the camera denoted by S_{TI} and the size of one pixel denoted by S_{P} , i.e.,

$$\rho \propto \frac{S_{\text{TI}}}{S_{\text{P}}}.$$

Assuming that all targets under surveillance have the same topological shape and the size of their projections on the ground plane also maintain the same as S_{T} , we can derive that

$$\rho \propto \frac{S_{\text{TI}}}{S_{\text{T}}} \frac{S_{\text{T}}}{S_{\text{P}}} = \frac{b^2 S_{\text{T}}}{h^2 S_{\text{P}}}, \quad (6.6)$$

where h is the altitude of the UAV and b is the fixed distance between the image and

the lens (as shown in Fig. 6.2). In a multi-agent system, for the i -th agent at time k , we have $\rho_{i,k} \propto \frac{b^2 S_T}{h_{i,k}^2 S_P}$. From (6.6), we may get $\rho_{i,k} \rightarrow \infty$ as $h_{i,k} \rightarrow 0$. However, in reality, $\rho_{i,k}$ cannot be infinitely large. And also for the reason of safety, there should be an upper limit when $h_{i,k}$ is smaller than a threshold $\underline{h}$. That is to say, the target discrimination ability will not be improved any more if a UAV is descending very close to the ground.

The target discriminability determines the detection probability when a UAV is detecting the existence of targets within each cell under surveillance. It is natural to conceive that the detection probability $p_{i,k}$ increases as $\rho_{i,k}$ increases. When the altitude of the UAV becomes larger than a threshold $\bar{h}$, it loses its ability to discriminate any target, which means that the detection result does not rely on the true existence of the target any more. That is, if $h_{i,k} \geq \bar{h}$, we have $P(Z_{i,g,k} = 0 | \theta_g = 1) = P(Z_{i,g,k} = 1 | \theta_g = 1)$, i.e., $p_{i,k} = q_{i,k}$. On the other hand, since the false alarm only happens when no target exists, $q_{i,k}$ is not affected by the target discriminability and only depends on the system itself. Therefore, we assume that $q_{i,k} \equiv q$ is a constant as it is independent of the position of each UAV which has the same hardware installation. Generally, when $h_{i,k} \in [\underline{h}, \bar{h}]$ ($\underline{h} < \bar{h}$), $p_{i,k}$ should be a monotonically increasing function of $\rho_{i,k}$, or more explicitly, a monotonically decreasing function of $h_{i,k}$. Therefore, we assume the following detection probability model:

$$p_{i,k} = \begin{cases} q, & \text{if } h_{i,k} \geq \bar{h}; \\ f(h_{i,k}), & \text{if } \underline{h} < h_{i,k} < \bar{h}; \\ \check{p}, & \text{if } 0 < h_{i,k} \leq \underline{h}, \end{cases} \quad (6.7)$$

where $f'(h_{i,k}) < 0$ for $h_{i,k} \in (\underline{h}, \bar{h})$, $f(\underline{h}) = \check{p}$ and $f(\bar{h}) = \check{q}$. Recalling the discussions in the previous part, we assume $1 > \check{p} > q > 0$. In this chapter, the altitude $h_{i,k}$ of an agent is allowed to vary from 0 to ∞ for theoretic analysis, though it may not happen in the real world due to system limitations.

Remark 6.1. *Model (6.7) is motivated by the natural understanding of the interaction between the altitude of an agent and its detection and false alarm probabilities.*

It only reflects the general relation between those parameters, and is not restricted to a specific parametric representation of function $f(h_{i,k})$. Hence, our method is applicable for any detection probability function that fits for the model. An experimental detection probability model of CCD camera has been given in [126].

Denoting by $m_{i,g,k}$ the number of observations taken over cell g up to time k by agent i and defining $m_{g,k} \triangleq \sum_{i=1}^N m_{i,g,k}$, the following lemma has been given in Chapter 5 which shows the almost sure (a.s.) convergence of $\frac{\mathcal{Q}_{i,g,k}}{m_{g,k}}$, under the condition that the detection probability $p_{i,k} \equiv p$ and the false alarm probability $q_{i,k} \equiv q$ are constant for all agents, the targets are static and the measurements are not exchanged except the information of $\mathcal{Q}_{i,g,k}$.

Lemma 6.1. *Given the initial prior probability map $0 < \mathcal{P}_{i,g,0} < 1$ for each agent i , if $q \neq p$ and $0 < p, q < 1$, and the network topology $\mathcal{G}_k$ is connected at all times, the following conclusions hold by implementing the map updating rule (6.4) and (6.5).*

1. If any target is present within cell g , $\mathcal{Q}_{i,g,k} \xrightarrow{a.s.} -\infty$ (i.e., $\mathcal{P}_{i,g,k} \xrightarrow{a.s.} 1$) and $\frac{\mathcal{Q}_{i,g,k}}{m_{g,k}} \xrightarrow{a.s.} \frac{p}{N} \ln \frac{q}{p} + \frac{1-p}{N} \ln \frac{1-q}{1-p}$ for all agents as $m_{g,k} \rightarrow +\infty$.
2. If no target is present within cell g , $\mathcal{Q}_{i,g,k} \xrightarrow{a.s.} +\infty$ (i.e., $\mathcal{P}_{i,g,k} \xrightarrow{a.s.} 0$) and $\frac{\mathcal{Q}_{i,g,k}}{m_{g,k}} \xrightarrow{a.s.} \frac{q}{N} \ln \frac{q}{p} + \frac{1-q}{N} \ln \frac{1-q}{1-p}$ for all agents as $m_{g,k} \rightarrow +\infty$.

Now we generalize the above conclusions into the case of time-varying detection probabilities with static targets.

Theorem 6.1. *Given the initial prior probability map $0 < \mathcal{P}_{i,g,0} < 1$ for each agent i , if there exist constants $\check{p}$ and $\hat{p}$ such that the detection probability for each agent i is subject to $1 > \check{p} \geq p_{i,k} \geq \hat{p} > q > 0$ and the network topology $\mathcal{G}_k$ is connected for all k , the following conclusions hold by implementing the map updating rules (6.4) and (6.5).*

1. If a target is present within cell g , $\mathcal{Q}_{i,g,k} \xrightarrow{a.s.} -\infty$ (i.e., $\mathcal{P}_{i,g,k} \xrightarrow{a.s.} 1$) for all agents as $m_{g,k} \rightarrow +\infty$.

2. If no target is present within cell g , $\mathcal{Q}_{i,g,k} \xrightarrow{a.s.} +\infty$ (i.e., $\mathcal{P}_{i,g,k} \xrightarrow{a.s.} 0$) for all agents as $m_{g,k} \rightarrow +\infty$.

Proof. First, we consider Case 1, where a target is present within cell g . Define the following augmented notations for the entire system:

$$\begin{aligned}\Upsilon_{g,k} &\triangleq [\mathcal{Q}_{1,g,k}, \mathcal{Q}_{2,g,k}, \dots, \mathcal{Q}_{N,g,k}]^T, \\ \Phi_{g,k} &\triangleq [v_{1,g,k}, v_{2,g,k}, \dots, v_{N,g,k}]^T.\end{aligned}$$

Then, the updating rule (6.4) and (6.5) can be replaced by the following equation:

$$\Upsilon_{g,k} = W_k (\Upsilon_{g,k-1} + \Phi_{g,k}) = \prod_{t=1}^k W_t \Upsilon_{g,0} + \sum_{l=1}^k \prod_{t=l}^k W_t \Phi_{g,l}. \quad (6.8)$$

Hence,

$$\begin{aligned}\mathbb{E} \left[\sum_{i=1}^N \mathcal{Q}_{i,g,k} \right] &= \mathbb{E} [\mathbf{1}^T \Upsilon_{g,k}] = \mathbb{E} [\mathbf{1}^T \Upsilon_{g,0}] + \sum_{t=1}^k \mathbb{E} [\mathbf{1}^T \Phi_{g,t}] \\ &= \sum_{i=1}^N \mathbb{E} [\mathcal{Q}_{i,g,0}] + \sum_{t=1}^k \sum_{i=1}^N \sum_{j \in \mathcal{N}_{i,t}} \mathbb{E} [v_{j,g,t}],\end{aligned}$$

where

$$\mathbb{E} [v_{j,g,t}] = \left[p_{j,t} \ln \frac{q}{p_{j,t}} + (1 - p_{j,t}) \ln \frac{1-q}{1-p_{j,t}} \right] \mathbb{1}_{\{g \in \mathbb{C}_{j,t}\}},$$

and $\mathbb{1}_{\{g \in \mathbb{C}_{j,t}\}}$ is the indicator function defined as

$$\mathbb{1}_{\{g \in \mathbb{C}_{j,t}\}} = \begin{cases} 1, & \text{if } g \in \mathbb{C}_{j,t}; \\ 0, & \text{otherwise.} \end{cases}$$

Since $\mathcal{Q}_{i,g,0}$, v_{j_1,g,η_1} and v_{j_2,g,η_2} are independent for $j_1 \neq j_2$ or $\eta_1 \neq \eta_2$, we can get the variance

$$\mathbb{D} \left[\sum_{i=1}^N \mathcal{Q}_{i,g,k} \right] = \sum_{i=1}^N \mathbb{D} [\mathcal{Q}_{i,g,0}] + \sum_{t=1}^k \sum_{i=1}^N \sum_{j \in \mathcal{N}_{i,t}} \mathbb{D} [v_{i,g,l}],$$

where

$$D[v_{j,g,t}] = p_{j,t} (1 - p_{j,t}) \left(\ln \frac{q}{p_{j,t}} - \ln \frac{1-q}{1-p_{j,t}} \right)^2 \mathbb{1}_{\{g \in \mathbb{C}_{j,t}\}}.$$

Considering that $D[v_{j,g,t}]$ is a continuous function of $p_{j,t}$ and $\check{p} \geq p_{j,t} \geq \hat{p} > q$, there exists a constant real number σ such that $D[v_{j,g,t}] \leq \sigma^2$ for $g \in \mathbb{C}_{j,t}$, which implies

$$\begin{aligned} \lim_{m_{i,g,k} \rightarrow +\infty} \sum_{t=1}^k \sum_{i=1}^N \sum_{j \in \mathcal{N}_{i,t}} \frac{D[v_{j,g,t}]}{m_{g,t}^2} &\leq \lim_{m_{i,g,k} \rightarrow +\infty} \sum_{t=1}^k \sum_{i=1}^N \sum_{j=1}^N \frac{\sigma^2 \mathbb{1}_{\{g \in \mathbb{C}_{j,t}\}}}{m_{g,t}^2} \\ &= \lim_{m_{i,g,k} \rightarrow +\infty} \sum_{t=1}^k \frac{N \sigma^2 (m_{g,t} - m_{g,t-1})}{m_{g,t}^2} \\ &= \lim_{m_{g,k} \rightarrow +\infty} \sum_{l=1}^{m_{g,k}} \frac{N^2 \sigma^2}{l^2} < \infty. \end{aligned}$$

According to the Kolmogorov Strong Law of Large Numbers, [127] we get

$$\frac{\sum_{t=1}^k \sum_{i=1}^N \sum_{j \in \mathcal{N}_{i,t}} E[v_{j,g,t}]}{m_{g,t}} \xrightarrow{a.s.} 0, \quad \text{as } m_{g,t} \rightarrow +\infty.$$

Hence,

$$\begin{aligned} \frac{\sum_{i=1}^N \mathcal{Q}_{i,g,k}}{m_{g,k}} &= \frac{\sum_{i=1}^N \mathcal{Q}_{i,g,0}}{m_{g,k}} + \frac{\sum_{t=1}^k \sum_{i=1}^N \sum_{j \in \mathcal{N}_{i,t}} (v_{j,g,t} - E[v_{j,g,t}])}{m_{g,t}} \\ &\quad + \frac{\sum_{t=1}^k \sum_{i=1}^N \sum_{j \in \mathcal{N}_{i,t}} E[v_{j,g,t}]}{m_{g,t}} \\ &\xrightarrow{a.s.} \frac{\sum_{t=1}^k \sum_{i=1}^N \sum_{j \in \mathcal{N}_{i,t}} E[v_{j,g,t}]}{m_{g,t}}, \quad \text{as } m_{g,t} \rightarrow +\infty. \end{aligned}$$

On the other hand, following Lemma 6.1 it is straightforward to get

$$E[v_{j,g,t}] \leq \hat{p} \ln \frac{q}{\hat{p}} + (1 - \hat{p}) \ln \frac{1-q}{1-\hat{p}} < 0.$$

Hence,

$$\limsup_{m_{i,g,k} \rightarrow +\infty} \frac{\sum_{i=1}^N \mathcal{Q}_{i,g,k}}{m_{g,k}} \leq \limsup_{m_{i,g,k} \rightarrow +\infty} \frac{\sum_{t=1}^k \sum_{i=1}^N E[v_{i,g,t}]}{m_{g,t}} = \hat{p} \ln \frac{q}{\hat{p}} + (1 - \hat{p}) \ln \frac{1-q}{1-\hat{p}} < 0$$

which implies $\sum_{i=1}^N \mathcal{Q}_{i,g,k} \xrightarrow{a.s.} -\infty$. Since the network is connected all the time, $\mathcal{Q}_{i,g,k}$ for each agent i will almost surely converge to $-\infty$ by implementing the average consensus protocol (6.5) (as shown in [117, 128]).

Following the same procedure of the proof above, we can prove the conclusion of Case 2. $\square$

Remark 6.2. According to Theorem 6.1, the detection probability $p_{i,k}$ for each agent i has to be bounded within a constant range $[\hat{p}, \check{p}] \subset (q, 1)$ to guarantee the convergence of $\mathcal{Q}_{i,g,k}$. In (6.7), $p_{i,k} \leq \check{p} < 1$ has already satisfied the constraint of upper bound. Then, we only need to control the altitudes of the UAVs to make the constraint of lower bound satisfied, i.e., $p_{i,k} \geq \hat{p} > q$, which will be discussed in the next section.

Remark 6.3. In the map fusion step (6.5), each agent is required to know the total number of agents N . However, such global information sometimes is hard to obtain accurately. For example, due to the need of some special task, existing UAVs may be withdrawn from the group or new agents are added into the group, which makes the total number of agents vary from time to time. In this case, we can let $w_{i,i,k} = 1 - \frac{d_{i,k} - 1}{\bar{N}}$, $w_{i,j,k} = \frac{1}{\bar{N}}$ for $j \in \mathcal{N}_{i,k}$ ($j \neq i$) and $w_{i,j,k} = 0$ for $j \notin \mathcal{N}_{i,k}$, where $\bar{N}$ is an upper bound of the total number of agents in all possible cases and assumed to be known beforehand. The same consensus and convergence results can be obtained according to the analysis on consensus protocols in [6, 38, 117].

