UNIVERSITY OF CALIFORNIA,
IRVINE

Distributed Dynamic Tracking: Multi-Agent Leader-Following and Targets Coverage

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mechanical and Aerospace Engineering

by

Yi-Fan Chung

Dissertation Committee:
Professor Solmaz Kia, Chair
Professor Faryar Jabbari
Professor Haithem Taha

2020

DEDICATION

To my family

TABLE OF CONTENTS

LIST OF FIGURES

List of Algorithms

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Professor Solmaz Kia, for her continuous support and endless patience. I learned so much from her about the expertise and positive attitude to the research. She encouraged me to explore the knowledge while guided me with valuable insights and constructive comments. I also received great help from her in academic writing. She taught me the proper way to write an academic paper and the rigorous logic to present our research achievements. Without her, I would not have been able to complete this work. Her mentoring not only assisted me in achieving my Ph.D. but also trained me to be a better researcher.

I would like to express my gratitude to Professor Faryar Jabbari, Professor Haithem Taha, Professor Tryphon Georgiou and Professor Marco Levorato for serving as my committee members and share their insights and expertise on my research. I would also like to thank Professor A. Lee Swindlehurst for his generous guidance and advice about my research's development in the wireless communication field.

I would like to thank my program sponsor, National Chung-Shan Institute of Science and technology, for supporting the funding through the four years and giving me this valuable opportunity to study aboard.

I would acknowledge the support of my research colleagues, Jianan Zhu, Hossein Moradian, Navid Rezazadeh, Amir-Salar (Noah) Esteki and Changwei Chen. I would also like to thank all my friends, especially Rui Fu, Anqi Dong, Joseph Hsieh, my snowboarding friends, for making the life at Irvine a great and enjoyable experience.

Finally, I would like to express my sincere thanks to my family for their love and support through all the years. I end the long list of acknowledgments with the special person in my life, Hsiao-Yuan Huang. Thank you for all you do for me.

Chapter 2 is taken, in part, from the papers published as "Distributed Minimum Energy Leader-Follower Algorithm for Multi-Agent Systems with an Active Non-Homogenous Leader" by Yi-Fan Chung and Solmaz S. Kia in 2019 18th European Control Conference (ECC), IEEE, 2019; as well as, "Distributed leader following of an active leader for linear heterogeneous multi-agent systems" by Yi-Fan Chung, and Solmaz S. Kia in Systems & Control Letters 137 (2020): 104621. The dissertation author is the primary investigator and author of these papers.

Chapter 3 is taken, in full, from the papers published as "Distributed dynamic containment control over a strongly connected and weight-balanced digraph" by Yi-Fan Chung and Solmaz S. Kia in IFAC Workshop on Distributed Estimation and Control in Networked Systems, IFAC-PapersOnLine 52.20 (2019): 25-30. The dissertation author is the primary investigator and author of this paper.

Chapter 4 is taken, in part, from the paper "Dynamic active average consensus" by Yi-Fan Chung and Solmaz S. Kia, that is accepted for publication in IEEE Control Systems Letters,

2020. The dissertation author is the primary investigator and author of this paper.

Chapter 5 is taken, in part, from the papers as "A distributed service-matching coverage via heterogeneous mobile agents" by Yi-Fan Chung and Solmaz S. Kia, prepared to submit to Automatica, 2021. The dissertation author is the primary investigator and author of this paper.

CURRICULUM VITAE

Yi-Fan Chung

EDUCATION

Ph D. student, in Mechanical and Aerospace Engineering **2016-2020**
University of California, Irvine *Irvine, U.S.A*

M.S., in Mechanical Engineering **2012**
National Cheng Kung University *Tainan, Taiwan*

B.S., in Mechanical Engineering **2010**
National Cheng Kung University *Tainan, Taiwan*

RESEARCH EXPERIENCE

Assistant Technician and Researcher **2012–2016**
National Chung-Shan Institute of Science & Technology *Taoyuan, Taiwan*

PUBLICATIONS

Yi-Fan Chung, Solmaz Kia,"Distributed minimum energy leader-follower algorithm for multi-agent systems with an active non-homogenous leader", *European Control Conference*, Naples, Italy, 2019.

Yi-Fan Chung, Solmaz S. Kia,"Distributed dynamic containment control over a strongly connected and weight-balanced digraph", *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, Chicago, IL, 2019.

Yi-Fan Chung, Solmaz S. Kia,"Distributed leader following of an active leader in formation for heterogeneous multi-agent systems", *Systems & Control Letters*, 137 (2020): 104621.

Yi-Fan Chung, Solmaz S. Kia,"Dynamics active average consensus", *IEEE Control Systems Letters*, 2020 (accepted for publication).

Yi-Fan Chung, Solmaz S. Kia,"Matching based coverage control for mobile heterogeneous agents deployment", *2021 American Control Conference (ACC)*, under preparation, to be submitted on Sept. 15th.

Yi-Fan Chung, Solmaz S. Kia,"A distributed service-matching coverage via heterogeneous mobile agents", *Automatica*, under preparation, to be submitted on Sept. 15th.

ABSTRACT OF THE DISSERTATION

Distributed Dynamic Tracking: Multi-Agent Leader-Following and Targets Coverage

By

Yi-Fan Chung

Doctor of Philosophy in Mechanical and Aerospace Engineering

University of California, Irvine, 2020

Professor Solmaz Kia, Chair

With the advances in low-cost reliable electronic devices, autonomous multi-agent systems play an important role in a wide range of applications, such as sensing networks, smart grid, smart transportation, and exploration in hazardous situations. In recent years, factors such as avoiding a single failure point, demand for privacy preservation, and opting for lower computation and communication costs have created the expectation that the autonomous multi-agent systems should be operated in a distributed manner with no central control. In many operations, coordinating tasks among autonomous multi-agent systems involve some forms of distributed leader-following problems. That is, it is expected that a group of networked autonomous agents, normally referred to as followers, should use their local information and local interactions with their neighboring agents to determine their actions so that the entire network achieves a desired system-level behavior that depends on the state(s) of single or multiple leaders. When the dynamics of the leader(s) is unknown, e.g., in target tracking problems, and only a subset of the agents can measure the state(s) of the leader(s) online, the limited information increases the challenge to meet leader-following objectives. The focus of this dissertation is providing practical leader-following solutions that require the least possible assumptions on the dynamics of the leader(s). More specifically, we consider four types of leader-following objectives.

The first problem addressed is a *single leader-following problem* for a group of heterogeneous linear time-invariant followers, where a subset of the followers has access to the state of an unknown leader in only specific sampling times. We propose a distributed control that uses a minimum-energy control framework to enable the followers to arrive at the sampled state of the leader by the time the next sample arrives. The next problem we consider is a *containment control problem*, a leader-following problem with multiple leaders, where a group of mobile agents aim to stay in the convex hull spanned by a group of moving leaders with unknown dynamics. Our proposed distributed control enables a group of networked unicycle mobile agents to track the convex hull of the leaders. This algorithm requires the agents to only communicate with each other in discrete-time fashion. The innovation in our containment control design is to model the problem in the form of an *active average consensus problem*, for which we also propose a novel distributed solution both in continuous-time and discrete-time form. Active average consensus by itself constitutes the third leader-following problem that we study, in which a set of networked agents aim to track the average of the dynamic signals measured by the active agents. The fourth leader-following problem we consider in this dissertation is a *coverage problem* via a set of mobile agents for a group of dense dynamic targets whose distribution in the space is not known a priori. For this problem, we propose a novel distributed coverage control that first uses a distributed estimation process to enable all the agents to obtain an estimate of the targets' distribution when only a subset of agents in the network can observe the targets. Then, we develop a distributed deployment solution that enables the agents to re-position themselves in a way that their collective quality of service (QoS) distribution is in close accordance with the estimated density distribution of the targets. In our setting, the agents are heterogeneous in the sense that their spatial QoS distribution is different. We demonstrate our results for event detection via sensor networks and UAV-aided wireless communication coverage problems.

Chapter 1

Introduction

1.1 Background and Motivation

Researchers have long noticed different forms of coordinated behaviors in animal groups in nature that help the group members to meet objectives that benefit everyone but are beyond the capability of individual members. For example, fish school to defend against predators, wolves hunt in a pack, birds flock when migrating or ants swarm to collect food. Inspired by these natural phenomena and the technological advances in miniaturizing low-cost reliable devices for computing, communication, sensing, and actuation, engineering coordinated behaviors in multi-agent network systems to enhance capabilities of dynamical systems have attracted a lot of attention by researchers in recent years.

Multi-agent systems have several advantages over a single agent, including robustness to individual agents' failures and the ability to perform challenging tasks that cannot be achieved by a single agent. The potential applications are, for example, monitoring forest, tracking wildlife, border patrol, surveillance, and reconnaissance. To enable these applications, various coordinated control designs need to be developed such that agents in such networks

can achieve system-level objectives. The centralized control scheme, which gathers all of the information in a single place, performs the computation, and then sends the solution back through the network to each agent, is the straight forward solution. Although simple, the centralized approach has numerous drawbacks [1]: 1) it is fragile because the failure of the central agent causes the shutdown of the entire system, 2) it is not scalable because the amount of communication and computation increases with the size of the network, 3) the data from each agent needs a unique identifier to avoid double-counting, 4) the delay of data transmission and computation time grows with the size of the network, and 5) the data is lacking privacy because the information from each agent is exposed over the entire network. This motivates the interest in developing distributed solutions for multi-agent systems. For distributed cooperative control, agents in a network are required to use only local information to determine their own decisions such that the whole network achieves the prescribed system-level behavior.

Consensus problem of multi-agent systems is an important component of many cooperative control problems, such as rendezvous [2], formation control [3], flocking control [4], containment control [5] and sensor networks [6]. Consensus problems can be roughly categorized into leaderless and leader-following. In the leaderless case, the agents aim to reach to a static or dynamic agreement on a common value [1, 7, 8]. The most famous leaderless consensus problem is the average consensus problem where the system-level objective is to all agents to agree on the average value of the reference signals at each agent. On the other hand, leader-following is a more challenging problem. The agents (which are usually referred to as followers) aim to reach a desired system-level behavior which depends on the state(s) of the leader(s). It is common in the literature to consider the leader(s) as an agent(s) in the multi-agent system so the leader(s) would be with a certain known dynamics and willing to share its (their) states or even coordinating its (their) motion(s) with the follower agents. However, the concept of a leader(s) can be further extended to an uncooperative target(s), an unknown object(s), or even a virtual signal(s) that the agents aim to track. In these

scenarios, the dynamics of the leader(s) may be unknown to the agents and only a subset of the agents can measure the state(s) of the leader(s) online. Hence, the limited information increases the challenge to achieve the leader-following objective.

This dissertation investigates four different leader-following scenarios. Chapter 2 considers the single leader-following problem, e.g., a single target tracking. The objective is the agents to follow the state of a leader whose dynamics is unknown to them. The only information available about the leader is its sampled state that is collected by a subset of agents in the network. The potential application is, for example, the formation control for UAVs to follow a target (see Fig. 1.1). Chapter 3 considers the containment control problem, which is a multiple leader-following problem. In the Containment control problem agents want to stay in the convex hull spanned by the leaders. A prime application example for containment problem, see Fig. 1.2, is when a group of communicating robots follows a group of leader robots that can avoid obstacles when they transport through a hazardous area [9]. Other potential applications include formation control for UAVs [10] and underwater vehicles [11], hazardous material delivery [12] and mobile sensor networks [13]. Chapter 4 considers an extension of dynamic average consensus problem which is referred to as dynamic *active* average consensus. In this problem, only a subset of the agents are the active agents that collect measurements. The objective then is for all active and passive agents to track the (weighted) average of the collected measurements, see Fig. 1.3. The scenario that contains the active and passive agents is prevalent in a multi-agent system, for example, the underwater exploration as shown in Fig. 1.4. Although dynamic average consensus problem is usually categorized into the leaderless class, it also can be considered as a leader-following problem when the local reference signal at each agent is constructed by the leaders measured by the agent. The agents aim to follow the average of the reference signals corresponding to the measured leaders. In Chapter 4, we also show that the containment control problem can be formulated as an active average consensus problem and solved in a efficient way using our proposed dynamic active average consensus algorithm. Chapter 5

Figure 1.1 – A leader-following application in formation control. (Retrieved from DARPA)

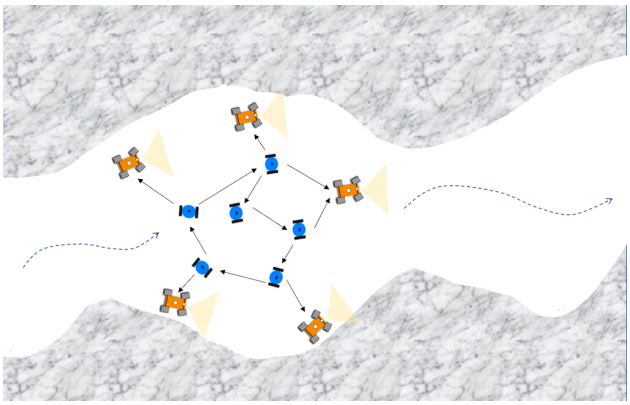

Figure 1.2 – A containment control application in transporting robots through a hazardous area.

considers the coverage control for mobile agent deployment to track and provide service for a group of dense targets. Coverage control is a deployment strategy and we use these two terms interchangeably in the context. The purpose of coverage control is to deploy a group of agents to provide their service (monitoring, data collection, wireless communication, etc.) to cover an area occupied by targets. This problem is in leader-following framework because the agents' deployment depends on the locations of the targets (leaders) they aim to cover. Some important applications investigated in literature include wildfire surveillance [14], water quality monitoring [15], environmental boundary monitoring [16], and UAVs wireless network coverage [17] (see Fig. 1.5).

The following section reviews the literature about leader-following control, containment con-

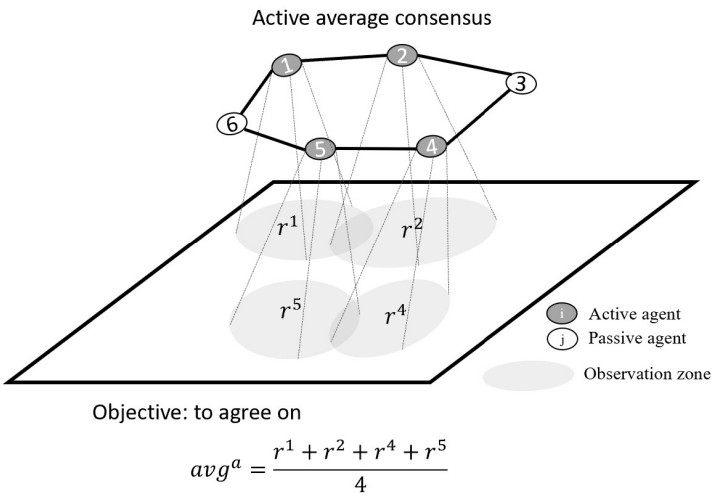

Figure 1.3 – The active average consensus problem: Agent 1,2,4 and 5 are active agents which have local measured signal; Agent 3 and 6 are passive agents which do not measure a useful signal or do not have the ability to measure. The objective of the active average consensus is to track $\frac{r^1+r^2+r^4+r^5}{4}$.

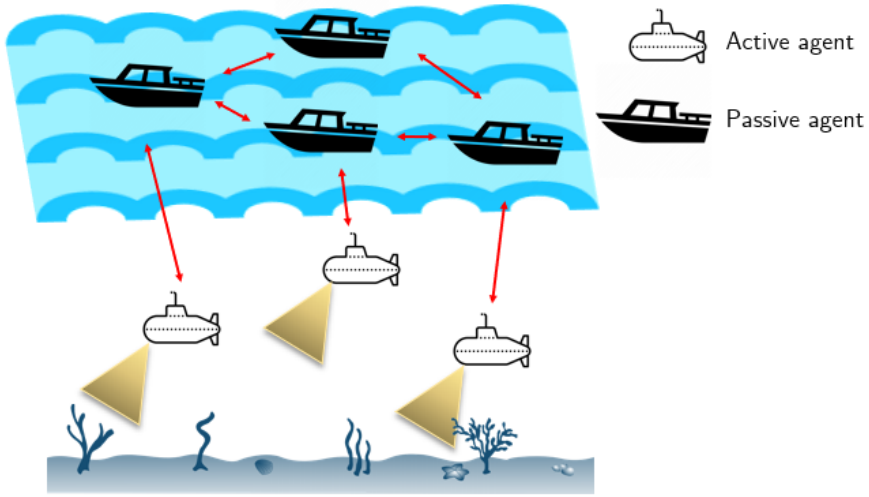

Figure 1.4 – A underwater exploration scenario contains active agents (submarines), which can detect the objects of interest, and passive agents (boats).

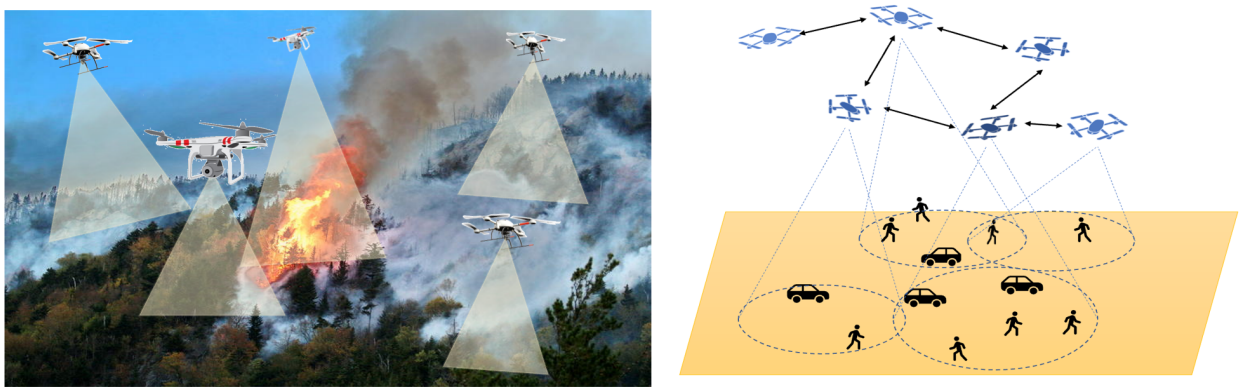

Figure 1.5 – The applications of mobile agents deployment in (a) wildfire monitoring and (b) wireless network coverage.

trol, dynamics average consensus control, and coverage control.