6.2.3 Environment-Based Probability Map

In the map updates (6.4) and (6.5), the effect of the environmental changes has not been considered, such as the information decaying and transmission between cells. For example, if targets randomly appear or disappear during search, the historical information about the target existence cannot reflect the true current situation and revisits of certain frequency to the detected cells are needed for information update. This problem can be formulated as the information decaying for each cell. If a target

moves from one cell to another, then part of the information for the former cell should be removed and counted as the new information for the latter cell. This problem can be formulated as the information transmission between each two cells. Therefore, we need to generalize the aforementioned map updating model to be applied to the case with such environmental changes. Similar to the assumption made in [62], we assume that $\mathcal{Q}_{i,g,k}$ decays exponentially for each cell if no information is received. The information transmission between cells due to target movement is modeled based on the transition of probabilities. In addition, the prior knowledge about the environmental change is taken as the system input. All these lead to the following generalized updating model for $\mathcal{Q}_{i,g,k}$:

$$\begin{aligned} H_{i,g,k} &= e^{-\alpha T} \sum_{r \in \mathbb{O}} a_{i,g,r,k} b_{i,g,r,k} \mathcal{Q}_{i,r,k-1} + \sum_{j \in \mathcal{N}_{i,k}} v_{j,g,k} + \xi_{i,g,k}, \\ \mathcal{Q}_{i,g,k} &= \sum_{j \in \mathcal{N}_{i,k}} w_{i,j,k} H_{j,g,k}, \end{aligned} \quad (6.9)$$

where $\alpha \geq 0$ is the information decaying factor, T is the sampling period of all UAVs, $a_{i,g,r,k}$ and $b_{i,g,r,k}$ are the information transmission factors which are nonnegative, and $\xi_{i,g,k}$ is the input information vector given by the prior knowledge about the target existence within cell g . Specifically, $b_{i,g,r,k}$ satisfies $b_{i,g,g,k} = 1$ and $b_{i,g,r,k} = 0$ ($g \neq r$) for $\mathcal{Q}_{i,r,k-1} > 0$, and $b_{i,g,r,k} = P(\theta_{g,k} = 1 | \theta_{r,k-1} = 1)$ for $\mathcal{Q}_{i,r,k-1} \leq 0$. $a_{i,g,r,k}$ is determined by $a_{i,g,\hat{r}_i,k} = 1$ and $a_{i,g,r,k} = 0$ ($r \neq \hat{r}_i$), where

$$\begin{aligned} \hat{r}_i &= \arg \min_{r \in \mathcal{B}_{i,g,k}} b_{i,g,r,k} \mathcal{Q}_{i,r,k-1}, \\ \mathcal{B}_{i,g,k} &= \{r \in \mathbb{O} : b_{i,g,r,k} > 0\}. \end{aligned} \quad (6.10)$$

Remark 6.4. $a_{i,g,r,k}$ and $b_{i,g,r,k}$ are defined based on the physical meaning of information transmission due to the target movement in the real world. Since the combination of $\mathcal{Q}_{i,r,k}$ ($r \in \mathbb{O}$) into a cell g involves the combination of historical measurement information of all cells $r \in \mathbb{O}$, the correlation of which may not be known, we need to be careful in dealing with the fusion of such information. If $\mathcal{Q}_{i,g,k} > 0$, we are more confident that no target exists within cell g . Otherwise, we

are more confident that a target exists within cell g . Since information transmission out of a cell at time k is expected to occur only when a target exists within the cell at time $k-1$, we let $b_{i,g,r,k} = 0$ if $\mathcal{Q}_{i,r,k-1} > 0$, which means there is no transmission of information (or target movement) from cell r to cell g . If $\mathcal{Q}_{i,r,k-1} \leq 0$, information transmission occurs from cell r to cell g due to the possible target movement from r to g and the amount of information transmitted should be proportional to $P(\theta_{g,k} = 1 | \theta_{r,k-1} = 1)$, i.e., equal to $b_{i,g,r,k} \mathcal{Q}_{i,r,k-1}$. The smaller the $b_{g,g,k}$ is, the less amount of information is retained for cell g . Furthermore, by assuming that within one cell there can only exist up to one target at a time, i.e., at most one target can move into a cell at a time, we select the information stream with the largest transmitted amount as the newly stored information for cell g when there are incoming information streams from multiple cells $r \in \mathcal{B}_{g,k}$. That is, to take the smallest $b_{i,g,r,k} \mathcal{Q}_{i,r,k-1}$ subject to $\mathcal{Q}_{i,r,k-1} \leq 0$ as the newly stored information after the transmission, which corresponds to the most probable target movement to g in all possible movements to g from different cells. The information decaying factor α is set to be positive in the case that the prior knowledge of $b_{i,g,r,k}$ is not accurate or targets may appear and disappear unpredictably during the search. In this case, the information decaying makes the agents revisit the detected regions at a certain frequency.

Defining the following augmented variables

$$\begin{aligned} \mathbf{Q}_{i,k} &\triangleq [\mathcal{Q}_{i,g_1,k}, \dots, \mathcal{Q}_{i,g_M,k}]^T, & \mathbf{Q}_k &\triangleq [\mathbf{Q}_{1,k}^T, \dots, \mathbf{Q}_{N,k}^T]^T, \\ \mathbf{V}_{i,k} &\triangleq \left[\sum_{j \in \mathcal{N}_{i,k}} v_{j,g_1,k}, \dots, \sum_{j \in \mathcal{N}_{i,k}} v_{j,g_M,k} \right]^T, & \mathbf{V}_k &\triangleq [\mathbf{V}_{1,k}^T, \dots, \mathbf{V}_{N,k}^T]^T, \\ \boldsymbol{\xi}_{i,k} &\triangleq [\xi_{i,g_1,k}, \dots, \xi_{i,g_M,k}]^T, & \boldsymbol{\xi}_k &\triangleq [\boldsymbol{\xi}_{1,k}^T, \dots, \boldsymbol{\xi}_{N,k}^T]^T, \\ A_{i,k} &\triangleq [a_{i,g_\tau,g_s,k} b_{i,g_\tau,g_s,k}]_{M \times M} (\tau, s = 1, \dots, M), & A_k &\triangleq \text{diag}[A_{1,k}, \dots, A_{N,k}], \end{aligned}$$

where τ and s are respectively the row and column indices of an appropriate cell in $A_{i,k}$, and M is the total number of cells, we get the following generalized updating model:

$$\mathbf{Q}_k = e^{-\alpha T} (W_k \otimes I) A_k \mathbf{Q}_{k-1} + (W_k \otimes I) (\mathbf{V}_k + \boldsymbol{\xi}_k), \quad (6.11)$$

where $\otimes$ denotes the Kronecker product.

Proposition 6.1. *If $\alpha > 0$ and $\mathbf{V}_k = 0$, $\boldsymbol{\xi}_k = 0$ for all k , then $\lim_{k \rightarrow +\infty} \mathbf{Q}_k = 0$.*

Proof. From (6.11), we have

$$\mathbf{Q}_k = e^{-\alpha k T} \left(\prod_{t=1}^k (W_t \otimes I) A_t \right) \mathbf{Q}_0.$$

From the definitions, A_t is a column stochastic matrix and W_t is a doubly stochastic matrix, which implies that $\prod_{t=1}^k (W_t \otimes I) A_t$ is still a column stochastic matrix and its spectrum radius is 1 according to [125]. Therefore,

$$\lim_{k \rightarrow +\infty} e^{-\alpha k T} \prod_{t=1}^k (W_t \otimes I) A_t = 0$$

which implies $\lim_{k \rightarrow +\infty} \mathbf{Q}_k = 0$. □

Proposition 6.2. *If $\alpha > 0$ and the input information is bounded by $\|\boldsymbol{\xi}_k\| < \bar{\xi}$ for all k , then $\|\mathbf{Q}_k\|$ is also bounded.*

Proof. Following (6.11), we have

$$\begin{aligned} \mathbf{Q}_k &= e^{-\alpha k T} \left(\prod_{t=1}^k (W_t \otimes I) A_t \right) \mathbf{Q}_0 \\ &\quad + \sum_{l=1}^{k-1} e^{-\alpha(k-l)T} \left(\prod_{t=l+1}^k (W_t \otimes I) A_t \right) (W_l \otimes I) (\mathbf{V}_l + \boldsymbol{\xi}_l). \end{aligned}$$

Since $(W_t \otimes I) A_t$ and $\left(\prod_{t=l+1}^k (W_t \otimes I) A_t \right) (W_l \otimes I)$ are both column stochastic matrices, we can find a number $\varpi > 0$ such that $\left\| \prod_{t=1}^k (A_t \otimes W_t) \right\| \leq \varpi$ and

$\left\| \prod_{t=l+1}^k (A_t \otimes W_t) (I \otimes W_l) \right\| \leq \varpi$ for any k . Therefore,

$$\begin{aligned} \|\mathbf{Q}_k\| &\leq e^{-\alpha k T} \varpi \|\mathbf{Q}_0\| + \sum_{l=1}^{k-1} e^{-\alpha(k-l)T} \varpi (\|\mathbf{V}_l\| + \|\boldsymbol{\xi}_l\|) \\ &< \varpi \|\mathbf{Q}_0\| + \sum_{l=1}^{k-1} e^{-\alpha(k-l)T} \varpi (\|\mathbf{V}_l\| + \|\boldsymbol{\xi}_l\|). \end{aligned}$$

It is known that $\|\mathbf{V}_l\|$ and $\|\boldsymbol{\xi}_l\|$ are bounded for all l , then we have

$$\|\mathbf{Q}_k\| < \varpi \|\mathbf{Q}_0\| + \lim_{k \rightarrow +\infty} \sum_{l=1}^{k-1} e^{-\alpha(k-l)T} \varpi (\bar{V} + \bar{\xi}) = \frac{1}{1 - e^{-\alpha T}} \varpi (\bar{V} + \bar{\xi}), \quad (6.12)$$

where $\bar{V}$ is a number satisfying $\bar{V} \geq \|\mathbf{V}_l\|$ for all l . □

Remark 6.5. *Proposition 6.1 illustrates that with no information input, $\mathbf{Q}_k$ tends to 0. This corresponds to that $\mathcal{P}_{i,g,k}$ approaches 0.5 for all cells and agents, which means no preference is made between the existence and nonexistence of a target within the cell. Thus, $\|\mathbf{Q}_k\|$ can be seen as the gathered information for decision making on the target existence and the larger the $\|\mathbf{Q}_k\|$, the higher the confidence on the target existence or nonexistence. Proposition 6.2 illustrates that the gathered information $\|\mathbf{Q}_k\|$ will not be accumulated to infinity when the information decaying is present no matter how many measurements are received. Hence, our aim of controlling the UAVs is to maximize $\|\mathbf{Q}_k\|$ in some sense, which will be discussed in the following section.*

6.3 Cooperative Coverage Control

We consider the first-order discrete-time motion model for the UAVs:

$$\mu_{i,k} = \mu_{i,k-1} + u_{i,k},$$

where $u_{i,k} \in \mathbb{R}^3$ is the control input of the i -th agent at time k . Following (6.11), we can get

$$\mathbf{Q}_k = \mathbf{G}_k + (W_k \otimes I) \mathbf{V}_k,$$

where $\mathbf{G}_k \triangleq e^{-\alpha T} (W_k \otimes I) A_k \mathbf{Q}_{k-1} + (W_k \otimes I) \boldsymbol{\xi}_k$. At time $k-1$, $\mathbf{G}_k$ can be seen as the prior information, and $\mathbf{V}_k$ the information to be gathered from measurements. Since $\mathbf{V}_k$ and $E[\mathbf{V}_k]$ are both related to the true target existence which is unknown, we can only estimate the values of $\mathbf{Q}_k$ or $E[\mathbf{Q}_k]$ before taking measurements. What we can do at time $k-1$ is to find the optimal next time sampling position $\mu_{i,k}$ so as to maximize the information to be gathered at time k . More precisely, the problem can be formulated as the optimization problem:

$$\max_{\boldsymbol{\mu}_k} E \left[\|\mathbf{Q}_k - \mathbf{G}_k\|^2 \mid \mathbf{Q}_{k-1}, \boldsymbol{\xi}_k \right] = E \left[\|(W_k \otimes I) \mathbf{V}_k\|^2 \right], \quad (6.13)$$

where $\boldsymbol{\mu}_k \triangleq [\mu_{1,k}^T, \dots, \mu_{N,k}^T]^T$. Considering that W_k includes the global topological information which is often hard to obtain for each individual agent in a distributed system, and $\|(W_k \otimes I) \mathbf{V}_k\| \leq \|\mathbf{V}_k\|$, we replace (6.13) with the following suboptimal optimization,

$$\max_{\boldsymbol{\mu}_k} E \left[\|\mathbf{V}_k\|^2 \right] = \sum_{g \in \mathbb{O}} \sum_{i=1}^N \sum_{j \in \mathcal{N}_{i,k}} E \left[v_{j,g,k}^2 \right] \mathbb{1}_{\{g \in \mathbb{C}_{j,k}\}}. \quad (6.14)$$

Following (6.14), we should try to maximize $E[v_{j,g,k}^2]$ and the collective sensing area of all agents. From (6.3), we get for $g \in \mathbb{C}_{j,k}$,

$$E[v_{j,g,k}^2] = \begin{cases} \left\| p_{j,k} \ln \frac{q}{p_{j,k}} + (1 - p_{j,k}) \ln \frac{1-q}{1-p_{j,k}} \right\|^2, & \text{if } \theta_{g,k} = 1; \\ \left\| q \ln \frac{q}{p_{j,k}} + (1 - q) \ln \frac{1-q}{1-p_{j,k}} \right\|^2, & \text{otherwise.} \end{cases}$$

It is straightforward to find that $E[v_{j,g,k}^2]$ is a monotonically increasing function of $p_{j,k}$ no matter $\theta_{g,k} = 1$ or not. Thus, (6.14) is further replaced with the following

optimization problem,

$$\max_{\boldsymbol{\mu}_k} \mathcal{H}(\boldsymbol{\mu}_k) = \sum_{i=1}^N \int_{\mathbb{M}_{i,k}} \phi_k(r) (p_{i,k} - q) \mathbb{1}_{\{r \in \mathbb{C}_{i,k}\}} dr, \quad (6.15)$$

where $\phi_k(r)$ is a given nonnegative weighting function of $r \in \mathbb{O}$ at time k , and its influence on the control law will be shown later. $\{\mathbb{M}_{1,k}, \dots, \mathbb{M}_{N,k}\}$ is a partition of $\mathbb{O}$ at time k , where $\mu_{i,k} \in \mathbb{M}_{i,k}$ and $\mathbb{M}_{i,k} \cap \mathbb{M}_{j,k} = \emptyset$ for $i \neq j$. The introduction of the partition is for avoidance of collision of UAVs and ease of dealing with the overlapped sensing regions between neighboring agents. Since $p_{i,k} \geq q$, $\mathcal{H}(\boldsymbol{\mu}_k)$ is always nonnegative. Denoting by $\partial(\bullet)$ the boundary of the corresponding region and $n_{\partial(\bullet)}(r)$ the outward pointing normal vector of the boundary $\partial(\bullet)$ at point r , we can compute the gradient of $\mathcal{H}(\boldsymbol{\mu}_k)$ as follows.