1.2 Literature Survey

1.2.1 Distributed Leader Following Control

The leader-following algorithms for the single integrator and double integrator dynamics are presented in [18], and for homogeneous LTI systems are proposed in [19] and [20]. For systems constituted of heterogeneous LTI followers, [21] and [22] propose the algorithms to synchronize with a passive zero-input LTI leader. [23] and [24] develop the controls for the single and double integral systems, respectively, to track an active leader (the active leader is a leader that has a control input). But their works assume the leader's control input is available to all the followers. [25] and [26] propose a leader-following algorithm respectively for homogeneous LTI and heterogeneous nonlinear MASs in which the unknown input of the leader is bounded and is not available to any follower. But the control inputs in [25] and [26] have the sliding mode structure and suffer from the well-known undesirable chattering behavior. We recall that from a practical perspective, chattering is undesirable

and leads to excessive control energy expenditure [27].

[28] is the recent result for the leader-following problem, which is based on the result of [25] and develops a distributed observer to estimate the leader's state for each follower. Then, the output synchronization of heterogeneous leader-follower linear systems is achieved by optimal local tracking of the output of the observer. We note that in both [25] and [28], the active leader is restricted to be linear and have limited input.

The work reviewed so far are all converge to leader following in an asymptotic manner, i.e., the settling time to reach an agreement is infinity. For fast convergence, [29], [30] and [31] propose the finite-time synchronization algorithms for single and double integral MASs, where the upper bound of the settling time explicitly depends on the initial state of the MAS. Therefore, to use these algorithms, the centralized knowledge of the initial state of the MAS is essential to estimate the settling time. [32] and [33] propose the fixed-time synchronization algorithms, where the settling time is bounded and independent of the initial state of the MAS. However, for both these finite and fixed-time algorithms, the settling time is upper bounded by a conservative estimation. [34] introduces the specified-time synchronization control for the leaderless MASs in which one can determine the settling time exactly in advance. Specified-time synchronization can be useful to the applications that require precise acting time, such as target attack at a specified time.

1.2.2 Distributed Containment Control

Containment control has been of interest in the literature in recent years. [35] proposes control for a group of unicycle agents that drives the leaders to a formation and the followers to the convex hull. [36] design a containment control for single integrator agents based on the theory of partial differential equations and a stop-go policy for the moving leaders. For double integrator agents, the containment control problem is considered in [37] and [38]. The

algorithms proposed in [35–38] require the communication topology of the followers to be an undirected graph.

In some situations, the interaction topology among agents may be a directed graph due to realistic communication restrictions. Accordingly, [39] extends the work of [36] and focus on switching directed interaction topology among agents. The agents with more complex dynamics under directed interaction graphs are investigated in [40] for general linear dynamics and [41] for nonlinear Lagrangian dynamics, respective.

Nevertheless, the work mentioned so far need to continuously exchange information among the network, which may not be realistic in practice. In [42] a group of agents with discrete-time dynamics is considered and the containment control problem is solved by a discrete-time scheme of hybrid model predictive control. However, it is preferable to having control for continuous-time dynamics agents with discrete-time communications with their neighbors. [9] propose a containment control based on periodic sampled-data for agents with continuous-time single and double integrator dynamics over a directed graph. Furthermore, aperiodic sampled-based containment controls for double integrators and continuous-time linear agents are studied by [43] and [44], respectively.

It is worth mentioning that all of the work mentioned above is considering the homogeneous network systems, that is all agents are with identical dynamics. [45] consider a heterogeneous multi-agent system that the leader group and the follower group can be with either single or double integrator dynamics, respectively. [46] study a more general case that the follower agents can be with different linear dynamics and the leader group has the same passive linear dynamics (linear dynamics without input). However, for the work of [45] and [46], the leaders are still homogeneous and they follow a certain dynamics that are assumed to be known to the followers.

1.2.3 Distributed Dynamics Average Consensus Control

Static average consensus, where agents aim to reach consensus on the average of the agents' initial static values, is one of the most popular distributed algorithms in coordination of network systems [7,18,47–49]. *Dynamic average consensus*, initially discussed in [50], on the other hand, considers the problem of achieving agreement on the average of the local time-varying signals of each agent. Designing dynamic average consensus algorithms that achieve tracking with zero error is challenging. This due to the time that it takes for any agent level information to propagate through the network. Thus, there are various algorithms in the literature that try to address this problem with small tracking error. In [51], the authors propose a proportional-integral dynamic average consensus algorithm that, from any initial condition, converges with a bounded steady state error, which vanishes if the signals are static. However, this algorithm reacquires the reference signal of the agents to be bounded and the graph topology to be connected. On the other hand, [52] proposes a dynamic average consensus algorithm that works over strongly connected and weight-balanced digraphs and only requires the rate of change of the agents' reference signal to be bounded. The steady-state error of this algorithm is controllable using a parameter in the algorithm, however, this algorithm requires special initialization and thus it is not robust to agents leaving and then coming back to the group. To eliminate the tracking error, [53] extends the work of [51] by making the assumption that a priori information about the model of the reference signal of the agents is available. More specifically, [53] uses the internal model principle built around a certain class of time-varying reference signals whose Laplace transform is a rational function with no poles in the left-hand complex plane. On the other hand, [54–56] proposes average consensus algorithms with zero steady-state error tracking for arbitrary reference signals with known bound on the signal value or its derivatives. However, these algorithms use a sliding mode like components and, thus suffer from the well-known chattering problem for sliding mode controllers. Recall that from a practical perspective, chattering is undesirable

and leads to excessive control energy expenditure.

Dynamic active average consensus is an extension of a dynamic average consensus problem. In dynamic active average consensus, a group of agents interacting over a connected graph should track the average of their 'measured' local signals. However, at any time, only a subset of these agents are active, meaning that only a subset of agents collects measurements. The objective then in the active average consensus is for all agents, both active and passive, to obtain the average of the collected measurements, which is the sum of the collected measurements divided by the numbers of the measurements (or equivalently numbers of the active agents). The conventional average consensus problem reviewed above is in fact a special case of the *active average consensus* with all the agents being active at all times. The active average consensus problem can be also viewed as a *weighted average consensus problem* [7], in which the weights are 1 for active agents and 0 for passive agents. However, the solutions for weighted average consensus (see e.g., [7, 57, 58]) use the notation of the 'equivalent' Laplacian matrix, which is the multiplication of the inverse of the weight matrix and the Laplacian matrix, so that the aggregate vector of the weights becomes the left eigenvector corresponding to the zero eigenvalue of the equivalent Laplacian [7]. Therefore, the weights should be non-zero, and thus these solutions cannot solve the active average consensus problem. Solutions specifically addressing the active average consensus problem are proposed in [59–61], but, they require both the reference inputs and their derivatives to be bounded to guarantee bounded error tracking. [59, 60] also assume that the active and passive role of the agents are fixed and agents cannot alternate between modes. On the other hand, [61] allows the agents to change mode but requires that this change be smooth.

1.2.4 Distributed Coverage Control for Mobile Agents Deployment

Over the last decade, deploying a group of networked mobile agents to cover a region with a service objective such monitoring, data collection, and wireless communication have attracted considerable attention, see for a few examples [14–16]. The deployment strategy commonly includes partitioning the environment and allocating agents to those partitions. That is, the area of interest is partitioned into subregions and each agent is allocated to a location in the subregion such that some coverage metric is optimized. The classic Voronoi-based deployment strategy [62–77] is a prime example of multi-agent deployment for area coverage. [62] as one of the initial work in this area develops a deployment algorithm based on the Llyod method to compute the Voronoi partition and allocate the agents to the Centroidal Voronoi configuration which is well-known as the optimal configuration of a class of locational optimization cost function [63]. Based on this framework, similar algorithms are also developed by [64, 65] considering communication constraints, [66] for non-convex environments and [67] for discovering the coverage holes and increasing coverage.

The original Voronoi-based deployment strategy is developed based on the assumption that the agents are homogeneous. To reach the optimal coverage with heterogeneous agents whose service (sensory) capabilities are different, [68–70] employs the weighted Voronoi diagram (power diagram) where the weightings account for heterogeneity among the agents. The heterogeneity of agents is also addressed in [71] where the authors present a new locational cost function that encodes the different service capabilities through heterogeneous density functions. The works mention above assume the footprint of the service provided by an agent is disk-shaped, i.e., the distribution of Quality of service (QoS) is isotropic. However, an anisotropic service model is more realistic because sensory systems such as cameras, directional antenna, and radars are anisotropic. [72, 73] for wedge-shape and [74] for elliptic footprint adapt an anisotropic service model by modifying the partitioning diagram to match

the features of the anisotropy of the sensors. But these methods increase the complexity of the deployment design, which make the design of distributed optimal deployment strategies very challenging.

The heterogeneity in deployment algorithm design can also be due to non-uniformity in area of interest. To deal with such scenarios, a priority (sensory) function of the position is introduced to indicate the importance level over the area, where a location needs higher QoS if the value at this location is higher. The work [62–74] mentioned above assume the priority function is known to each agent. However, it is unrealistic to assume this function is a priori in many scenarios. For example, in deploying agents to clean up an oil spill, the spatial distribution of the oil spill is the priority function, but the distribution of the concentration is not known in advance. [75] uses the parameterized basis functions to model the priority distribution, and [76] models the distribution by a zero-mean Gaussian random field. Then, in both [75] and [76], the agents gradually fitting their model to the true distribution from local sensor measurement while exploring the area. In [77], the authors assume the unknown priority function is a function of the position of some unknown targets. The search agents detecting the targets while exploiting the area, and then, broadcast their information about the environment to the service agents so the service agents can focus on the deployment problem.

1.3 Objectives and Contributions

The contributions of this dissertation lie in providing practical solutions to overcome some challenges and relax some of the theoretical restrictive assumptions on the design of distributed solutions for leader-following problems for a group of networked cooperative autonomous agents. This dissertation investigates four leader-following scenarios in the consecutive chapters. The objectives and contributions of each chapter are stated in what

follows.

- Chapter 2: Distributed single leader-following control

  We consider a leader-following problem in which the only information available about the leader is its instantaneous sampled state that is known only to a subset of a group of homogeneous or heterogeneous LTI followers at the sampling times. We make no assumptions about the input of the leader or the structural form of its dynamics. That is, the state of the leader is perceived by the followers as an exogenous signal. The sampled states of the leader can be the states of a physical system (e.g., in a pursuit-evasion problem) or a set of desired reference states of a virtual leader (e.g., in a waypoint tracking problem). Given the limited information about the leader, we seek a practical solution that enables the followers to arrive at the sampled state of the leader before the next sampling time. That is, we design a distributed algorithm that steers a group of homogeneous or heterogeneous LTI followers to be at the sampled states of the leader at a finite time just before the next sampled state is obtained. We note that practical one step lagged tracking has also been used in [78–80] for a set of dynamic average consensus algorithms with asymptotic tracking behavior. Our solution is inspired by the minimum energy controller design [81] in the classical optimal control theory, and is proposed for problems where the interaction topology of the followers plus the leader is an acyclic digraph with the leader as the global sink. Directed acyclic interaction topology can be interpreted as the agents only obtaining information from those in front of them (see, [82, 83] for algorithms designed over acyclic graphs). Our algorithm also allows the followers to track the sampled state of the leader with a locally chosen offset, which can be time-varying, to form a formation. This offset, when the followers are mobile agents and their whole state or part of it is the position vector, can be used to enable the followers to form a transnational invariant formation [84] about the sampled state of the leader. For a special class

of non-homogeneous LTI MAS, we show that our results can be extended to solve a leader-following problem where we want only an output of the followers to follow the leader's sampled state. Finally, if the followers are homogeneous, our algorithm not only results in a leader following behavior but also it makes the states and inputs of the followers fully synchronized after the first sampling epoch.

The contributions of the proposed leader following algorithm is summarized as follow:

  - We propose a practical leader following solution to track a leader with unknown dynamics and input.
  - The proposed leader following algorithm adapts to both homogeneous and heterogeneous group of LTI followers.
  - The algorithm steers the followers to be at the sampled states of the leader at the specified arrival times and in a specified formation.
  - The algorithm is a minimum energy control.

- Chapter 3: Distributed containment control.

  We consider the problem of the distributed containment control for a group of mobile agents, which communicate over a strongly connected and weight-balanced directed graph. The objective is to drive a group of mobile agents into a region that is enclosed by a group of moving leaders with unknown dynamics. In the literature, to provide perfect tracking, the containment controls normally assume that the leaders are static or if they are dynamic they either follow a certain dynamics that are known to the followers or the leaders' motions have to be coordinated with the followers (for example, in [36] the leaders have to execute the stop-go rule to "wait" for the followers). We consider the tracking problems where the position of the leaders is only measured online. We make no assumption about the dynamics of the leaders (so the leaders can be heterogeneous) except that the change of the state of the leaders is bounded. Under this setting, we provide a practical solution to let the mobile agents track the convex

hull of the leader with a bounded tracking error. This tracking error is expected, similar to what is known in dynamic consensus literature, as the online time-varying information takes some time to propagate through the network [1]. In our design, the agents jointly detect the group of moving leaders in a periodic sampling time. Each agent detects a subset (could be empty) of the leaders and computes the geometric centers of the subset. We show that the average of the geometric centers of the observed leaders at each agent is a point in the convex hull of the leaders. Then, we design a discrete-time consensus-based containment control algorithm to track this average. Hence, even though some of the agents do not observe any leaders, they still can track the average which is in the convex hull. In our setting, in accordance with the practical physical features of the problem, the mobile agents have continuous-time unicycle dynamics but they only need to communicate with each other and detect the leaders in discrete-time fashion.

The contributions of the proposed containment control are summarized as follow:

  - We propose a practical containment control solution to track the convex hull of the leaders with unknown dynamics.
  - The leaders can be with heterogeneous dynamics and they do not have to cooperate with the follower agents.
  - The follower agents communicate over a strongly connected and weight-balanced directed graph which is less restrictive than connected undirected graphs normally considered in the literature.
  - The proposed containment control scheme allows the follower agents to have a continuous–time dynamics, but only communicate in the discrete-time fashion.

- Chapter 4: Distributed dynamic active weighted average consensus

  This chapter studies the dynamic active weighted average consensus problem, which is motivated by the observation that we made in Chapter 3 that the containment control

can be cast as a dynamic active average consensus problem. In the containment control, at any given time, only a part of follower agents, which is referred to as active agents, observer a subset of leaders. Each active agent computes the geometric center of its observed leaders as its local reference. We theoretically show that the average of the local references collected by active agents is a point in the convex hull of the leaders. Therefore, the containment control problem can be cast as a dynamic active average consensus problem whose objective is both active and passive agents to track the average of the collected local references.

We propose a continuous-time solution for dynamic active weighted average consensus over connected graphs that requires only the rate of the change of the reference inputs to be bounded. Also, the agents can switch between active and passive modes instantaneously, as long as a dwell time exists between the switching incidences. Abrupt switching is usually the case for practical problems where agents are observing dynamic activities that can enter or leave the observation zone of the agents and thus change the agents' role from active to passive or vice versa in a non-smooth fashion. Next, we study the discrete-time implementation of our proposed dynamic active weighted average consensus algorithm and use it to solve a containment control problem. In the previous chapter, we used two parallel conventional dynamic average consensus algorithms, one to generate the sum of the measurements divided by the size of the network and the other to obtain the sum of the active agents divided by the size of the network. Then, the average of the active measurements is obtained from dividing the output of the first algorithm by that of the second one. But when the second one tends to cross zero, it causes infinity tracking error so we need to introduce zero crossing protection measure. This chapter is offering a computationally more efficient algorithm, which has a lower communication complexity and avoids zero-crossing problem for its approach to solve dynamic active weighted average consensus problem.

The contributions of the proposed dynamic active average consensus algorithm is sum-

marized as follow:

  - Compared to [59–61], to guarantee bounded error tracking, our algorithm only requires the derivative of the reference signals to be bounded.
  - The proposed algorithm allows the agents switch between active and passive modes instantaneously.
  - We show that a containment control problem can be formulated as an active average consensus problem and solved using our proposed discrete-time algorithm.
  - The proposed algorithm has the same computational and communication complexity of conventional dynamic average consensus algorithms.

- Chapter 5: Distributed coverage control for mobile agents deployment

  In this chapter, we propose a distributed deployment solution for a group of mobile agents that are deployed over a set of dense targets to provide a service. The agents are heterogeneous in a sense that their QoS, modeled as a spatial density distribution, is not homogeneous. The deployment objective is to match the collective QoS of the service agents as close as possible to the the density distribution of the targets such that the service provided by the agents efficiently covers the targets. In our setting the target density distribution of the target is unknown a priori. We propose a consensus-based distributed expectation-maximization (EM) algorithm for agents to estimate the target density distribution, which we model as a Gaussian mixture model (GMM). The GMM not only decomposes the target density distribution to a set of Gaussian bases but also partitions the area to subregions each of which represents a Gaussian basis. Unlike the distributed Voronoi partitioning that require the agents to communicate to their Voronoi neighbors, which may be unrealistic because Voronoi neighbors may be our of communication range of each other, our approach only requires the communication graph among the agents to be connected. The agents use the Kullback-Leibler divergence (KLD) to evaluate the similarity between their service

distribution and each Gaussian basis of subregion. Then, a multi-agent assignment problem is formulated under the framework of the optimal mass transport to allocate each agent to a subregion of the targets estimated GMM density distribution by taking the KLD as the cost of assignment. The agents cooperate to solve the assignment problem, as a result, the summation of the divergences corresponding to each paired agent's QoS distribution and subregion's Gaussian basis is minimal. We demonstrate our results for the case that the QoS is modeled as a anisotropic Gaussian distribution, and discuss how our results can be used for non-Gaussian distributions, as well. We demonstrate the applications of the proposed coverage control in the deployments of sensor network and UAV-aid wireless communication network.

The contributions of the proposed coverage control are summarized as follow:

- The proposed deployment strategy considers mobile agents with heterogeneous and anisotropic QoS distribution.
- Agents are not required to know the density distribution (priority function) of targets in advance.
- The implementation is distributed and only requires the communication graph among the agents to be connected.