Theorem 6.2. *The gradient of the cost function $\mathcal{H}(\boldsymbol{\mu}_k)$ with respect to $\mu_{i,k}$ is given by*

$$\begin{aligned} \frac{\partial \mathcal{H}(\boldsymbol{\mu}_k)}{\partial c_{i,k}} &= (p_{i,k} - q) \int_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} \phi_k(r) n_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})}(r) dr, \\ \frac{\partial \mathcal{H}(\boldsymbol{\mu}_k)}{\partial h_{i,k}} &= f'(h_{i,k}) \int_{\mathbb{M}_{i,k} \cap \mathbb{C}_{i,k}} \phi_k(r) dr + (p_{i,k} - q) \tan \varphi \int_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} \phi_k(r) dr, \end{aligned} \quad (6.16)$$

for $h_{i,k} \in (\underline{h}, \bar{h})$ and

$$\begin{aligned} \frac{\partial \mathcal{H}(\boldsymbol{\mu}_k)}{\partial c_{i,k}} &= (\check{p} - q) \int_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} \phi_k(r) n_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})}(r) dr, \\ \frac{\partial \mathcal{H}(\boldsymbol{\mu}_k)}{\partial h_{i,k}} &= (\check{p} - q) \tan \varphi \int_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} \phi_k(r) dr, \end{aligned} \quad (6.17)$$

for $h_{i,k} \in (0, \underline{h})$, where $c_{i,0} \in \mathbb{O}$ and $h_{i,0} \in (0, \bar{h})$.

Proof. We first consider $h_{i,k} \in (\underline{h}, \bar{h})$. From (6.15), we get

$$\mathcal{H}(\mu_k) = \sum_{i=1}^N \int_{\mathbb{M}_{i,k}} \phi_k(r) (p_{i,k} - q) \mathbb{1}_{\{r \in \mathbb{C}_{i,k}\}} dr = \sum_{i=1}^N \int_{\mathbb{M}_{i,k} \cap \mathbb{C}_{i,k}} \phi_k(r) (p_{i,k} - q) dr.$$

Defining a set of agents for agent i , $\hat{\mathcal{N}}_{i,k} \triangleq \{j : \partial(\mathbb{M}_{j,k}) \cap \mathbb{C}_{i,k} \neq \emptyset\}$, it follows that

$$\begin{aligned} & \frac{\partial \mathcal{H}(\mu_k)}{\partial \mu_{i,k}} \\ &= \int_{\mathbb{M}_{i,k} \cap \mathbb{C}_{i,k}} \phi_k(r) \frac{\partial p_{i,k}}{\partial \mu_{i,k}} dr + \int_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} \phi_k(r) (p_{i,k} - q) \frac{\partial r}{\partial \mu_{i,k}} n_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})}(r) dr \\ &+ \sum_{j \in \hat{\mathcal{N}}_{i,k}} \int_{(\partial(\mathbb{M}_{j,k}) \setminus \partial(\mathbb{O})) \cap \mathbb{C}_{i,k}} \phi_k(r) (p_{i,k} - q) \frac{\partial r}{\partial \mu_{i,k}} n_{\partial(\mathbb{M}_{j,k}) \setminus \partial(\mathbb{O}) \cap \mathbb{C}_{i,k}}(r) dr \\ &+ \sum_{j \in \hat{\mathcal{N}}_{i,k}} \int_{(\partial(\mathbb{M}_{j,k}) \cap \partial(\mathbb{O})) \cap \mathbb{C}_{i,k}} \phi_k(r) (p_{i,k} - q) \frac{\partial r}{\partial \mu_{i,k}} n_{\partial(\mathbb{M}_{j,k}) \cap \partial(\mathbb{O}) \cap \mathbb{C}_{i,k}}(r) dr. \end{aligned}$$

For $r \in [(\partial(\mathbb{M}_{j_1,k}) \setminus \partial(\mathbb{O})) \cap \mathbb{C}_{i,k}] \cap [(\partial(\mathbb{M}_{j_2,k}) \setminus \partial(\mathbb{O})) \cap \mathbb{C}_{i,k}]$ ($j_1, j_2 \in \hat{\mathcal{N}}_{i,k}$ and $j_1 \neq j_2$), we have $n_{(\partial(\mathbb{M}_{j_1,k}) \setminus \partial(\mathbb{O})) \cap \mathbb{C}_{i,k}}(r) = -n_{(\partial(\mathbb{M}_{j_2,k}) \setminus \partial(\mathbb{O})) \cap \mathbb{C}_{i,k}}(r)$. For $r \in (\partial(\mathbb{M}_{j,k}) \cap \partial(\mathbb{O})) \cap \mathbb{C}_{i,k}$ ($j \in \hat{\mathcal{N}}_{i,k}$), we have $\frac{\partial r}{\partial \mu_{i,k}} = 0$. Hence,

$$\begin{aligned} \frac{\partial \mathcal{H}(\mu_k)}{\partial \mu_{i,k}} &= \int_{\mathbb{M}_{i,k} \cap \mathbb{C}_{i,k}} \phi_k(r) \frac{\partial p_{i,k}}{\partial \mu_{i,k}} dr \\ &+ \int_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} \phi_k(r) (p_{i,k} - q) \frac{\partial r}{\partial \mu_{i,k}} n_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})}(r) dr. \end{aligned}$$

From (6.7) and according to the results of [63], we get

$$\frac{\partial p_{i,k}}{\partial c_{i,k}} = 0, \quad \frac{\partial p_{i,k}}{\partial h_{i,k}} = f'(h_{i,k}),$$

and

$$\frac{\partial r}{\partial c_{i,k}} n_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})}(r) = n_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})}(r), \quad \frac{\partial r}{\partial h_{i,k}} n_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})}(r) = \tan \varphi,$$

where φ is half of the angle width of the field of view of each agent (as shown in

Fig. 6.2). Therefore, (6.16) holds.

According to (6.7), $p_{i,k} = \check{p}$ for $h_{i,k} \in (0, \underline{h})$. It is straightforward to get (6.17) by the same derivation above.

□

Following Theorem 6.2, a gradient-based control law is given by

$$u_{i,k} = K_u \frac{\partial \mathcal{H}(\boldsymbol{\mu}_k)}{\partial \mu_{i,k}}, \quad (6.18)$$

where K_u is a positive gain parameter. Note that the control input is always upper bounded in real systems, i.e., $\|u_{i,k}\| \leq u_{max}$ for some positive number u_{max} .

Remark 6.6. Following (6.16), $\frac{\partial \mathcal{H}(\boldsymbol{\mu}_k)}{\partial h_{i,k}} \geq 0$ implies that

$$p_{i,k} \geq q + \frac{\kappa \lambda h_{i,k}^{-\lambda-1} \int_{\mathbb{M}_{i,k} \cap \mathbb{C}_{i,k}} \phi_k(r) dr}{\tan \varphi \int_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} \phi_k(r) dr} \geq q + \frac{\kappa \lambda \bar{h}^{-\lambda-1} \int_{\mathbb{M}_{i,k} \cap \mathbb{C}_{i,k}} \phi_k(r) dr}{\tan \varphi \int_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} \phi_k(r) dr} \triangleq \hat{q} > q.$$

Hence, by control law (6.18), $p_{i,k}$ is always no less than $\hat{p}$ with $h_{i,0} \in (\underline{h}, \bar{h})$. Recalling the comments of Remark 6.2, the condition on $p_{i,k}$ in Theorem 6.1 is always satisfied. On the other hand, since $\check{p} - q > 0$ implies $\frac{\partial \mathcal{H}(\boldsymbol{\mu}_k)}{\partial h_{i,k}} > 0$ according to (6.17), agents will never crash on the ground by the control law (6.18) when their altitudes are less than $\underline{h}$. Hence, with $h_{i,0} \in (0, \bar{h})$, it can be guaranteed that $h_{i,k} \in (0, \bar{h})$ for all $k > 0$.

Remark 6.7. In the real system implementations, the integrals in (6.16) and (6.17) are approximated by the following discretized summations over cells:

$$\begin{aligned} \int_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} \phi_k(r) n_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})}(r) dr &= \sum_{g \in \mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} p_{i,k} \phi_{i,k}(g) \varsigma \frac{g - \mu_{i,k}}{\|g - \mu_{i,k}\|}, \\ \int_{\mathbb{M}_{i,k} \cap \mathbb{C}_{i,k}} \phi_k(r) dr &= \sum_{g \in \mathbb{M}_{i,k} \cap \mathbb{C}_{i,k}} \phi_{i,k}(g) \varsigma^2, \\ \int_{\mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} \phi_k(r) dr &= \sum_{g \in \mathbb{M}_{i,k} \cap \partial(\mathbb{C}_{i,k})} \tan \varphi \phi_{i,k}(g) \varsigma, \end{aligned}$$

where for some region $\mathbb{G} \in \mathbb{R}^2$, $g \in \mathbb{G}$ means the center of cell g is within $\mathbb{G}$, and $g \in \partial(\mathbb{G})$ means the boundary $\partial(\mathbb{G})$ crosses over g , ς and ς^2 are the edge length and the area of a cell respectively.

Generally, it is favored that UAVs stay longer in a region with less gathered information to take more measurements. Thus, we define the weighting function $\phi_{i,k}(g)$ as a function of the gathered information $\|\mathcal{Q}_{i,g,k-1}\|$ for each grid, i.e.,

$$\phi_{i,k}(g) = e^{-K_\phi \|\mathcal{Q}_{i,g,k-1}\|}, \quad (6.19)$$

where K_ϕ is a positive gain parameter. By this model, the cells with less gathered information are given higher weights for detection.

Remark 6.8. *The partition $\{\mathbb{M}_{1,k}, \dots, \mathbb{M}_{N,k}\}$ can be static or time-varying. Partition is commonly used in the real world where each UAV only takes charge of one part of the whole surveillance region so that the whole searching task is shared by multiple agents. Users can predefine the task regions for each UAV or let the UAVs dynamically compute the partition following some rules. An example of the dynamic partition is the Voronoi partition which has been widely used in the distributed control [11]. It should be noted that the computation of the gradient in (6.16) relies on the known partition $\mathbb{M}_{i,k}$ for agent i . However, dynamic partitions usually require the positions of all related agents to be known by agent i , which is hard to achieve with limited communication range. Such effect should be taken into consideration when a specific partition is used. For the Voronoi partition with a limited communication range, the control law derived above is still applicable but may not be optimal. It works in the sense that each agent computes its control law based on a Voronoi partition determined only by using the positions of its neighbors while ignoring all the other agents. In this way, the collision between agents can still be avoided when they move into within the communication range. In addition to the coverage control, communication topology control should also be considered to maintain network connectivity. There has been work on communication control in distributed networks*

such as [124, 128].

Remark 6.9. To avoid the case that agents stop at local optimum solutions, we can give a perturbation to an agent if the total weights over its individual probability map is nonzero when its control input given by (6.18) approaches zero. For example, we may set the control law of an agent in this case pointing to the cell that takes the largest weight over its whole individual probability map (we can select the cell with the smallest index number if more than one cell takes the largest weight). However, the control input should be bounded such that the agent will not jump out of its current task region for collision avoidance, i.e., to let $\mu_{i,k+1} \in \mathbb{M}_{i,k}$ all the time.

6.4 Simulation

6.4.1 Simulation Environment

We deploy multiple UAVs to search for four ground targets. The whole surveillance region is a square region of $[0, 50] \times [0, 50] \text{ m}^2$ as shown in Fig. 6.3(a), within which lie two crossing roads denoted by $\mathbb{O}_R \subset \mathbb{O}$. There are four targets staying or moving only on the roads and no target appearing outside the roads in the surveillance region. At time k , each target z ($z = 1, 2, 3, 4$) randomly moves to one of the cells in the set $\{g \in \mathbb{O}_R : \|g - \text{Tar}_{z,k-1}\| \leq V_{\text{Tar}}\}$ where $\text{Tar}_{z,k-1}$ is the cell it stays in at time $k-1$ and V_{Tar} is the largest possible speed of target movement. Hence, $P(\theta_{g,k} = 1 | \theta_{r,k-1} = 1) = 1 / \sum_{g \in \mathbb{O}_R} \mathbb{1}_{\{g \in \mathcal{D}_r\}}$ for $r \in \mathbb{O}_R$, where $\mathcal{D}_r = \{g \in \mathbb{O}_R : \|g - r\| \leq V_{\text{Tar}}\}$. Initially, we set $\mathcal{Q}_{i,g,0} = 0$ for all i and g within roads (i.e., $\mathcal{P}_{i,g,k} = 0.5$ for $g \in \mathbb{O}_R$), and $\mathcal{Q}_{i,g,0}$ to a fixed large value for g outside the roads (i.e., $\mathcal{P}_{i,g,k} \approx 0$ for $g \notin \mathbb{O}_R$). The detection probability function is assumed to be $f(h_{i,k}) = K_1 e^{-K_2(h_{i,k} - \underline{h})^2}$, where K_1 and K_2 are positive parameters satisfying the conditions in (6.7). We test the proposed target search method in two scenarios. In Scenario I, all targets appear at $k = 0$ and keep stationary during the whole searching process, i.e., $V_{\text{Tar}} = 0 \text{ m/s}$. In Scenario II, we set $V_{\text{Tar}} = 1 \text{ m/s}$ to

test the influence of target mobility on the convergence of probability maps. The four targets also appear at $k = 0$ and do not disappear during the search. In these two scenarios, we verify the effectiveness of the proposed target search method by deploying different number of UAVs. The initial positions of UAVs are randomly selected within region $[0, 5] \times [0, 5] \text{ m}^2$, and the initial heights are 5m. The partition $\{\mathbb{M}_{1,k}, \dots, \mathbb{M}_{N,k}\}$ is generated by Voronoi partition. The communication range is set as $R_c = 20\text{m}$ and the communication control protocol in Chapter 5 is applied for connectivity maintenance. Some other key parameters are respectively set as $K_u = 0.3$, $K_\eta = 2$, $q = 0.1$, $\check{p} = 0.99$, $\bar{h} = 10\text{m}$, $\underline{h} = 5\text{m}$, $\alpha = 0$, $u_{max} = 2\text{m/s}$ and $T = 1\text{s}$.

Since the convergence of the individual probability map $\mathcal{P}_{i,g,k}$ of agent i implies that the weight $\phi_{i,k}(g)$ defined by (6.19) approaches 0 for each cell, we define the following weight average to evaluate the convergence performance of the whole network:

$$\phi_k = \frac{1}{NM_R} \sum_{i=1}^N \sum_{g \in \mathbb{O}_R} \phi_{i,k}(g),$$

where M_R denotes the total number of cells within the roads. It is easy to find that the initial value of ϕ_k is $\phi_0 = \frac{1}{NM_R} \sum_{i=1}^N \sum_{g \in \mathbb{O}_R} e^{-K_\phi \|\mathcal{Q}_{i,g,0}\|} = 1$. In the simulations, we compare the results of ϕ_k with different system parameters.

6.4.2 Simulation Results

Fig. 6.3 shows an example of the convergence process of individual probability maps in Scenario I with stationary targets, where the probabilities converge to 1 for the cells within which targets truly exist and 0 for the cells within which no target exists. The snapshots of UAVs in Scenario I are shown in Fig. 6.4. Additionally, ϕ_k finally converges to 0 and the more agents are deployed, the faster it converges as shown in Fig. 6.7(a).

The convergence process of an individual probability map in Scenario II with

mobile targets is shown in Fig. 6.5, where the probabilities for the cells around targets may not converge to 0 as in Scenario I due to the random mobility of targets. However, we still can infer that there are four targets on the roads and have a rough estimation of their positions based on the envelopes of the final probability maps of UAVs. The snapshots of UAV positions in Scenario II at different times are shown in Fig. 6.6. In this case, ϕ_k does not converge to 0 as shown in Fig. 6.7(b). However, a smaller ϕ_k can be obtained with more agents deployed since the collective sensing area becomes larger. Compared with the results in Scenario I, the number of deployed agents has a greater impact on the convergence performance of probability maps in Scenario II with random target mobility. Hence, the algorithm is more robust with more UAVs deployed.

6.5 Conclusions

In this chapter, we have studied the three-dimensional vision-based cooperative control and information fusion in target search by a group of UAVs with limited sensing and communication capabilities. First, a heuristic detection probability model was built which is related to the target discriminability of a camera and varies as a function of altitude. Then, we formulated the target search problem as a coverage optimization problem by balancing the coverage area and the detection performance. A generalized probability map updating model was proposed by considering the information decaying and transmission due to environmental changes such as the target movement. The simulation results showed that the proposed algorithms can make the individual probability maps of all agents converge to the same one which reflects the true environment when the targets are stationary. The influence of target mobility and the number of deployed UAVs on the convergence of probability maps has also been illustrated by simulation.

(a) $k = 0$ second. (b) $k = 10$ seconds. (c) $k = 30$ seconds. (d) $k = 50$ seconds. (e) $k = 70$ seconds. (f) $k = 90$ seconds.

Figure 6.3: The convergence of the probability map of an agent in Scenario I.

(a) $k = 0$ second. (b) $k = 10$ seconds. (c) $k = 30$ seconds. (d) $k = 50$ seconds. (e) $k = 70$ seconds. (f) $k = 90$ seconds.

Figure 6.4: Snapshots of UAVs in Scenario I.

(a) $k = 0$ second. (b) $k = 10$ seconds. (c) $k = 30$ seconds. (d) $k = 50$ seconds. (e) $k = 70$ seconds. (f) $k = 90$ seconds.

Figure 6.5: The convergence of the probability map of an agent in Scenario II.

(a) $k = 0$ second. (b) $k = 10$ seconds. (c) $k = 30$ seconds. (d) $k = 50$ seconds. (e) $k = 70$ seconds. (f) $k = 90$ seconds.

Figure 6.6: Snapshots of UAVs in Scenario II.

(a) Scenario I.

(b) Scenario II.

Figure 6.7: Weight average η_k .

Chapter 7

Adaptive Sensing in Multi-Agent Target Tracking

In this chapter, the adaptive sensing in three-dimensional target tracking by multiple autonomous vehicles based on the time-difference-of-arrival (TDOA) measurements is studied. In our sensing strategy, the framework of Kalman filtering is applied to iteratively estimate the target location and a gradient-based control law is derived for each agent to reach a local optimum configuration. Our sensing strategy still can provide good tracking performance with two agents, i.e., with only one measurement at each time, for estimation of a three-dimensional target location. The connections between different optimization criteria including the CRLB are also analyzed.

In Section 7.1, the sensing model and basic assumptions are introduced. An iterated estimation algorithm is given in Section 7.2. In Section 7.3, an optimal adaptive sensing strategy for target tracking is proposed. The comparisons with other optimization criteria and potential modifications are discussed in Section 7.4. Simulation results are provided in Section 7.5. Section 7.6 concludes the chapter.

7.1 Basic Definitions and Assumptions

Consider a target at position $s \in \mathbb{R}^3$, the discrete-time motion model of which is given by

$$s_{k+1} = f(s_k, \eta_k, w_k),$$

where k is an integer denoting the discrete time instant, s_k denotes the value of s evaluated at time k and η_k is the target motion control input. η_k may be computed from unknown random information such as the unknown velocity and acceleration. However, for the ease of expression, we assume that η_k is known, which in fact is not a limitation of our proposed method. The w_k is assumed to be a zero-mean white Gaussian noise and $E[w_k w_k^T] = Q_k$, $E[w_k w_l^T] = 0$ for $k \neq l$. Q_k is assumed to be positive definite.

The same observation model as given in [88] is applied in our research, i.e., at each time k , agent i at position $\mu_{i,k} = [x_{i,k}, y_{i,k}, z_{i,k}]^T \in \mathbb{R}^3$ ($i = 1, 2, \dots, N$), paired with agent j , obtains a TDOA measurement given by

$$t_{i,j,k} = \frac{\|s_k - \mu_{i,k}\| - \|s_k - \mu_{j,k}\|}{v} + v_{i,j,k},$$

where $v \in \mathbb{R}$ is the propagation speed of the signal emitted by the target and $v_{i,j,k} \in \mathbb{R}$ is the measurement noise. We assume that s and μ are both given under the Cartesian coordinate system. v can be normalized as 1 and N is assumed to be no less than 2. Assuming that each agent can communicate with all the other agents, we only consider the TDOA measurements taken by agent 1 paired with all the other agents which form the measurement vector

$$\mathbf{t}_k = [t_{1,2,k}, t_{1,3,k}, \dots, t_{1,N,k}]^T.$$

We assume that $v_{1,i,k}$ is a zero-mean white Gaussian noise with $E[v_{1,i,k} v_{1,i,k}^T] = 2V_k$, $E[v_{1,i,k} v_{1,j,k}^T] = V_k$ for $i \neq j$ and $E[v_{1,i,k} v_{1,j,l}^T] = 0$ for $k \neq l$, where $V_k > 0$ [88]. Defining $h(s, \mu_1, \mu_i) = \|s - \mu_1\| - \|s - \mu_i\|$ for $i = 2, 3, \dots, N$, $\boldsymbol{\mu} = [\mu_1^T, \mu_2^T, \dots, \mu_N^T]^T$,

$\mathbf{h}(s, \boldsymbol{\mu}) = [h(s, \mu_1, \mu_2), \dots, h(s, \mu_1, \mu_N)]^T$ and $\mathbf{v} = [v_{1,2}, v_{1,3}, \dots, v_{1,N}]^T$ which can be evaluated at arbitrary time, we have

$$\mathbf{t}_k = \mathbf{h}(s_k, \boldsymbol{\mu}_k) + \mathbf{v}_k.$$

We further define $\mathbf{R}_k = \mathbb{E}[\mathbf{v}_k \mathbf{v}_k^T] = V_k \text{circulant}([2, 1, \dots, 1]_{N-1})$, where $\text{circulant}(\bullet)$ generates a square circulant matrix with a given vector [88] and $[2, 1, \dots, 1]_{N-1}$ denotes a row vector of $N - 1$ dimensions with the first entry equal to 2 and all the other entries equal to 1. The estimate of the target location time k based on the measurement information up to time l is denoted as $\hat{s}_{k|l}$. The estimation error is given by $\tilde{s}_{k|l} = s_{k|l} - \hat{s}_{k|l}$ and its associated error covariance matrix by $P_{k|l}$.

7.2 Iterative Estimation

We apply the framework of Kalman filter to estimate the target location, which is similar to the method given in [94]. The position estimate and its associated error covariance matrix are computed iteratively by two steps: prediction step and correction step at each time k .

7.2.1 Prediction Step:

Given the estimate $\hat{s}_{k-1|k-1}$ and its associated error covariance matrix $P_{k-1|k-1}$, the estimate of the state prediction is given by

$$\begin{aligned}\hat{s}_{k|k-1} &= f(\hat{s}_{k-1|k-1}, \eta_{k-1}, 0), \\ P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + W_k Q_k W_k^T,\end{aligned}$$

where $F_k = \frac{\partial f(\hat{s}_{k-1|k-1}, \eta_{k-1}, 0)}{\partial s}$ and $W_k = \frac{\partial f(\hat{s}_{k-1|k-1}, \eta_{k-1}, 0)}{\partial w}$.

7.2.2 Correction Step:

In this step, each agent first collects measurements from all agents, and then solve the following optimization problem:

$$\min_{\xi \in \mathbb{R}^3} C(\xi) = \begin{bmatrix} \mathbf{t}_k - \mathbf{h}(\xi, \boldsymbol{\mu}_k) \\ \hat{s}_{k|k-1} - \xi \end{bmatrix}^T \begin{bmatrix} \mathbf{R}_k \\ P_{k|k-1} \end{bmatrix} \begin{bmatrix} \mathbf{t}_k - \mathbf{h}(\xi, \boldsymbol{\mu}_k) \\ \hat{s}_{k|k-1} - \xi \end{bmatrix}.$$

A minimizer of $C(\xi)$ is taken as the new estimate, i.e.,

$$\hat{s}_{k|k} = \arg \min_{\xi \in \mathbb{R}^3} C(\xi), \quad (7.1)$$

and its associated error covariance matrix is given by

$$P_{k|k} = \left(P_{k|k-1}^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \right)^{-1}, \quad (7.2)$$

where $\mathbf{H}_k = \left[\frac{\partial h(\hat{s}_{k|k}, \mu_{2,k})}{\partial s}, \dots, \frac{\partial h(\hat{s}_{k|k}, \mu_{N,k})}{\partial s} \right]^T$.

One can use Gauss-Newton method with line search to solve (7.1), which gives the following iterative calculations [94]:

$$\begin{aligned} s_{k,0}^+ &= \hat{s}_{k|k-1}, \\ P_{k,0}^+ &= P_{k|k-1}, \\ s_{k,m+1}^+ &= (1 - \alpha_{k,m}) s_{k,m}^+ \\ &\quad + \alpha_{k,m} \left(\hat{s}_{k|k-1} + K_{k,m} \left[\mathbf{t}_k - \mathbf{h}(s_{k,m}^+, \boldsymbol{\mu}_k) - \mathbf{H}_{k,m} (\hat{s}_{k|k-1} - s_{k,m}^+) \right] \right), \\ P_{k,m+1}^+ &= \left(P_{k|k-1}^{-1} + (\mathbf{H}_{k,m}^+)^T \mathbf{R}_k^{-1} \mathbf{H}_{k,m}^+ \right)^{-1}, \end{aligned} \quad (7.3)$$

where

$$\begin{aligned} \mathbf{H}_{k,m}^+ &= \left[\frac{\partial h(s_{k,m}^+, \mu_{1,k}, \mu_{2,k})}{\partial s}, \dots, \frac{\partial h(s_{k,m}^+, \mu_{1,k}, \mu_{N,k})}{\partial s} \right]^T, \\ K_{k,m} &= P_{k|k-1} (\mathbf{H}_{k,m}^+)^T \left(\mathbf{H}_{k,m}^+ P_{k|k-1} (\mathbf{H}_{k,m}^+)^T + \mathbf{R}_k \right)^{-1}. \end{aligned}$$

$m \geq 0$ is the iteration step and $\alpha_{k,m} \geq 0$ is the line search parameter to vary the step size. Stop iteration for computing $s_{k,m+1}^+$ and let $\hat{s}_{k|k} = s_{k,m}^+$, $P_{k|k} = P_{k,m}^+$, if $\|s_{k,m}^+ - s_{k,m-1}^+\| \leq \delta$ or $m = M$, where $\delta > 0$ and $M > 0$ are user defined iteration condition bounds and M should be an integer. When $\alpha_{k,m} = 1$ for all k and m , (7.3) becomes the Iterated Kalman filter [95]. Further, when $\alpha_{k,m} = 1$ for all k and m and $M = 1$ (i.e., only one iteration is implemented for (7.3)), it becomes the well-known Extended Kalman filter.