Chapter 2

Distributed Single Leader Following Control

2.1 Introduction

The first objective of this chapter is to design a leader-following algorithm, which steers a group of homogeneous followers with linear dynamics to be at the sampled states of a leader agent at finite specified times. Then the result is extended to heterogeneous followers that aim to follower the leader in a specified formation. We assume that the only information available about the dynamics of the leader is its sampled states, which is known only to a subset of the followers at the sampling times. We make no assumptions about the input of the leader or the structural form of its dynamics. The sampled states of the leader can be the states of a physical system (e.g., in a pursuit-evasion problem) or a set of desired reference states of a virtual leader (e.g., in a waypoint tracking problem).

Inspired by the classical optimal control results, we propose a distributed minimum energy control strategy to solve the leader-following problem for networks that the interaction

topology of the followers plus the leader is an acyclic digraph with the leader as the global sink. Directed acyclic interaction topology can be interpreted as the agents only obtaining information from those in front of them (see, [82, 83] for algorithms designed over acyclic graphs).

For the homogeneous followers, the algorithm not only results in a leader following behavior, but also it makes the states and inputs of the followers become fully synchronized after the first sampling epoch. A sufficient condition is provided for heterogeneous followers to achieve the same synchronization. Our algorithm also allows the followers to track the sampled state of the leader with a locally chosen offset, which can be time-varying. This offset, when the followers are mobile agents and their whole state or part of it is the position vector, can be used to enable the followers to form a transnational invariant formation [84] about the sampled state of the leader. Moreover, for a special class of LTI MAS, we show that our results can be extended to solve a leader-following problem where we want only an output of the followers to follow the leader's state.

Four numerical examples are used to demonstrate the effectiveness of the algorithm. The first example considers a case of homogeneous followers and shows the application of the proposed algorithm in a leader-following task under a specific formation structure for a group of unicycle robots. In this example, the leader's dynamics is nonlinear while the dynamics of the followers are feedback linearized. In the second example, we demonstrate the use of our algorithm for reference state tracking via a group of second order integrator followers with bounded control. Using the intrinsic properties of our leader-following algorithm, in this example, we show that the arrival times at the reference states can be designed in such a way that the inputs of the agents stay within the pre-specified saturation bounds. The third example consider a case of heterogeneous followers, and shows the application of our leader-following algorithm in following a nonlinear mass-spring-damper leader under a specific formation structure for a group of heterogeneous linear mass-spring-damper systems.

The last example demonstrates an output-tracking scenario for a group of aircraft.

The outline of this chapter is as follows: Section 2.2 gathers basic notation and graph-theoretic terminology and notions. Section 2.3 discusses about the leader-following control for homogeneous followers. The problem definition, distributed leader-follower algorithm and two numerical application examples are presented sequentially in the subsections. Section 2.4 extends the leader-following control to heterogeneous followers tacking a leader with a specified formation. Similarly, the problem definition, distributed leader-follower algorithm and two numerical application examples for heterogeneous followers are discussed in the subsections. Section 2.5 concludes the results of this chapter.

2.2 Notations and Preliminaries

Notation: We let $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Z}$, and $\mathbb{Z}_{\geq 0}$ denote the set of real, positive real, non-negative real, integer, and non-negative integer numbers, respectively. The transpose of a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ is $\mathbf{A}^\top$.

Graph theoretic notations and definitions: Here we review our graph related notations and relevant definitions and concepts from graph theory following [49]. A *digraph*, is a triplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{1, \ldots, N\}$ is the *node set* and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the *edge set*, and $\mathbf{A} = [\mathsf{a}_{ij}] \in \mathbb{R}^{N\times N}$ is the *adjacency* matrix of the graph defined according to $\mathsf{a}_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $\mathsf{a}_{ij} = 0$, otherwise. An edge (i, j) from i to j means that agent j can send information to agent i. Here, i is called an *in-neighbor* of j and j is called an *out-neighbor* of i. A *directed path* is a sequence of nodes connected by edges. A directed path that starts and ends at the same node and all other nodes on the path are distinct is called a *cycle*. A digraph without cycles is called *directed acyclic graph*. The *out-degree* of a node i is $\mathsf{d}^i_{\text{out}} = \Sigma_{j=1}^N \mathsf{a}_{ij}$. The out-degree matrix of a graph is $\mathbf{D}_{\text{out}} = \text{Diag}(\mathsf{d}^1_{\text{out}}, \mathsf{d}^2_{\text{out}}, \cdots, \mathsf{d}^N_{\text{out}})$. We

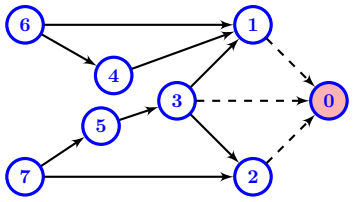

Figure 2.1 – A leader-follower network. The interaction topology of the follower agents, $\mathcal{G}$, shown via the network with solid edges, is an acyclic digraph. Agent 0 is the leader. The edges of $\mathcal{G}_l$ is shown by the dashed arrow. Here, the leader is the global sink of the $\mathcal{G} \cup \mathcal{G}_l$, therefore, its information reaches all the agents in an explicit or implicit manner.

denote the set of in-neighbors of an agent i by $\mathcal{N}_{\text{in}}^i$ and the out-neighbors of agent i by $\mathcal{N}_{\text{out}}^i$. A node $i \in \mathcal{V}$ is called a *global sink* of $\mathcal{G}$ if it outdegree $\mathsf{d}_{\text{out}}^i = 0$ and for every node $j \in \mathcal{V}$ there is at least a path from j to i.

2.3 Homogeneous group of followers with an active non-homogeneous leader

2.3.1 Problem definition

In this section we formalize our problem of interest. We consider a group of N MAS whose dynamics is described by

$$\dot{\mathbf{x}}^i(t) = \mathbf{A}\,\mathbf{x}^i(t) + \mathbf{B}\,\mathbf{u}^i(t), \quad i \in \mathcal{V} = \{1, \cdots, N\}, \tag{2.1}$$

where $\mathbf{x}^i \in \mathbb{R}^n$ is the state vector and $\mathbf{u}^i \in \mathbb{R}^m$ is the control vector. These agents aim to follow a dynamic signal $\mathbf{x}^0(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$. This signal can be a dynamic reference signal of a virtual leader or the state of an active physical leader with nonlinear dynamics

$$\dot{\mathbf{x}}^0(t) = f^0(\mathbf{x}^0(t), \mathbf{u}^0(t), t). \tag{2.2}$$

in which the control vector $\mathbf{u}^0 \in \mathbb{R}^{m^0}$ is unknown. The interaction topology between the follower agents is described by a acyclic digraph, denoted by $\mathcal{G}$. A subset of agents in $\mathcal{G}$ has access to $\mathbf{x}^0(t)$ at the sampling times t_k, $k \in \mathbb{Z}_{\geq 0}$. Throughout the paper we assume that $T_k = t_{k+1} - t_k \in \mathbb{R}_{>0}$ for any $k \in \mathbb{Z}_{\geq 0}$ with $t_0 = 0$. Moreover, we let $\mathcal{N}_{\text{in}}^0$ be the subset of the agents in $\mathcal{G}$ that are connected to the leader. We let $\mathcal{G}_l$ be the digraph consisted of the leader and $\mathcal{N}_{\text{in}}^0$ and the directed edges connecting $\mathcal{N}_{\text{in}}^0$ to the leader. In what follows, we assume that the leader is the global sink of the $\mathcal{G} \cup \mathcal{G}_l$, so that its information reaches all the agents in an explicit or implicit manner (see Fig. 2.1 for an example).

The objective of this chapter is to design a distributed control rule for the input vector $\mathbf{u}^i(t)$ of each agent such that they can cooperatively steer the group to be at the state $x^0(t_k)$ of the leader in time before the next sampling time t_{k+1}, i.e.,

$$\mathbf{x}^i(t_{k+1}) = \mathbf{x}^0(t_k), \quad i \in \{1, \ldots, N\}.$$

Note that the agents have no information about the dynamics that creates the sampled states $x^0(t_k)$. Here, we assume that agents' dynamics (2.1) is controllable. Recall that if a linear system is controllable, there always exists a control to move the state of the system from any point in the state space to any other point in finite time.

2.3.2 Leader-following control for homogeneous followers

In this section, we develop a novel distributed solution to solve the leader-follower problem stated in Section 2.3.1. We start by using a classical optimal control result to make the following statement. To present this result, we recall that

$$\mathbf{G}(t) = \int_0^t \mathrm{e}^{\mathbf{A}(t-\tau)} \mathbf{B}\mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t-\tau)} d\tau, \tag{2.3}$$

is the controllability Gramian of $(\mathbf{A}, \mathbf{B})$ for any finite time $t \in \mathbb{R}_{>0}$. Since $(\mathbf{A}, \mathbf{B})$ is controllable, $\mathbf{G}(t)$ is full rank and invertible at each time $t \in \mathbb{R}_{>0}$.

Lemma 2.3.1. Consider a leader-follower problem where each agent' dynamics is given by (2.1). Suppose i is a follower agent in $\mathcal{G}$ that has access to $\mathbf{x}^0(t)$ of the leader at each sampling time t_k, $k \in \mathbb{Z}_{\geq 0}$, i.e., $i \in \mathcal{N}_{\text{in}}^0$. Starting at an initial condition $\mathbf{x}^i(t_0) \in \mathbb{R}^m$ and $\mathbf{u}^i(t_0) = \mathbf{0}$, for any $i \in \mathcal{N}_{\text{in}}^0$ let

$$\mathbf{u}^i(t) = \mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top (t_{k+1}-t)} \, \mathbf{G}_k^{-1} (\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k} \, \mathbf{x}^i(t_k)), \qquad t \in (t_k, t_{k+1}], \tag{2.4}$$

where $T_k = t_{k+1} - t_k \in \mathbb{R}_{>0}$, and

$$\mathbf{G}_k = \mathbf{G}(T_k) = \int_0^{T_k} \mathrm{e}^{\mathbf{A}(T_k - \tau)} \, \mathbf{B}\mathbf{B}^\top \, \mathrm{e}^{\mathbf{A}^\top (T_k - \tau)} d\tau. \tag{2.5}$$

Then, for every $i \in \mathcal{N}_{\text{in}}^0$ we have $x^i(t_{k+1}) = x^0(t_k)$ for all $k \in \mathbb{Z}_{\geq 0}$. Moreover, at each time interval $[t_k, t_{k+1}]$, the control input $\mathbf{u}^i(t)$ of $i \in \mathcal{N}_{\text{in}}^0$ satisfies

$$\mathbf{u}^i(t) = \operatorname{argmin} \int_{t_k}^{t_{k+1}} \mathbf{u}^i(\tau)^\top \mathbf{u}^i(\tau) \mathrm{d}\tau, \quad s.t. \tag{2.6a}$$

$$\dot{\mathbf{x}}^i(t) = \mathbf{A} \, \mathbf{x}^i(t) + \mathbf{B} \, \mathbf{u}^i(t), \tag{2.6b}$$

$$\mathbf{x}^i(t_k) = \mathbf{x}^i(t_k), \quad \mathbf{x}^i(t_{k+1}) = \mathbf{x}^0(t_k). \tag{2.6c}$$

Proof. The proof follows from the classical finite time minimum energy optimal control design [81, page 138]. □

Recall here that $\mathbf{G}_k$ in (2.5) is the controllability Gramian of $(\mathbf{A}, \mathbf{B})$. Since $(\mathbf{A}, \mathbf{B})$ is controllable, the matrix $\mathbf{G}_k$ is invertible.

Lemma 2.3.1 essentially states that any follower that samples the leader, in the inter-sampling time interval can use the classical minimum energy control to steer towards the latest sampled

state of the leader. Next, we show that this idea can be extended to a distributed setting in which only a subset of the followers have access to the leader's sampled state. To present our results we first introduce some notations. We denote the adjacency matrix and out-degree matrix of the followers' interaction topology $\mathcal{G}$, respectively, by $\mathbf{A} = [\mathsf{a}_{ij}]$ and $\mathbf{D}_{\text{out}} = \text{Diag}(\mathsf{d}^1_{\text{out}}, \mathsf{d}^2_{\text{out}}, \cdots, \mathsf{d}^N_{\text{out}})$. We let

$$1^i = \begin{cases} 1, & i \in \mathcal{N}^0_{\text{in}}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.7}$$

be the indicator operator that defines the state of connectivity of agent i of $\mathcal{G}$ to the leader. We also define

$$\bar{\mathbf{G}}_k(t) = \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}(t-\tau)}\, \mathbf{B}\mathbf{B}^\top\, \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-\tau)} d\tau, \quad t \in [t_k, t_{k+1}]. \tag{2.8}$$

We notice that $\bar{\mathbf{G}}_k(t) = \mathbf{G}(t - t_k)\mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)}$, where $\mathbf{G}$ is the controllability gramian (2.3). Therefore at each finite time $t \in (t_k, t_{k+1}]$, by virtue of controllability of $(\mathbf{A}, \mathbf{B})$, $\bar{\mathbf{G}}_k(t)$ is invertible.

With the proper notations at hand, we present our distributed solution to solve our leader-follower problem of interest as follows.

Theorem 2.3.1. Consider a leader-follower problem where the follower agents' dynamics are given by (2.1). Suppose the leader's time-varying state is $\mathbf{x}^0 : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$. Let the network topology be such that $\mathcal{G} \cup \mathcal{G}_l$ an acyclic digraph with 0 as the global sink. Suppose every follower agent $i \in \mathcal{N}^0_{\text{in}}$ has access to $\mathbf{x}_0(t)$ at each sampling time t_k, $k \in \mathbb{Z}_{\geq 0}$. Let $\mathbf{P}(t) = \bar{\mathbf{G}}_k^{-1}(t)$ for $t \in (t_k, t_{k+1}]$. Starting at an initial condition $\mathbf{x}^i(t_0) \in \mathbb{R}^n$ and $\mathbf{u}^i(t_0) = \mathbf{0}$, let for $t \in (t_k, t_{k+1}]$

$$\mathbf{u}^i(t) = \omega_l \left(\mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)}\, \mathbf{G}_k^{-1} (\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k}\, \mathbf{x}^i(t_k)) \right) +$$

$$\omega_f\Big(\mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)}\,\mathbf{P}(t)\times\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathbf{x}^j(t)-\mathrm{e}^{\mathbf{A}(t-t_k)}\,\mathbf{x}^j(t_k))+ \tag{2.9}$$
$$\mathbf{B}^\top\mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)}\,\mathbf{G}_k^{-1}\,\mathrm{e}^{\mathbf{A}T_k}\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathbf{x}^j(t_k)-\mathbf{x}^i(t_k))\Big),$$

where $\omega_l=\frac{1^i}{1^i+\mathsf{d}^i_{\text{out}}}$, $\omega_f=\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}$. Then, the followings hold for $t\in\mathbb{R}_{\geq 0}$ and $k\in\mathbb{Z}_{\geq 0}$:

(a) $x^i(t_{k+1})=x^0(t_k)$, $i\in\{1,\ldots,N\}$;

(b) the trajectory of every follower $i\in\{1,\ldots,N\}$ is

$$\mathbf{x}^i(t)=\mathrm{e}^{\mathbf{A}(t-t_k)}\mathbf{x}^i(t_k)+\bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k)-\mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k)); \tag{2.10}$$

(c) the control input $\mathbf{u}^i(t)$ of every agent $i\in\{1,\ldots,N\}$ is equal to (2.4).

Proof. For $\mathcal{G}\cup\mathcal{G}_l$ an acyclic digraph with 0 as the global sink, the agents can be sorted into a series of hierarchical subsets. Without loss of generality, we sort the agents as follows. Recall that $\mathcal{V}=\{1,\cdots,N\}$ is the set of the followers. We let $\mathcal{V}_0=\{0\}$. Next, we let $\mathcal{V}_1$ to the subset of agents in $\mathcal{G}$ that are connected to the leader but they have no out-neighbor in $\mathcal{G}$, i.e., $\mathcal{V}_1=\{i\in\mathcal{V}|\ 1^i=1 \text{ and } \mathcal{N}^i_{\text{out}}=\{\}\}$. We sequentially define the lower subset as $\mathcal{V}_k=\{i\in\mathcal{V}\backslash\cup_{j=1}^{k-1}\mathcal{V}_j|\ \mathcal{N}^i_{\text{out}}\subseteq\cup_{j=0}^{k-1}\mathcal{V}_j\}$, where $k\in\{2,\cdots,m\}$, such that $\cup_{j=1}^{m}\mathcal{V}_j=\mathcal{V}$. In short, in this hierarchy, the agents in the lower subset only connects to the agents in the higher subsets.

The proof of the statements can be carried out by induction over time intervals $(t_k,t_{k+1}]$, $k\in\mathbb{Z}_{\geq 0}$. For space limitation we skip the proof of validity of $k=0$. We assume that the statements hold for $[t_0,t_k]$ and we show the validity of the statements at $(t_k,t_{k+1}]$ as follows via the mathematical induction over $\mathcal{V}_l$ where $l\in\{1,\cdots,m\}$.

Consider first the dynamics of the follower agents in $\mathcal{V}_l$. For $l=1$, the control (2.9) reduces

to (2.4), since $\omega_l = 1$ and $\sum_{j=1}^{N} \mathsf{a}_{ij} = 0$. Hence statement (c) holds. The trajectory of $\mathbf{x}^i(t)$ after substituting for the control input $\mathbf{u}^i$ is

$$\begin{aligned}\mathbf{x}^i(t) &= \mathrm{e}^{\mathbf{A}(t-t_k)}\,\mathbf{x}^i(t_k) + \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}(t-\tau)}\,\mathbf{B}\mathbf{u}^i(\tau)d\tau \\ &= \mathrm{e}^{\mathbf{A}(t-t_k)}\,\mathbf{x}^i(t_k) + \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}(t-\tau)}\,\mathbf{B}\mathbf{B}^\top\,\mathrm{e}^{\mathbf{A}^\top(t_{k+1}-\tau)}\,\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k))d\tau.\end{aligned}$$

Then given (2.8), the trajectories of agents $i \in \mathcal{V}_1$ is given by (2.10) for $t \in \mathbb{R}_{\geq 0}$, confirming Statement (b). Moreover, when $t = t_{k+1}$, the final state of the end of this period is

$$\mathbf{x}^i(t_{k+1}) = \mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k)\bar{\mathbf{G}}_k(t_{k+1})\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k)) = \mathbf{x}^0(t_k),$$

so statement (a) holds.