7.3 Cooperative Control for Adaptive Sensing

At each time k , after getting the estimate $\hat{s}_{k+1|k}$ and its associated error covariance matrix $P_{k+1|k}$, each agent needs to determine the sensing position $\mu_{i,k+1}$ at time $k+1$. We first assume the following first-order discrete-time motion model for the agents:

$$\mu_{i,k+1} = \mu_{i,k} + u_{i,k},$$

where $u_{i,k}$ is the control law of agent i at time k . At current sensing positions of $\boldsymbol{\mu}_k$ for all agents with the predicted error covariance matrix $P_{k+1|k}$, the covariance matrix $P_{k+1,1}^+$ to be computed at time $k+1$ by (7.3) can be seen as a function of the control law $\mathbf{u}_k = [u_{1,k}^T, \dots, u_{N,k}^T]^T$, or a function of the next-time sensing positions of $\boldsymbol{\mu}_{k+1}$. We choose the optimal next-time sensing positions of all agents so as to minimize the determinant of $P_{k+1,1}^+(\boldsymbol{\mu}_{k+1})$, i.e., to solve the following optimization problem:

$$\min_{\boldsymbol{\mu}_{k+1} \in \mathbb{R}^{3N}} \det(P_{k+1,1}^+(\boldsymbol{\mu}_{k+1}))$$

which is equivalent to

$$\begin{aligned} & \max_{\boldsymbol{\mu}_{k+1} \in \mathbb{R}^{3N}} \det\left((P_{k+1,1}^+(\boldsymbol{\mu}_{k+1}))^{-1}\right) \\ & = \det\left(P_{k+1|k}^{-1} + (\mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_{k+1}))^T \mathbf{R}_{k+1}^{-1} \mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_{k+1})\right). \end{aligned} \quad (7.4)$$

If $\mathbf{u}_k$ is constrained, a cell-based search for the optimal solution of $\mu_{i,k+1}$ across the whole region can be implemented, but the computation load may be very high depending on the constraint. Generally, if $\mathbf{u}_k$ is unconstrained and the optimization is non-convex, the global optimal solution cannot be found and one can only use the gradient-descent method to find a local optimal solution. For the generality of discussions, we assume that μ_{k+1} is unconstrained, and thus will try to develop a gradient-descent method for solving (7.4) which is also applicable with a constrained $\mathbf{u}_k$. A local optimum solution of $\mu_{i,k+1}$ can be found for each agent i by using Newton's method:

$$\begin{aligned} \mu_{i,k,0}^+ &= \mu_{i,k}, \\ \mu_{i,k,l+1}^+ &= \mu_{i,k,l}^+ - \beta_{i,k,l} \frac{\partial \left(\det \left((P_{k+1|k+1}(\mu_{k,l}^+))^{-1} \right) \right)}{\partial \mu_i}, \end{aligned} \quad (7.5)$$

where $\mu_{k,l}^+ = [\mu_{1,k,l}^+, \dots, \mu_{N,k,l}^+]$, $l \geq 0$ is the iteration step and $\beta_{i,k,l} \geq 0$ is the line search parameter to vary the step size. Stop iteration for computing $\mu_{i,k,l+1}^+$ and let $\mu_{i,k+1} = \mu_{i,k,l}^+$ if $\|\mu_{i,k,l}^+ - \mu_{i,k,l-1}^+\| \leq \epsilon$ or $l = L$, where $\epsilon > 0$ and $L > 0$ are user defined iteration condition bounds and L should be an integer. Thus, the control law for each agent at time k is given by $u_{i,k} = \mu_{i,k+1} - \mu_{i,k}$.

Remark 7.1. *If the control input is bounded by $u_{i,k} \leq \|u\|_{\max}$. Then, the control law can be determined by the following saturation rule:*

$$u_{i,k} = \begin{cases} \mu_{i,k,l}^+ - \mu_{i,k} & \text{if } \|\mu_{i,k,l}^+ - \mu_{i,k}\| \leq u_{\max}; \\ \frac{\mu_{i,k,l}^+ - \mu_{i,k}}{\|\mu_{i,k,l}^+ - \mu_{i,k}\|} u_{\max} & \text{otherwise,} \end{cases}$$

where $\mu_{i,k,l}^+$ is the final result of (7.5).

Theorem 7.1. *The derivative in (7.5) can be calculated as follows:*

$$\frac{\partial \left(\det \left((P_{k+1,1}^+(\mu_k))^{-1} \right) \right)}{\partial \mu_i}$$

$$= \frac{1}{R_{k+1} \det(P_{k+1,1}^+(\boldsymbol{\mu}_k))} \begin{bmatrix} \text{Tr} \left(P_{k+1,1}^+(\boldsymbol{\mu}_k) \left(\Delta_{i,k+1}(x) + (\Delta_{i,k+1}(x))^T \right) \right) \\ \text{Tr} \left(P_{k+1,1}^+(\boldsymbol{\mu}_k) \left(\Delta_{i,k+1}(y) + (\Delta_{i,k+1}(y))^T \right) \right) \\ \text{Tr} \left(P_{k+1,1}^+(\boldsymbol{\mu}_k) \left(\Delta_{i,k+1}(z) + (\Delta_{i,k+1}(z))^T \right) \right) \end{bmatrix}, \quad (7.6)$$

where $\Delta_{i,k+1}(\theta)$ for $\theta \in \mathbb{R}$ is given by

$$\Delta_{i,k+1}(\theta) = \begin{cases} \frac{\partial g(\hat{s}_{k+1|k}, \mu_{1,k})}{\partial(\theta)} \frac{\mathbf{1}^T}{N} \mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k) & \text{if } i = 1; \\ \frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial(\theta)} \left(\mathbf{e}_{i-1}^T - \frac{\mathbf{1}^T}{N} \right) \mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k) & \text{otherwise,} \end{cases} \quad (7.7)$$

and $g(s, u) = \frac{s - \mu}{\|s - \mu\|}$, $\mu = [x, y, z]^T$, $\mathbf{1}$ denotes the column vector of dimension $N - 1$ with all entries equal to 1 and $\mathbf{e}_i$ the column vector of dimension $N - 1$ with the i -th entry equal to one and all the other entries equal to 0.

Proof. First, we have

$$\frac{\partial \left(\det \left((P_{k+1,1}^+(\boldsymbol{\mu}_k))^{-1} \right) \right)}{\partial \mu_i} = \begin{bmatrix} \frac{\partial \left(\det \left((P_{k+1,1}^+(\boldsymbol{\mu}_k))^{-1} \right) \right)}{\partial x_i} \\ \frac{\partial \left(\det \left((P_{k+1,1}^+(\boldsymbol{\mu}_k))^{-1} \right) \right)}{\partial y_i} \\ \frac{\partial \left(\det \left((P_{k+1,1}^+(\boldsymbol{\mu}_k))^{-1} \right) \right)}{\partial z_i} \end{bmatrix}. \quad (7.8)$$

Now we consider the first entry in the above vector. Following the rule of derivative of determinant [129], we can get

$$\frac{\partial \left(\det \left((P_{k+1,1}^+(\boldsymbol{\mu}_k))^{-1} \right) \right)}{\partial x_i} = \frac{1}{\det(P_{k+1,1}^+(\boldsymbol{\mu}_k))} \text{Tr} \left(P_{k+1,1}^+(\boldsymbol{\mu}_k) \frac{\partial (P_{k+1,1}^+(\boldsymbol{\mu}_k))^{-1}}{\partial x_i} \right).$$

Further, from (7.4) we have

$$\begin{aligned}
\frac{\partial (P_{k+1,1}^+(\boldsymbol{\mu}_k))^{-1}}{\partial x_i} &= \frac{\partial \left[(\mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k))^T \mathbf{R}_{k+1}^{-1} \mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k) \right]}{\partial x_i} \\
&= \frac{\partial (\mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k))^T}{\partial x_i} \mathbf{R}_{k+1}^{-1} \mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k) \\
&\quad + (\mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k))^T \mathbf{R}_{k+1}^{-1} \frac{\partial (\mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k))}{\partial x_i} \\
&= \mathbf{R}_{k+1}^{-1} \left[\Delta_{i,k+1}(x) + (\Delta_{i,k+1}(x))^T \right],
\end{aligned} \tag{7.9}$$

where $\Delta_{i,k+1}(x)$ is defined as

$$\Delta_{i,k+1}(x) = \mathbf{R}_{k+1} \left(\frac{\partial \mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k)}{\partial x_i} \right)^T \mathbf{R}_{k+1}^{-1} \mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k). \tag{7.10}$$

and

$$\left(\frac{\partial \mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k)}{\partial x_i} \right)^T = \left[\frac{\partial^2 h(\hat{s}_{k+1|k}, \mu_{1,k}, \mu_{2,k})}{\partial s \partial x_i}, \dots, \frac{\partial^2 h(\hat{s}_{k+1|k}, \mu_{1,k}, \mu_{N,k})}{\partial s \partial x_i} \right]. \tag{7.11}$$

From the definition of $h(s, \mu_1, \mu_i)$, we can get for $i = 1$,

$$\frac{\partial^2 h(\hat{s}_{k+1|k}, \mu_{1,k}, \mu_{2,k})}{\partial s \partial x_1} = \dots = \frac{\partial^2 h(\hat{s}_{k+1|k}, \mu_{1,k}, \mu_{N,k})}{\partial s \partial x_1} = \frac{\partial g(\hat{s}_{k+1|k}, \mu_{1,k})}{\partial x}, \tag{7.12}$$

and for $i = 2, \dots, N, j = 1, \dots, N$,

$$\frac{\partial^2 h(\hat{s}_{k+1|k}, \mu_{1,k}, \mu_{i,k})}{\partial s \partial x_i} = \begin{cases} 0 & \text{if } i \neq j; \\ \frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial x} & \text{otherwise.} \end{cases} \tag{7.13}$$

Substituting (7.12) and (7.13) into (7.11), we get

$$\left(\frac{\partial \mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k)}{\partial x_i} \right)^T = \begin{cases} \frac{\partial g(\hat{s}_{k+1|k}, \mu_{1,k})}{\partial x} \mathbf{1}^T & \text{if } i = 1; \\ \frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial x} \mathbf{e}_{i-1}^T & \text{otherwise.} \end{cases}$$

On the other hand, since $\mathbf{R}_{k+1} = R_k \text{circulant}([2, 1, \dots, 1]_{N-1})$, we can get

$$\mathbf{R}_{k+1}^{-1} = R_{k+1}^{-1} \text{circulant} \left(\left[\frac{N-1}{N}, -\frac{1}{N}, \dots, -\frac{1}{N} \right] \right)$$

which implies that

$$\begin{aligned} \mathbf{1}^T \mathbf{R}_{k+1}^{-1} &= \frac{1}{R_{k+1}} \frac{\mathbf{1}^T}{N}, \\ \mathbf{e}_{i-1}^T \mathbf{R}_{k+1}^{-1} &= \frac{1}{R_{k+1}} \left(\mathbf{e}_{i-1}^T - \frac{\mathbf{1}^T}{N} \right). \end{aligned}$$

Then, we can derive from (7.10) that (7.7) holds for $\Delta_{i,k+1}(x)$. Combining (7.8) and (7.9), we can prove that (7.6) holds for the first entries of the vectors. In the same way with similar definitions of $\Delta_{i,k+1}(y)$ and $\Delta_{i,k+1}(z)$, we can show that (7.6) also holds for the second and the third entries of the vectors. $\square$

Theorem 7.2. *A set of stationary points for the agents by implementing the control law (7.5) is given by*

$$\Upsilon_k = \left\{ \boldsymbol{\mu}_k = [\mu_{1,k}^T, \dots, \mu_{N,k}^T]^T : \sum_{i=1}^N g(\hat{s}_{k+1|k}, \mu_{i,k}) = 0, \mu_{1,k}, \dots, \mu_{N,k} \in \mathbb{R}^3 \right\},$$

where $g(s, \mu)$ is the same as defined in (7.7).

Proof. First, at time k we have

$$\mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k) = \begin{bmatrix} [g(\hat{s}_{k+1|k}, \mu_{1,k}) - g(\hat{s}_{k+1|k}, \mu_{2,k})]^T \\ \vdots \\ [g(\hat{s}_{k+1|k}, \mu_{1,k}) - g(\hat{s}_{k+1|k}, \mu_{N,k})]^T \end{bmatrix}$$

which implies

$$\begin{aligned} \frac{\mathbf{1}^T}{N} \mathbf{H}_{k+1,0}^+ (\boldsymbol{\mu}_k) &= \left[g(\hat{s}_{k+1|k}, \mu_{1,k}) - \frac{1}{N} \sum_{i=1}^N g(\hat{s}_{k+1|k}, \mu_{i,k}) \right]^T, \\ \left(\mathbf{e}_{i-1}^T - \frac{\mathbf{1}^T}{N} \right) \mathbf{H}_{k+1,0}^+ (\boldsymbol{\mu}_k) &= \left[-g(\hat{s}_{k+1|k}, \mu_{i,k}) + \frac{1}{N} \sum_{i=1}^N g(\hat{s}_{k+1|k}, \mu_{i,k}) \right]^T, \\ &\quad \forall i = 2, \dots, N. \end{aligned} \quad (7.14)$$

On the other hand, we can introduce the following coordinate transformation for $\mu_{i,k}$ ($i = 1, \dots, N$) using the spherical coordinate system at each time k :

$$\mu_{i,k} = [x_{i,k}, y_{i,k}, z_{i,k}]^T = \hat{s}_{k+1|k} + r_{i,k} [\cos \theta_{i,k} \cos \alpha_{i,k}, \cos \theta_{i,k} \sin \alpha_{i,k}, \sin \theta_{i,k}]^T$$

such that

$$g(\hat{s}_{k+1|k}, \mu_{i,k}) = -[\cos \theta_{i,k} \cos \alpha_{i,k}, \cos \theta_{i,k} \sin \alpha_{i,k}, \sin \theta_{i,k}]^T.$$

Then, we have

$$\begin{aligned} \frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial r} &= 0, \\ \frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial \theta} &= [\sin \theta_{i,k} \cos \alpha_{i,k}, \sin \theta_{i,k} \sin \alpha_{i,k}, -\cos \theta_{i,k}]^T, \\ \frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial \alpha} &= [\cos \theta_{i,k} \sin \alpha_{i,k}, -\cos \theta_{i,k} \cos \alpha_{i,k}, 0]^T, \end{aligned}$$

which implies

$$\left[\frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial r}, \frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial \theta}, \frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial \alpha} \right] g(\hat{s}_{k+1|k}, \mu_{i,k}) = 0.$$

Thus,

$$\left[\frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial x}, \frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial y}, \frac{\partial g(\hat{s}_{k+1|k}, \mu_{i,k})}{\partial z} \right] g(\hat{s}_{k+1|k}, \mu_{i,k}) = 0.$$

From (7.7) and (7.14), if we have $\boldsymbol{\mu}_k \in \Upsilon_{k+1}$, i.e., $\sum_{i=1}^N g(\hat{s}_{k+1|k}, \mu_{i,k}) = 0$, then it follows that

$$\Delta_{i,k+1}(x) = \Delta_{i,k+1}(y) = \Delta_{i,k+1}(z) = 0,$$

which implies $\frac{\partial(\det(P_{k+1,1}^+(\boldsymbol{\mu}_k)))}{\partial \mu_i} = 0$. Hence, by implementing the control law (7.5), agents will stay stationary at $\boldsymbol{\mu}_k$. $\square$

Remark 7.2. *Theorem 7.2 shows one type of local optimum geometric configuration of the agents. In some applications, it is not necessary to consider three-dimensional motion models of targets and agents, where the models can be simplified by reducing the dimension of state vectors. For example, in the tracking of a ground target by airplanes which fly at a constant altitude, the target and airplane positions can all be considered as two-dimensional vectors. That is, $s_k = [\bar{s}^T, 0]^T$ and $\mu = [\bar{\mu}_i^T, b]^T$, where $\bar{s}, \bar{\mu}_i \in \mathbb{R}^2$ and $b > 0$ is a given flight altitude of the airplanes. In this case, we only need to calculate the estimate of $\bar{s}$ and its associated error covariance matrix $\bar{P}$, and control the planar coordinate of agent $\bar{\mu}_i$. The derivation of the formulae is the same with that given in Section 7.3 except that some matrices and vectors need to be redefined with appropriate dimensions.*

7.4 Discussions

If we replace (7.4) with the optimization problem:

$$\max_{\boldsymbol{\mu}_k \in \mathbb{R}^{3N}} \det \left((\mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k))^T \mathbf{R}_{k+1}^{-1} \mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k) \right), \quad (7.15)$$

then the optimal control problem is equivalent to Problem 3 in [88] which minimizes the determinant of the CRLB. However, such optimization criterion is applicable only when $\mathbf{H}_{k+1,0}^+(\boldsymbol{\mu}_k)$ is of full column rank, which requires the number of agents is no less than 4. This increases the application cost especially when the vehicles are expensive to get. The sensing strategy (7.15) using the CRLB will be compared with our optimal sensing strategy (7.4) in simulations.