Next, let statements (a), (b) and (c) be true for $i \in \mathcal{V}_{l-1}$. Then, for the follower $i \in \mathcal{V}_l$ we have:

$$\begin{aligned}\mathbf{x}^i(t) &= \mathrm{e}^{\mathbf{A}(t-t_k)}\,\mathbf{x}^i(t_k) + \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}(t-\tau)}\,\mathbf{B}\mathbf{u}^i(\tau)d\tau \\ &= \mathrm{e}^{\mathbf{A}(t-t_k)}\,\mathbf{x}^i(t_k) \\ &\quad + \frac{1^i}{1^i + \mathsf{d}^i_{\text{out}}} \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}(t-\tau)}\,\mathbf{B}\mathbf{B}^\top\,\mathrm{e}^{\mathbf{A}^\top(t_{k+1}-\tau)}\,\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k))d\tau \\ &\quad + \frac{1}{1^i + \mathsf{d}^i_{\text{out}}} \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}(t-\tau)}\,\mathbf{B}\mathbf{B}^\top\,\mathrm{e}^{\mathbf{A}^\top(t_{k+1}-\tau)}\,\mathbf{P}(\tau)\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathbf{x}^j(\tau) - \mathrm{e}^{\mathbf{A}(\tau-t_k)}\,\mathbf{x}^j(t_k))d\tau \\ &\quad + \frac{1}{1^i + \mathsf{d}^i_{\text{out}}} \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}(t-\tau)}\,\mathbf{B}\mathbf{B}^\top\,\mathrm{e}^{\mathbf{A}^\top(t_{k+1}-\tau)}\,\mathbf{G}_k^{-1}\,\mathrm{e}^{\mathbf{A}T_k}\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathbf{x}^j(t_k) - \mathbf{x}^i(t_k))d\tau \\ &= \mathrm{e}^{\mathbf{A}(t-t_k)}\,\mathbf{x}^i(t_k) \\ &\quad + \frac{1^i}{1^i + \mathsf{d}^i_{\text{out}}}\bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k)) \\ &\quad + \frac{1}{1^i + \mathsf{d}^i_{\text{out}}} \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}(t-\tau)}\,\mathbf{B}\mathbf{B}^\top\,\mathrm{e}^{\mathbf{A}^\top(t_{k+1}-\tau)}\,\mathbf{P}(\tau)\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathbf{x}^j(\tau) - \mathrm{e}^{\mathbf{A}(\tau-t_k)}\,\mathbf{x}^j(t_k))d\tau\end{aligned}$$

$$+\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}\bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}\,\mathrm{e}^{\mathbf{A}T_k}\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathbf{x}^j(t_k)-\mathbf{x}^i(t_k))$$

Since $j\in\mathcal{V}_m$, where $m<l$, the trajectory $\mathbf{x}^j(\tau)$ of agent j is assume to follow (2.10). Therefore, we can put (2.10) into $\mathbf{x}^j(\tau)$.

$$\begin{aligned}
\mathbf{x}^i(t)=&\,\mathrm{e}^{\mathbf{A}(t-t_k)}\,\mathbf{x}^i(t_k)\\
&+\frac{1^i}{1^i+\mathsf{d}^i_{\text{out}}}\bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k)-\mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k))\\
&+\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}\int_{t_k}^{t}\mathrm{e}^{\mathbf{A}(t-\tau)}\,\mathbf{B}\mathbf{B}^\top\,\mathrm{e}^{\mathbf{A}^\top(t_{k+1}-\tau)}\\
&\times\mathbf{P}(\tau)\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathrm{e}^{\mathbf{A}(\tau-t_k)}\,\mathbf{x}^j(t_k)+\bar{\mathbf{G}}_k(\tau)\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k)-\mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^j(t_k))-\mathrm{e}^{\mathbf{A}(\tau-t_k)}\,\mathbf{x}^j(t_k))d\tau\\
&+\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}\bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}\,\mathrm{e}^{\mathbf{A}T_k}\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathbf{x}^j(t_k)-\mathbf{x}^i(t_k))\\
=&\,\mathrm{e}^{\mathbf{A}(t-t_k)}\,\mathbf{x}^i(t_k)\\
&+\frac{1^i}{1^i+\mathsf{d}^i_{\text{out}}}\bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k)-\mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k))\\
&+\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}\bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathbf{x}^0(t_k)-\mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^j(t_k))\\
&+\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}\bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}\,\mathrm{e}^{\mathbf{A}T_k}\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathbf{x}^j(t_k)-\mathbf{x}^i(t_k))\\
=&\,\mathrm{e}^{\mathbf{A}(t-t_k)}\,\mathbf{x}^i(t_k)\\
&+\frac{1^i}{1^i+\mathsf{d}^i_{\text{out}}}\bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k)-\mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k))\\
&+\frac{\mathsf{d}^i_{\text{out}}}{1^i+\mathsf{d}^i_{\text{out}}}\bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k)-\mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k))\\
=&\,\mathrm{e}^{\mathbf{A}(t-t_k)}\,\mathbf{x}^i(t_k)+\bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k)-\mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k))\\
\mathbf{x}^i(t_{k+1})=&\,\mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k)+\bar{\mathbf{G}}_k(t_{k+1})\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k)-\mathrm{e}^{\mathbf{A}T_k}\,\mathbf{x}^i(t_k))=\mathbf{x}^0(t_k)
\end{aligned}$$

Thereby, statement (a) and (b) also hold for the case $l = l$. Then we show that control (2.9) is equivalent to (2.4):

$$
\begin{aligned}
\mathbf{u}^i(t) =& \frac{1^i}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)} \mathbf{G}_k^{-1}(\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k} \mathbf{x}^i(t_k))] \\
&+ \frac{1}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)} \mathbf{P}(t) \sum_{j=1}^{N} \mathsf{a}_{ij}(\mathbf{x}^j(t) - \mathrm{e}^{\mathbf{A}(t-t_k)} \mathbf{x}^j(t_k)) \\
&+ \mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)} \mathbf{G}_k^{-1} \mathrm{e}^{\mathbf{A}T_k} \sum_{j=1}^{N} \mathsf{a}_{ij}(\mathbf{x}^j(t_k) - \mathbf{x}^i(t_k))] \\
=& \frac{1^i}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)} \mathbf{G}_k^{-1}(\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k} \mathbf{x}^i(t_k))] \\
&+ \frac{1}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)} \mathbf{P}(t) \\
&\times \sum_{j=1}^{N} \mathsf{a}_{ij}(\mathrm{e}^{\mathbf{A}(t-t_k)} \mathbf{x}^j(t_k) + \bar{\mathbf{G}}_k(t)\mathbf{G}_k^{-1}(\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k} \mathbf{x}^j(t_k)) - \mathrm{e}^{\mathbf{A}(t-t_k)} \mathbf{x}^j(t_k)) \\
&+ \mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)} \mathbf{G}_k^{-1} \mathrm{e}^{\mathbf{A}T_k} \sum_{j=1}^{N} \mathsf{a}_{ij}(\mathbf{x}^j(t_k) - \mathbf{x}^i(t_k))] \\
=& \frac{1^i}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)} \mathbf{G}_k^{-1}(\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k} \mathbf{x}^i(t_k))] \\
&+ \frac{\mathsf{d}^i_{\text{out}}}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)} \mathbf{G}_k^{-1}(\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k} \mathbf{x}^i(t_k))] \\
=& \mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top(t_{k+1}-t)} \mathbf{G}_k^{-1}(\mathbf{x}^0(t_k) - \mathrm{e}^{\mathbf{A}T_k} \mathbf{x}^i(t_k)).
\end{aligned}
$$

Therefore, statement (c) holds.

Since both the base case $l = 1$ and the inductive step have been proved, by mathematical induction statement (a), (b) and (c) hold for all $l \in \{1, \cdots, m\}$. □

In the following, we give several remarks regarding the structural properties of the leader-follower algorithm of Theorem 2.3.1. First, it is worth to note here the interesting synchronization property that the leader-follower algorithm described in Theorem 2.3.1 has.

Corollary 2.3.1 (Followers' synchronization)**.** Consider the Leader-follower interaction de-

scribed in Theorem 2.3.1. Then, $\mathbf{x}^i(t) = \mathbf{x}^j(t)$ for $t \in [t_1, \infty)$ and $\mathbf{u}^i(t) = \mathbf{u}^j(t)$ for $t \in (t_1, \infty)$, for every $i, j \in \{1, \ldots, N\}$. Moreover, if $\mathbf{x}^i(0) = \mathbf{x}_0 \in \mathbb{R}^n$ for all $i \in \{1, \ldots, N\}$, then these equalities also hold for $t \in [0, t_1]$ □

Next we observe the following minimum energy control property which follows from the statement (c) of Theorem 2.3.1 and the classical result in Lemma 2.3.1.