To simplify the computation, the optimal sensing strategy (7.4) can be replaced with the following optimization problem for $i \geq 2$:

$$\max_{\mu_{i,k+1} \in \mathbb{R}^3} \det \left(P_{k+1|k}^{-1} + (\mathcal{H}_{i,k+1,0}^+ (\mu_{i,k+1}))^T \mathcal{R}_{i,k+1}^{-1} \mathcal{H}_{i,k+1,0}^+ (\mu_{i,k+1}) \right), \quad (7.16)$$

where

$$\mathcal{H}_{i,k+1,0}^+ (\mu_{i,k+1}) = \left[\frac{\partial h(\hat{s}_{k+1|k}, \mu_{1,k+1}, \mu_{2,k+1})}{\partial s}, \dots, \frac{\partial h(\hat{s}_{k+1|k}, \mu_{1,k+1}, \mu_{i,k+1})}{\partial s} \right]^T,$$

$$\mathcal{R}_{i,k+1} = V_k \text{circulant}([2, 1, \dots, 1]_{i-1}),$$

given that $\mu_{j,k+1}$ ($j = 1, \dots, i-1$) have been determined. Hence, $\mu_{i,k+1}$ ($i = 2, \dots, N$) can be determined in a sequential order by solving the optimization problem (7.16) with $\mu_{1,k+1} = \mu_{1,k}$. It can be found that the row dimension of $\mathcal{H}_{k+1,0}^+$ is less than that of $\mathbf{H}_{k+1,0}^+$ for $i < N$. Hence, the computation load is reduced for agents $i < N$. Specifically, agent 1 does not need to update position by the suboptimal strategy. Such suboptimal sensing strategy will be compared with the optimal one.

Remark 7.3. A similar suboptimal sensing strategy is used to choose the sensing positions in [93], where the noises of different measurements are assumed to be independent. Our case is more general in that the measurement noises are correlated, i.e., $\mathcal{R}_{i,k+1}$ is not diagonal.

Due to the speed limitation of agents and collision avoidance, it is better to keep the agents at a fixed but relatively small distance to the target. Thus, we can refine the optimization problem (7.4) using the potential field method as follows:

$$\begin{aligned} \max_{\mu_{k+1} \in \mathbb{R}^{3N}} \mathcal{H}(\mu_{k+1}) = & \det \left((P_{k+1,1}^+ (\mu_{k+1}))^{-1} \right) - \Gamma_1 \sum_{i=1}^N (\|\mu_{i,k+1} - \hat{s}_{k+1|k}\| - d)^2 \\ & - \Gamma_2 \sum_{i=1}^N \sum_{j=i}^N \|\mu_{i,k+1} - \mu_{j,k+1}\|^{-1}, \end{aligned}$$

where d is the distance to be maintained between each agent and the estimated target location and Γ_1 is the gain parameter for the trade-off between the two costs, Γ_2 is the gain parameter to make the repelling force between each two agents become significant only when the distance between them becomes very small.

7.5 Simulation

7.5.1 Simulation Environment

In this section, we validate the effectiveness of the proposed algorithm by simulation, and compare the performances of different algorithms. To distinguish different algorithms, we name the one based on (7.4) the “optimal method”, the one based on (7.16) the “suboptimal method”, the one based on (7.4) but using EKF filter the “EKF method”, and the one based on (7.15) the “CRLB method”. The target motion model is set to be

$$s_{k+1} = f(s_k, w_k) = s_k + \begin{bmatrix} 25 \cos \frac{\pi}{20} (k+1) - 25 \cos \frac{\pi}{20} k \\ 25 \sin \frac{\pi}{20} (k+1) - 25 \sin \frac{\pi}{20} k \\ 0.01 \end{bmatrix} + w_k.$$

In addition, we set $s_0 = [25, 0, -25]^T$, $\hat{s}_{1|0} = [0, 0, 0]^T$, $P_{1|0} = \text{diag}(10^4, 10^4, 10^4)$, $Q_k = \text{diag}(1, 1, 1)$, $V_k = 30$, $M = 20$, $W = 10$, $\delta = 0.01$, $\epsilon = 0.01$, $\Gamma_1 = 0.5$, $\Gamma_2 = 0.1$ and $\|u\|_{\max} = 5$.

We use the mean square estimation error as the performance index, i.e.,

$$D_k = \text{E} \left[\|\hat{s}_{k|k} - s_k\|^2 \right].$$

We use the sampled mean to approximate the true value of D_k , which is averaged over 1000 Monte Carlo simulations. In each simulation, the agents are initially randomly deployed within region $[-20, 20]^3$. Furthermore, to examine the converged

positions of agents according to Theorem 7.2, we define the following averaged vector norm:

$$G_k = \left\| \sum_{i=1}^N \frac{\hat{s}_{k|k-1} - \mu_{i,k}}{\|\hat{s}_{k|k-1} - \mu_{i,k}\|} \right\|.$$

7.5.2 Simulation Results

Fig. 7.1 shows the snapshots of agent positions and the estimate and true position of the target position (red square: current estimate of target position, blue star: true target position, green balls: agent positions). Fig. 7.2 to Fig. 7.5 show that the optimal and suboptimal sensing strategies as well as the conventional EKF method can properly track the target using our optimization criterion even when the number of agents is less than 4. It is also shown that the suboptimal sensing strategy has the worst steady-state performance compared with the other three strategies, but similar performance can be obtained when the number of agents is large. In addition, the optimal sensing strategy has the best performance when the CRLB method is inapplicable and has nearly the same performance with the CRLB method when the number of agents is no less than 4, both of which are superior to the other two methods. Based on the above results, when the number of vehicles is not sufficient, one can still track the target using our optimization criteria, no matter optimal or suboptimal; when the number of vehicles is sufficient, one may prefer to use the suboptimal sensing strategy to reduce the computation load while obtaining a good enough tracking performance. In the meantime, Fig. 7.6 shows that the agents by our optimal sensing strategy always converge to the stationary points defined by Theorem 7.2.

7.6 Conclusions

In this chapter, we investigated the adaptive sensing for three-dimensional target tracking by multiple autonomous vehicles based on observations from TDOA sensors.

Figure 7.1: Snapshots of agents, target and estimate trajectories.

(a) Initial stage.

(b) Steady state.

Figure 7.2: The estimation performance with 2 agents.

(a) Initial stage.

(b) Steady state.

Figure 7.3: The estimation performance with 3 agents.

(a) Initial stage.

(b) Steady state.

Figure 7.4: The estimation performance with 4 agents.

(a) Initial stage.

(b) Steady state.

Figure 7.5: The estimation performance with 6 agents.

(a) Initial stage.

(b) Steady state.

Figure 7.6: Convergence of geometric configuration.

A filtering method similar to the IKF method was applied to estimate the target position. Then, the optimal adaptive sensing strategy was developed by minimizing the determinant of the estimation error covariance matrix. A gradient-based control law was derived to achieve the local optimum for each agent and a class set of geometric configurations for the converged agent positions was given. A suboptimal sensing strategy was also put forward to reduce computation load of the optimal sensing strategy. Finally, we testified the proposed sensing strategies and compared them with the sensing strategy using the conventional EKF method and the CRLB by simulation. It was shown that the proposed optimal sensing strategy is superior to other sensing strategies, and can still provide good tracking performance even when the number of vehicles is insufficient.

Chapter 8

Development of A Multi-agent Simulator and Testing Results

In this chapter, a multi-purpose three dimensional simulation platform for MSNs is developed based on Unreal Engine, LabView, Matlab and OMNet++. The platform is used to testify the performance of the proposed target search, information fusion and cooperative control algorithms in this thesis. The main objectives of the platform are described as follows.

- To perform multiple tasks such as formation flight, collaborative search and exploration (especially, vision-based approaches), collaborative tracking and dynamic task assignment.
- To simulate different three dimensional environments in the real world by importing the true environment information such as urban, rural and indoor environments.
- To support kinematic models of different types of vehicles, which can be tuned and tested through the hardware interface.
- To operate distributively in a networked environment and provide a user-friendly and interactive visual interface.

Figure 8.1: The platform architecture.

8.1 Design of the Simulation Platform

The overall architecture of the simulation platform is divided into six portions, namely Real World, Transmission, Control Console, Control Algorithms, Virtual Network and Virtual World, as displayed in Fig. 8.1.

Virtual world is simulated by Unreal Engine and created by Unreal Editor [130], which help users build their own map and environmental setups such as roads and buildings, etc. LabView [131], acting as the console center for both hardware and software, is the interface of Unreal Engine and virtual environment. It communicates with the Unreal server to obtain information from sensors (as shown in Fig. 8.2), and sends commands generated by implementing the Matlab control programme to control the motions of vehicles. OMNet++ [132] is used to simulate the corresponding virtual networks for vehicle communications. For localization in real world, Ultra-Wideband localization software [133] is used in indoor environment, and Globe Position System (GPS) is used in outdoor environment. In the virtual

Figure 8.2: Sensors on UAVs.

world, range scanner sensor (sonar or IR) is used to measure distance. Camera is employed in both real and virtual worlds for tracking and navigation.

In our design, a robot is constructed of four parts: chassis, parts, joints, and attached items. The procedures of building a robot is as follows: 1) build geometric models for all objects used to construct a robot; 2) create sub-classes for all robot parts, and form a new class for the overall robot (a class is a piece of Unreal Script to describe the physical attributes of a robot); 3) configure how the chassis, parts and auxiliary items are connected to each other. In our experiment, the GAUI 330X Quad Flyer UAV [134] is used as shown in Fig. 8.3, which is a quad-rotor electrical helicopter with flight stabilization control.

The platform testing in virtual world is held in both indoor and outdoor environments, e.g., NTU Sensor Network Lab (SNL) and NTU Student Recreation Centre (SRC). The dimension of the virtual lab is $16.4m \times 9.5m \times 2.6m$, which is the same size of the real world SNL, and the dimension of SRC space is $300m \times 250m \times 400m$, which is an approximated dimension of the real world SRC based on Google map. The comparison between real and virtual environments is shown in Fig. 8.4. We

Figure 8.3: UAV model.

also create some new virtual environments for implementation of experiments such as the environment shown in Fig. 8.5 which includes buildings and trees as obstacles and cars and roads as monitoring objects.

8.2 Experiments

We use the platform to implement the simulation experiments of vision-based cooperative target search as addressed in Chapter 5. In the experiment, we simulate the the environment of two car accidents where two wounded men are lying on the roads waiting for search and rescue (as shown in Fig. 8.6). Two UAVs with downward facing cameras are deployed to search for the targets and they know that the targets are within the roads. The geometries of the roads and buildings surrounding the roads are assumed to be known by the UAVs (as shown in Fig. 8.7). During the search, the UAVs share information by communication and fly by the control strategy given in Chapter 5. Obstacle avoidance measures are also taken to avoid collision with the buildings. The targets can be detected by the UAVs with a detection probability of 0.98 if they are within the sight field of UAVs. The UAVs are flying at a constant height of 9.5m above the roads with an angle of horizon of 60 degrees. False alarm probability is set to be 0.01 and the communication range is set to be large enough so that the two UAVs are always connected. We let the UAVs

(a) NTU Sensor Network Lab.

(b) Virtual lab.

(c) NTU Student Recreation Centre.

(d) Virtual SRC.

Figure 8.4: Simulating the real environment.

Figure 8.5: Virtual environment.

Figure 8.6: Environment for vision-based target search.

hover over the targets if the targets are found and assign each target to a different UAV for allocation of the multi-target search task.

Fig. 8.8 shows the simulation result of the moving trajectories of the UAVs (red circles: trajectory of UAV 1, blue crosses: trajectory of UAV 2). It can be seen that the two UAVs successfully locate the two targets and finally hover over the target positions. Their trajectories are both within the road region with no collision into the building obstacles. Furthermore, the information sharing between the two UAVs leads to a good collaborative task allocation for multiple target search.

8.3 Conclusions

In this chapter, we have developed a three dimensional simulation platform for MSNs using softwares including Unreal Engine, LabView, Matlab and OMNet++. The platform has been used to simulate the environment of vision-based target search and testify the effectiveness of the proposed target search algorithm.

Figure 8.7: Starting positions of UAVs and roads.

Figure 8.8: Moving trajectories of the two UAVs.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

The information fusion and cooperative control problems are interesting topics of MSNs. This thesis summarizes the results on the analysis and design of information fusion and cooperative control algorithms for target search and localization.

- First, a CI-based diffusion Kalman filtering (CI-DKF) algorithm has been designed which allows each agent to obtain a stable estimate by sharing information only with its neighbors. The CI-DKF algorithm can be applied in the case of lacking local observability. Additionally, in the case that all agents are locally unobservable, i.e., the estimates of all agents are unstable merely based on the local measurement information, a consensus-based information diffusion scheme has been designed to achieve the local observability within a finite time duration.
- Second, a distributed multiple targets localization and pursuit scheme for MSNs has been proposed. Based on the discrete measurements of the unknown target energy intensity field, a distributed control law was derived by approximating the spatial gradient of the field. Robots, that are categorized into two

types: leaders and followers, move by different control strategies to accomplish different tasks. Leaders are mainly controlled by the gradient climbing force to pursue the targets while followers are controlled by the formation control force to keep team formation and network connectivity. The relation between convergence error and system parameter setting was also analyzed.

- Third, the cooperative control and information fusion in target search by a group of UAVs with two and three dimensional discrete-time motion models has been studied. By dividing the whole surveillance region into cells, each agent keeps an individual probability map about the target existence within each cell. The map is first updated by local measurements based on Bayesian rule, which was simplified to a linear update by introducing a nonlinear transformation of the probability map. A consensus-like distributed fusion scheme was further proposed for multi-agent map fusion. In the vision-based target search where the target discriminability of a camera varies as a function of altitude, a time-varying detection probability is induced during the movement of UAVs. Further, the influence of information decaying and transmission due to environmental changes such as the target movement have been considered.
- Fourth, the adaptive sensing for three-dimensional target tracking in MSNs has been investigated based on the observations from TDOA sensors. The framework of Kalman filtering was applied to iteratively estimate the target location which incorporates the Gauss-Newton method. By minimizing the determinant of the estimation covariance matrix, a gradient-based control law was derived for each agent to reach a local optimum geometric configuration at each time. Comparisons between the algorithms using different optimization criteria and filtering methods have been made and some potential extensions of the proposed algorithm has also been analyzed.
- Last, a multi-purpose three dimensional simulation platform for MSNs has been built up based on Unreal Engine, LabView, Matlab and OMNet++. The platform can be used to create virtual environments for simulating the real envi-

ronments of implementations and testify the performance of target search, information fusion and cooperative control algorithms. The vision-based target search has been simulated on the platform, which was shown to be applicable in environments with obstacles.

9.2 Future Work

Except for those topics mentioned in the conclusion part, several other potential research directions are summed up as follows.

- The discussion of the estimation and fusion methods in Chapter 3 is based on linear system models. However, in many real applications, the system models may be nonlinear. Thus, it is meaningful to modify the proposed methods to be applied with nonlinear system models. Furthermore, the proposed methods require the transmission of estimation error covariance matrices among neighboring agents which might be energy consuming if the target state is of high dimension. Therefore, it is very interesting and necessary to design fusion methods that still offer good estimation performance but with low communication requirements. One possible means is simplification of the estimation error covariance matrix by compressed sensing or component analysis.
- If the accurate energy intensity model of the observed targets is known and parameterized, we can combine the control strategy proposed in Chapter 4 and the CI-DKF algorithm proposed in Chapter 3 to pursue multiple targets while estimating their source energy and locations adaptively. However, when such two methods are combined, a trade-off between the formation control and the estimation performance must be made, which motivates to design a new cost function to derive the optimal control law.
- In Chapter 5 and Chapter 6, only a first-order discrete-time motion model is assumed. In real applications, different types of UAV with different kinematic

models may be applied. Hence, the designed algorithms should be modified to fit for different nonlinear UAV kinematic models. Moreover, the detection probability model in Chapter 6 is based on the assumption of a fixed-zoom camera, while in reality the focus length of a camera is adjustable. Therefore, the detection probability model should be generalized to include the focus length as a parameter. With an adjustable focus length, it is of great interest to design a so-called “adaptive target search scheme”, i.e., each agent can adaptively focus with a higher detection probability on the region over which it has a larger uncertainty about the target existence.

- In Chapter 7, we have designed an adaptive sensing scheme for target tracking in MSNs based on the TDOA observation model. However, the estimation of each agent is based on all to all communications and may not be suitable for large-scale distributed networks. Therefore, it is worth our efforts to modify the scheme to realize distributed adaptive sensing which can also be applicable to the case with time-varying communication topologies.

Appendix A

Elementary Concepts and Notations

1 Graph

Here we present some basic terminologies and definitions of the graph theory used in this thesis following the treatments in the literatures [135–137]. The network topology of N agents with limited communication range R_c is modeled by a time-varying undirected graph $\mathcal{G}_k$ which consists of a constant vertex set $\mathcal{V} = \{1, 2, \dots, N\}$ and a time-varying edge set $\mathcal{E}_k = \{\{i, j\} : i, j \in \mathcal{V}; \|\mu_{i,k} - \mu_{j,k}\| \leq R_c\}$ at each time step k , i.e., $\mathcal{G}_k = (\mathcal{E}_k, \mathcal{V})$. For $i, j \in \mathcal{V}$ and $i \neq j$, $\{i, j\} \in \mathcal{E}_k$ is an unordered pair of vertices. $\mathcal{G}_k$ is connected at time k if for any two vertices i and j there exists a sequence of edges (a path) $\{i, \nu_1\}, \{\nu_1, \nu_2\}, \dots, \{\nu_{n-1}, \nu_n\}, \{\nu_n, j\}$ in $\mathcal{E}_k$. Let $\mathcal{N}_{i,k} = \{j \in \mathcal{V} | \{i, j\} \in \mathcal{E}_k\} \cup \{i\}$ denote the set of neighbors of vertex i at each time k , where a vertex is assumed to be a neighbor of itself. The degree (number of neighbors) of vertex i at time k is denoted as $d_{i,k} = |\mathcal{N}_{i,k}|$.

2 Stochastic Matrices

Here we introduce the definitions of stochastic matrices following the treatments in [125, 138]. Consider a square matrix $A = \{a_{m,n}\}$ with all entries being nonnegative. If $\sum_n a_{m,n} = 1$ for each m , then A is called a row stochastic matrix. If $\sum_m a_{m,n} = 1$

for each n , then A is called a column stochastic matrix. If $\sum_n a_{m,n} = 1$ for each m and $\sum_m a_{m,n} = 1$ for each n are both satisfied, then A is called a doubly stochastic matrix. Note that a doubly stochastic matrix must be a row stochastic matrix and a column stochastic matrix.

3 Voronoi Partition

We follow the literatures [124,139,140] to state the the definitions related to Voronoi partition. Given a closed set $S \in \mathbb{R}^p$, a partition of S is a subdivision of S into connected subsets that are disjoint except for their boundary. Formally, a partition of S is a collection of closed connected sets $\{\mathbb{W}_1, \dots, \mathbb{W}_N\}$ that verify

$$S = \bigcup_{i=1}^N \mathbb{W}_i \quad \text{and} \quad \text{int}(\mathbb{W}_i) \cap \text{int}(\mathbb{W}_j) = \emptyset,$$

for $i, j = 1, \dots, N$ and $i \neq j$. Given a distance function $f: \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}_{\geq 0}$, and N distinct points $\mu_{i,k} \in S$ ($i = 1, \dots, N$) at time k , the Voronoi partition of S at time k generated by the N points is the collection of sets $\{\mathbb{V}_{1,k}(\boldsymbol{\mu}_k), \dots, \mathbb{V}_{N,k}(\boldsymbol{\mu}_k)\}$ that verify

$$\mathbb{V}_{i,k}(\boldsymbol{\mu}_k) = \{r \in S \mid f(r, \mu_{i,k}) \leq f(r, \mu_{j,k}), \forall j \neq i \text{ and } j = 1, \dots, N\},$$

where $\boldsymbol{\mu}_k = [\mu_{1,k}^T, \dots, \mu_{N,k}^T]^T$. We refer to $\mathbb{V}_{i,k}(\boldsymbol{\mu}_k)$ as the Voronoi cell generated by $\mu_{i,k}$. We further define $\mathcal{N}_{V_{i,k}} = \{j \mid \mathbb{V}_{i,k} \cap \mathbb{V}_{j,k} \neq \emptyset, j \in \mathcal{N}_{i,k}\}$ as the set of Voronoi neighbors for each $i = 1, \dots, N$.

Author's Publications

Journal papers:

1. **J. Hu**, L. Xie, and C. Zhang, "Energy-Based Multiple Target Localization and Pursuit in Mobile Sensor Networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 1, pp. 212-220, 2012 (regular paper).
2. **J. Hu**, L. Xie, and C. Zhang, "Diffusion Kalman Filtering Based on Covariance Intersection," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 891-902, 2012 (regular paper).
3. **J. Hu**, L. Xie, and K.Y. Lum, J. Xu, "Multi-agent Information Fusion and Cooperative Control in Target Search," *Accepted as a regular paper by IEEE Transactions on Control Systems and Technology*.
4. **J. Hu**, L. Xie, and J. Xu, "Multi-agent Cooperative Target Search," *Submitted to Journal of Guidance, Control, and Dynamics*.

Conference papers:

1. **J. Hu**, C. Zhang, and L. Xie, "Multiple Mobile Target Localization and Pursuit Based on Discrete Measurements of the Energy Intensity Field," in *Proceedings of 8th World Congress on Intelligent Control and Automation (WCICA)*, Jinan, China, July, 2010, pp. 1776-1781.
2. **J. Hu**, L. Xie, and C. Zhang, "Diffusion Kalman Filtering Based on Covariance Intersection," in *Proceedings of 18th IFAC World Congress*, Milano, Italy, September, 2011, pp. 12471-12476.
3. **J. Hu**, L. Xie, and J. Xu, "Multi-agent Cooperative Target Search," *Accepted by 12th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Guangzhou, China, December, 2012.
4. J. Xu, **J. Hu**, L. Xie and K. Lum, "Distributed Coverage Control Under Generalized Locational Optimization Framework," *Accepted by 31st Chinese Control Conference (CCC)*, Hefei, China, July, 2012.

Bibliography

- [1] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, “Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet,” *ACM SIGOPS Operating Systems Review*, vol. 36, no. 5, pp. 96–107, 2002.
- [2] R. Van Dam, B. Borchers, J. Hendrickx, and S. Hong, “Soil effects on thermal signatures of buried nonmetallic landmines,” in *Proceedings of the Society of Photo-Optical Instrumentation Engineers Conference Series*, vol. 5089, 2003, pp. 1210–1218.
- [3] S. Ferrari, C. Cai, R. Fierro, and B. Perteet, “A geometric optimization approach to detecting and intercepting dynamic targets,” in *Proceedings of the American Control Conference*, 2007, pp. 5316–5321.
- [4] D. Spanos, R. Olfati-Saber, and R. Murray, “Approximate distributed Kalman filtering in sensor networks with quantifiable performance,” in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks*, 2005, pp. 133–139.
- [5] R. Olfati-Saber, “Distributed Kalman filter with embedded consensus filters,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 8179–8184.
- [6] —, “Distributed Kalman filtering for sensor networks,” in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 5492–5498.
- [7] H. Zhang, L. Xie, D. Zhang, and Y. Soh, “A reorganized innovation approach to linear estimation,” *IEEE Transactions on automatic control*, vol. 49, no. 10, pp. 1810–1814, 2004.
- [8] E. Burian, D. Yoerger, A. Bradley, and H. Singh, “Gradient search with autonomous underwater vehicles using scalar measurements,” in *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology*, 1996, pp. 86–98.
- [9] D. Hoskins, “Least action approach to collective behavior,” in *Proceedings of Society of Photo-Optical Instrumentation Engineers Conference*, vol. 2593, 1995, pp. 108–120.

- [10] P. Ogren, E. Fiorelli, and N. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.
- [11] 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, 2004.
- [12] M. Schwager and D. Rus, "Distributed coverage control with sensory feedback for networked robots," in *Proceedings of Robotics: Science and Systems*, 2006.
- [13] F. Cattivelli and A. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, 2010.
- [14] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [15] D. Li, K. Wong, Y. Hu, and A. Sayeed, "Detection, classification and tracking of targets in distributed sensor networks," *IEEE signal processing magazine*, vol. 19, no. 2, pp. 17–29, 2002.
- [16] J. Lin, W. Xiao, F. Lewis, and L. Xie, "Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 1886–1896, 2009.
- [17] Y. Zhai, M. Yeary, J. Havlicek, and G. Fan, "A new centralized sensor fusion-tracking methodology based on particle filtering for power-aware systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 10, pp. 2377–2387, 2008.
- [18] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient signal classification in ad hoc wireless sensor networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 1, pp. 190–196, 2008.
- [19] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
- [20] S. Dandach, B. Fidan, S. Dasgupta, and B. Anderson, "Adaptive source localization by mobile agents," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 2045–2050.
- [21] H. La and W. Sheng, "Moving targets tracking and observing in a distributed mobile sensor network," in *Proceedings of American Control Conference*, 2009, pp. 3319–3324.

- [22] I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: Formation of geometric patterns," *SIAM Journal on Computing*, vol. 28, p. 1347, 1999.
- [23] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 865–875, 2004.
- [24] E. Justh and P. Krishnaprasad, "Equilibria and steering laws for planar formations* 1," *Systems & Control Letters*, vol. 52, no. 1, pp. 25–38, 2004.
- [25] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.
- [26] H. Tanner, A. Jadbabaie, and G. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 863–868, 2007.
- [27] E. Klavins, R. Ghrist, and D. Lipsky, "A grammatical approach to self-organizing robotic systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 949–962, 2006.
- [28] V. Gazi and K. Passino, "Stability analysis of social foraging swarms: Combined effects of attractant/repellent profiles," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 3, 2002, pp. 2848–2853.
- [29] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *Control, Optimisation and Calculus of Variations*, vol. 11, no. 4, pp. 691–719, 2005.
- [30] J. Cortes and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM journal on control and optimization*, no. 5, pp. 1543–1574, 2006.
- [31] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," in *Proceedings of the The 43rd IEEE Conference on Decision and Control*, vol. 4, 2004.
- [32] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "Distributed memoryless point convergence algorithm for mobilerobots with limited visibility," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 818–828, 1999.
- [33] J. Lin, A. Morse, and B. Anderson, "The multi-agent rendezvous problem," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 2, 2003, pp. 1508–1513.
- [34] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, 2004.

- [35] J. Cortes, S. Martinez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, 2006.
- [36] V. Sharma, M. Savchenko, E. Frazzoli, and P. Voulgaris, "Time complexity of sensor-based vehicle routing," in *Proceedings of Robotics: Science and Systems*, 2005, pp. 297–304.
- [37] R. Olfati-Saber and R. 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.
- [38] W. Ren, R. Beard, and E. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.
- [39] C. Chong, S. Mori, and K. Chang, "Adaptive distributed estimation," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 26, 1987, pp. 2233–2238.
- [40] K. Chang, R. Saha, and Y. Bar-Shalom, "On optimal track-to-track fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1271–1276, 1997.
- [41] X. Li, K. Zhang, J. Zhao, and Y. Zhu, "Optimal linear estimation fusion. part v. relationships," in *Proceedings of the Fifth International Conference on Information Fusion*, vol. 1, 2002, pp. 497–504.
- [42] S. Mori, W. Barker, C. Chong, and K. Chang, "Track association and track fusion with nondeterministic target dynamics," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 659–668, 2002.
- [43] Y. Zhu, E. Song, Z. J., and Z. You, "Optimal dimensionality reduction of sensor data in multisensor estimation fusion," *IEEE Transactions on Signal Processing*, vol. 53, no. 5, pp. 1631–1639, 2005.
- [44] S. Marano, V. Matta, and P. Willett, "Distributed estimation in large wireless sensor networks via a locally optimum approach," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 748–756, 2008.
- [45] I. Schizas, G. Giannakis, and Z. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4284–4299, 2007.
- [46] F. Cattivelli, C. Lopes, and A. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, 2008.

- [47] H. Hashemipour, S. Roy, and A. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–94, 1988.
- [48] P. Alriksson and A. Rantzer, "Distributed Kalman filtering using weighted averaging," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006*.
- [49] F. Cattivelli and A. Sayed, "Diffusion distributed Kalman filtering with adaptive weights," in *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 2009.
- [50] S. Grime and H. Durrant-Whyte, "Data fusion in decentralized sensor networks," *Control engineering practice*, vol. 2, no. 5, pp. 849–863, 1994.
- [51] I. Schizas, G. Giannakis, S. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc wsns with noisy links-part ii: Distributed estimation and smoothing of random signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1650–1666, 2008.
- [52] S. Julier and J. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the American Control Conference*, vol. 4, 1997.
- [53] —, "General decentralized data fusion with covariance intersection (CI)," *Handbook of multisensor data fusion*, 2001.
- [54] J. Uhlmann, S. Julier, B. Kamgar-Parsi, M. Lanzagorta, and H. Shyu, "NASA Mars rover: A testbed for evaluating applications of covariance intersection," in *Proceedings of the Society of Photo-Optical Instrumentation Engineers Conference Series*, vol. 3693, 1999, p. 140.
- [55] P. Arambel, C. Rago, and R. Mehra, "Covariance intersection algorithm for distributed spacecraft state estimation," in *Proceedings of the American Control Conference*, vol. 6, 2002, pp. 4398–4403.
- [56] Y. Wang and X. Li, "A fast and fault-tolerant convex combination fusion algorithm under unknown cross-correlation," in *Proceedings of the 12th International Conference on Information Fusion*, 2009, pp. 571–578.
- [57] U. Khan and J. Moura, "Distributing the kalman filter for large-scale systems," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4919–4935, 2008.
- [58] F. Zhao and L. Guibas, *Wireless sensor networks: an information processing approach*. Morgan Kaufmann Pub, 2004.
- [59] A. Abbas and B. Jain, "Topology control in heterogeneous mobile ad hoc networks," in *Proceedings of the IEEE International Conference on Personal Wireless Communications*, 2005, pp. 47–51.

- [60] P. Gurumohan, T. Taylor, and V. Syrotiuk, "Topology control for manets," in *Proceedings of the IEEE Conference on Wireless Communications and Networking*, vol. 1, 2004, pp. 599–603.
- [61] M. Schwager, D. Rus, and J. Slotine, "Decentralized, adaptive coverage control for networked robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.
- [62] Y. Wang and I. Hussein, "Awareness coverage control over large-scale domains with intermittent communications," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1850–1859, 2010.
- [63] M. Schwager, B. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.
- [64] P. Ogren, E. Fiorelli, and N. Leonard, "Formations with a mission: Stable coordination of vehicle group maneuvers," in *Proceedings of the Symposium on Mathematical Theory of Networks and Systems*, 2002.
- [65] R. Bachmayer and N. Leonard, "Vehicle networks for gradient descent in a sampled environment (i)," in *Proceedings of the IEEE Conference on Decision and Control*, 2001, pp. 112–117.
- [66] J. Tan, "A scalable graph model and coordination algorithms for multi-robot systems," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2005, pp. 1529–1534.
- [67] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.
- [68] K. Lynch, I. Schwartz, P. Y., and R. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 710–724, 2008.
- [69] J. Choi, S. Oh, and R. Horowitz, "Distributed learning and cooperative control for multi-agent systems," *Automatica*, vol. 45, no. 12, pp. 2802–2814, 2009.
- [70] J. Cortés, "Distributed kriged Kalman filter for spatial estimation," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2816–2827, 2009.
- [71] K. Nygard, P. Chandler, and M. Pachter, "Dynamic network flow optimization models for air vehicle resource allocation," in *Proceedings of the American Control Conference*, vol. 3, 2001, pp. 1853–1858.
- [72] Y. Jin, A. Minai, and M. Polycarpou, "Cooperative real-time search and task allocation in UAV teams," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 1, 2003, pp. 7–12.

- [73] R. Beard, T. McLain, M. Goodrich, and E. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 911–922, 2002.
- [74] Y. Ben-Asher, S. Feldman, P. Gurfil, and M. Feldman, "Distributed decision and control for cooperative UAVs using ad hoc communication," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 3, pp. 511–516, 2008.
- [75] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part I: Fixed topology," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 2, 2003, pp. 2010–2015.
- [76] —, "Stable flocking of mobile agents part II: Dynamic topology," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 2, 2003, pp. 2016–2021.
- [77] L. Wang and F. Xiao, "Finite-time consensus problems for networks of dynamic agents," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 950–955, 2010.
- [78] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 19–33, 2008.
- [79] T. Arampatzis, J. Lygeros, and S. Manesis, "A survey of applications of wireless sensors and wireless sensor networks," in *Proceedings of the IEEE International Symposium on Intelligent Control*, 2005, pp. 719–724.
- [80] Y. Bar-Shalom, *Tracking and data association*. Academic Press Professional, Inc. San Diego, CA, USA, 1987.
- [81] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.
- [82] W. Xiao, J. Wu, and L. Xie, "Adaptive sensor scheduling for target tracking in wireless sensor network," in *Proceedings of the Society of Photo-Optical Instrumentation Engineers Conference Series*, vol. 5910, 2005, pp. 104–112.
- [83] C. Lin, W. Peng, and Y. Tseng, "Efficient in-network moving object tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, pp. 1044–1056, 2006.
- [84] Y. Toh, W. Xiao, L. Xie, and T. Yang, "A wireless sensor network target tracking system with distributed competition based sensor scheduling," in *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, 2007, pp. 257–262.

- [85] R. Olfati-Saber, "Distributed tracking for mobile sensor networks with information-driven mobility," in *Proceedings of the American Control Conference*, 2007, pp. 4606–4612.
- [86] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 41–53, 2005.
- [87] S. Nardone, A. Lindgren, and K. Gong, "Fundamental properties and performance of conventional bearings-only target motion analysis," *IEEE Transactions on Automatic Control*, vol. 29, no. 9, pp. 775–787, 1984.
- [88] A. Bishop, B. Fidan, B. Anderson, K. Dogancay, and P. Pathirana, "Optimality analysis of sensor-target localization geometries," *Automatica*, vol. 46, no. 3, pp. 479–492, 2010.
- [89] A. Bishop and P. Jensfelt, "An optimality analysis of sensor-target geometries for signal strength based localization," in *Proceedings of the 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2009, pp. 127–132.
- [90] K. Dogancay and H. Hmam, "Optimal angular sensor separation for AOA localization," *Signal Processing*, vol. 88, no. 5, pp. 1248–1260, 2008.
- [91] I. Kadar, "Optimum geometry selection for sensor fusion," in *Proceedings of SPIE Conference on Signal Processing, Sensors Fusion and Target Recognition VII*, vol. 3374, 1998, p. 96.
- [92] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.
- [93] K. Huguenin and J. Rendas, "An architecture for distributed collaborative adaptive sensing," in *Proceedings of the International Symposium on Unmanned Untethered Submersibles Technology*, 2007.
- [94] S. Tully, H. Moon, G. Kantor, and H. Choset, "Iterated filters for bearing-only SLAM," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2008, pp. 1442–1448.
- [95] B. Bell and F. Cathey, "The iterated kalman filter update as a gauss-newton method," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.
- [96] D. Li and Y. Hu, "Energy-based collaborative source localization using acoustic microsensor array," *EURASIP Journal on Applied Signal Processing*, vol. 2003, pp. 321–337, 2003.

- [97] D. Blatt and A. Hero, "Energy-based sensor network source localization via projection onto convex sets," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3614–3619, 2006.
- [98] Q. Shi and C. He, "A new incremental optimization algorithm for ml-based source localization in sensor networks," *IEEE Signal Processing Letters*, vol. 15, pp. 45–48, 2008.
- [99] C. Meesookho, U. Mitra, and S. Narayanan, "On energy-based acoustic source localization for sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 365–377, 2008.
- [100] X. Sheng and Y. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 1, pp. 44–53, 2005.
- [101] R. Vidal, O. Shakernia, H. Kim, D. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, 2002.
- [102] V. Isler, S. Kannan, and S. Khanna, "Randomized pursuit-evasion in a polygonal environment," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 875–884, 2005.
- [103] T. Chung, J. Burdick, and R. Murray, "A decentralized motion coordination strategy for dynamic target tracking," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 2416–2422.
- [104] M. Polycarpou, Y. Yang, and K. Passino, "A cooperative search framework for distributed agents," in *Proceedings of the IEEE International Symposium on Intelligent Control*, 2001, pp. 1–6.
- [105] L. Bertuccelli and J. How, "Robust UAV search for environments with imprecise probability maps," in *Proceedings of the IEEE Conference on Decision and Control*, 2005, pp. 5680–5685.
- [106] Y. Yang, A. Minai, and M. Polycarpou, "Decentralized cooperative search by networked UAVs in an uncertain environment," in *Proceedings of the American Control Conference*, vol. 6, 2004, pp. 5558–5563.
- [107] L. Stone, *Theory of optimal search*. Academic Press New York, 1975.
- [108] L. Bertuccelli and J. How, "UAV search for dynamic targets with uncertain motion models," in *Proceedings of the IEEE Conference on Decision and Control*, 2007, pp. 5941–5946.
- [109] Y. Yang, M. Polycarpou, and A. Minai, "Multi-UAV cooperative search using an opportunistic learning method," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, p. 716, 2007.

- [110] J. Berger and J. Happe, “Co-evolutionary search path planning under constrained information-sharing for a cooperative unmanned aerial vehicle team,” in *Proceedings of IEEE Congress on Evolutionary Computation*, 2010, pp. 1–8.
- [111] P. Millet, D. Casbeer, T. Mercker, and J. Bishop, “Multi-agent decentralized search of a probability map with communication constraints,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2010.
- [112] M. Zhong and C. Cassandras, “Distributed coverage control and data collection with mobile sensor networks,” in *Proceedings of the IEEE Conference on Decision and Control*, 2010, pp. 5604–5609.
- [113] S. Saripalli, J. Montgomery, and G. Sukhatme, “Visually guided landing of an unmanned aerial vehicle,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 371–380, 2003.
- [114] W. Niehsen, “Information fusion based on fast covariance intersection filtering,” in *Proceedings of the 5th International Conference on Information Fusion*, vol. 2, 2002.
- [115] D. Franken and A. Hupper, “Improved fast covariance intersection for distributed data fusion,” in *Proceedings of the 8th International Conference on Information Fusion*, 2005, p. 7.
- [116] B. Anderson and J. Moore, “Detectability and stabilizability of time-varying discrete-time linear systems,” *SIAM Journal on Control and Optimization*, vol. 19, no. 1, pp. 20–32, 1981.
- [117] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *Proceedings of the 4th international symposium on Information processing in sensor networks*, 2005, p. 9.
- [118] F. Araújo and L. Rodrigues, “Fast localized delaunay triangulation,” *Principles of Distributed Systems*, pp. 81–93, 2005.
- [119] E. Nettleton, “Decentralised architectures for tracking and navigation with multiple flight vehicles,” Ph.D. dissertation, The University of Sydney, 2003.
- [120] H. Durrant-Whyte and T. Henderson, “Multi sensor data fusion,” *Springer Handbook of Robotics*, pp. 585–610, 2008.
- [121] J. Bernardo, A. Smith, and M. Berliner, *Bayesian theory*. Wiley New York, 1994, vol. 62.
- [122] R. Gallager, *Discrete stochastic processes*. Springer, 1996.
- [123] F. Jensen, *An introduction to Bayesian networks*. UCL press London, 1996, vol. 74.

- [124] F. Bullo, J. Cortés, and S. Martinez, *Distributed control of robotic networks*. Princeton University Press, 2009.
- [125] R. Horn and C. Johnson, *Matrix analysis*. Cambridge University Press, 1990.
- [126] S. Hansen, T. McLain, and M. Goodrich, “Probabilistic searching using a small unmanned aerial vehicle,” *Proceedings of AIAA Infotech@ Aerospace*, 2007.
- [127] R. Ash and C. Doleans-Dade, *Probability and measure theory*. Academic Press, 2000.
- [128] J. Hu, L. Xie, K. Lum, and J. Xu, “Multi-agent information fusion and cooperative control in target search,” *To appear on IEEE Transactions on Control Systems Technology*, 2012.
- [129] K. Petersen and M. Pedersen, “The matrix cookbook,” *Technical University of Denmark*, pp. 7–15, 2008.
- [130] E. Games, “Unreal engine 3,” [http://www.linuxfoundation.org/p ... rneldevelopment.php](http://www.linuxfoundation.org/p...rneldevelopment.php), 2006.
- [131] G. Johnson, *LabVIEW graphical programming: practical applications in instrumentation and control*. McGraw-Hill School Education Group, 1997.
- [132] A. Varga, “The OMNeT++ discrete event simulation system,” in *Proceedings of the European Simulation Multiconference*, vol. 9, 2001.
- [133] J. Taylor, *Introduction to ultra-wideband radar systems*. CRC, 1995.
- [134] GAUI, “330X-S quad-flyer instruction manual,” <http://www.gauai.com.tw/html/download.asp>.
- [135] J. Bondy and U. Murty, *Graph theory with applications*. MacMillan London, 1976, vol. 290.
- [136] N. Biggs, *Algebraic graph theory*. Cambridge Univ Pr, 1993.
- [137] C. Godsil and G. Royle, *Algebraic graph theory*. Springer New York, 2001, vol. 8.
- [138] R. Bellman, *Introduction to matrix analysis*. Society for Industrial Mathematics, 1997, vol. 19.
- [139] A. Okabe, B. Boots, K. Sugihara, and S. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*. Wiley & Sons Chichester., 1992.
- [140] F. Aurenhammer, “Voronoi diagrams a survey of a fundamental geometric data structure,” *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.