Corollary 2.3.2 (Minimum energy control in $[t_k, t_{k+1}]$)**.** Consider the Leader-follower interaction described in Theorem 2.3.1. Then, at each time interval $[t_k, t_{k+1}]$, $k \in \mathbb{Z}_{\geq 0}$, the control input $\mathbf{u}^i$ of each follower agent $i \in \{1, \ldots, N\}$ is the minimum energy controller that transfers the agent from its current state $\mathbf{x}(t_k)$ to their desired state $\mathbf{x}(t_{k+1}) = \mathbf{x}^0(t_k)$. □

Remark 2.3.1 (Tracking a priori known desired states at exact sampling time and design of arrival times)**.** We note that if the leader is virtual and the sampled states are some desired states that are known a priori to $\mathcal{N}_{\text{in}}^0$ with desired arrival time in $\mathbb{R}_{>0}$, the agents can arrive at the desired state of the leader at the desired arrival time. Furthermore, in cases that the arrival times is not specified one of the followers in $\mathcal{N}_{\text{in}}^0$ (we refer to it as super node that knows the initial state of all the other followers) can design the arrival times to meet other optimality conditions or to avoid violating constraints such as input saturation. In case of input saturation, the fact that by virtue of statement (c) of Theorem 2.3.1 the form of input vector of the followers are known to be (2.4) can be instrumental to the super node in design of arrival times. Our second demonstrative example in the proceeding section offers the details.. □

Remark 2.3.2 (Robustness to state perturbations)**.** We observe that the leader-following algorithm of Theorem 2.3.1 has robustness to state perturbations similar to the well-known Model Predictive Control (MPC). Even though the controller implemented in each epoch $(t_k, t_{k+1}]$ is an open-loop control, since every follower exerts its state at time t_k as initial condition to the controller, the algorithm can account for the slight perturbations in the follower final state $\mathbf{x}^i(t_{k+1})$ at the end of each epoch. □

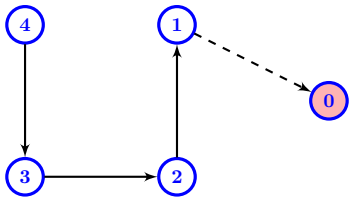

Figure 2.2 – Interaction topology of the leader-follower problem of first example. Agent 0 is the leader.

Remark 2.3.3 (Time-varying MAS dynamics and network topology)**.** From the proof of Theorem 2.3.1, we can see that the followers dynamics can be allowed to be time-varying but piece-wise constant over each time interval $(t_k, t_{k+1}]$, i.e., $\mathbf{A}(t) = \bar{\mathbf{A}}(t_k)$ and $\mathbf{B}(t) = \bar{\mathbf{B}}(t_k)$, $i \in \{1, \ldots, N\}$ for $t \in (t_k, t_{k+1}]$. Similarly the network topology can be allowed to be time-varying as long as between $(t_k, t_{k+1}]$ the topology is fixed and satisfies the connectivity condition of Theorem 2.3.1. □

2.3.3 Demonstrative examples

In this section, we demonstrate use of our proposed algorithm in Theorem 2.3.1 in solving two leader-follower problems for mobile agents.

A leader tracking problem for a group of unicycle robots

In this demonstrative example, we use our leader-follower algorithm of Theorem 2.3.1 to solve a leader-follower problem for a group of unicycle robots

$$\begin{aligned} \dot{x}^i &= v^i \cos\theta^i, \\ \dot{y}^i &= v^i \sin\theta^i, \qquad i \in \{0, 1, 2, 3, 4\}, \\ \dot{\theta}^i &= \omega^i, \end{aligned} \tag{2.11}$$

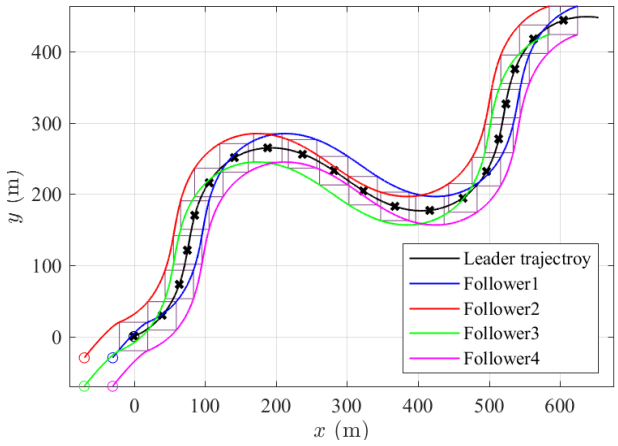

Figure 2.3 – The trajectories of the leader and the followers in the first numerical example.

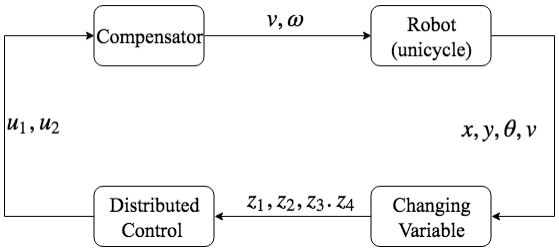

Figure 2.4 – The linearization procedure of unicycle.

where $v^i \in \mathbb{R}$ and $\omega^i \in \mathbb{R}$ are linear velocity and angular velocity of each agent i, respectively. Here, agents $\mathcal{V} = \{1, 2, 3, 4\}$ are the follower agents, and agent 0 is the leader robot which moves with a known constant linear velocity of $v^0 = 100\,(\text{m/min})$ but unknown angular velocity ω^0. The interaction topology of the agents is shown in Fig. 2.2. Agent 1 obtains the states of the leader with a sampling rate of 2 per minute, i.e., $T_k = 0.5$ minutes, $k \in \mathbb{Z}_{\geq 0}$. The followers start at $\mathbf{x}^1(0) = [-30 \quad -30 \quad 0]^\top$, $\mathbf{x}^2(0) = [-70 \quad -30 \quad 0]^\top$, $\mathbf{x}^3(0) = [-70 \quad -70 \quad 0]^\top$, $\mathbf{x}^4(0) = [-30 \quad -70 \quad 0]^\top$ in a rectangular formation. The follower team wants to follow the leader in a rectangle formation that preserves the initial vertical and horizontal relative distances of the agents. To satisfy this objective, we first feedback linearize the dynamics of the followers and then implement our proposed leader-follower algorithm of Theorem 2.3.1 as described below. The results of implementing our leader-follower algorithm is shown in Fig. 2.3. The '$\times$' represents the sampled leader positions and the gray window shows the resulting formation of the followers at the sampling times.

It is well known that the unicycle dynamics is feedback linearizable. The linearization procedure is described in [85] and is shown in Fig. 2.4. For each follower agent $i \in \{1,2,3,4\}$, the feedback linearized dynamics consists of two decoupled second-order integral systems of

$$\begin{bmatrix} \dot{z}_1^i \\ \dot{z}_2^i \\ \dot{z}_3^i \\ \dot{z}_4^i \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} z_1^i \\ z_2^i \\ z_3^i \\ z_4^i \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} u_1^i \\ u_2^i \end{bmatrix}, \tag{2.12}$$

with the changing variable

$$\begin{aligned} z_1^i &= x^i, \\ z_2^i &= v^i \cos\theta^i, \\ z_3^i &= y^i, \\ z_4^i &= v^i \sin\theta^i. \end{aligned} \tag{2.13}$$

The resulting dynamic compensator of each follower agent $i \in \{1,2,3,4\}$ is

$$\begin{aligned} \dot{\eta}^i &= u_1^i \cos\theta^i + u_2^i \sin\theta^i, \quad \eta^i(0) \in \mathbb{R}, \\ v^i &= \eta^i, \\ \omega^i &= \frac{u_2^i \cos\theta^i - u_1^i \sin\theta^i}{\eta^i}. \end{aligned} \tag{2.14}$$

In order to follow the states of the leader, we assume that agent 1 constructs $z^0(t_k) = \begin{bmatrix} x^0(t_k) & v^0\cos(\theta(t_k)) & y^0(t_k) & v^0\sin(\theta(t_k)) \end{bmatrix}^\top$ from the leader's sampled state vector $x^0(t_k) = \begin{bmatrix} x^0(t_k) & y^0(t_k) & \theta^0(t_k) \end{bmatrix}^\top$. The follower agents share their feedback linearized states z^i. The follower agents use (2.9) to obtain $\mathbf{u}^i \in \mathbb{R}^2$, $i \in \{1,2,3,4\}$, of (2.12). Then, they obtain their inputs (v^i, ω^i), $i \in \{1,2,3,4\}$, from (2.14). Here, we note that the leaders dynamics is nonlinear and is not required to be feedback linearized. Given the initial location of

the agents in Fig. 2.2, to preserve the initial rectangular formation at every $t = t_k$, the desired state of agent i is an offset with respect to the state of agent $j \in \mathcal{N}_{\text{out}}^i$. We define the offset parameter $\mathbf{F}^{ij} \in \mathbb{R}^4$ as the desired state offset of agent i at $t = t_k$ with respect to agent $j \in \mathcal{N}_{\text{out}}^i$. We set that the followers want to keep the sampled leader's states $\mathbf{x}^0(t_k)$ in the center of their rectangular formations at every $t = t_{k+1}$. Thereby, the offset parameters are $\mathbf{F}^{10} = [20 \quad 0 \quad 20 \quad 0]^\top$, $\mathbf{F}^{21} = [-40 \quad 0 \quad 0 \quad 0]^\top$, $\mathbf{F}^{32} = [0 \quad 0 \quad -40 \quad 0]^\top$ and $\mathbf{F}^{43} = [40 \quad 0 \quad 0 \quad 0]^\top$. Then, $\mathbf{x}^i(t_k) = \mathbf{x}^j(t_k) + \mathbf{F}^{ij}$, $j \in \mathcal{N}_{\text{out}}^i$, is achieved by adding the term $\mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top (t_{k+1}-t)} \mathbf{G}_k^{-1} \mathbf{F}^{ij}$ to the control (2.9). One can easily verify the validity of this approach in a similar way to the proof of Theorem 2.3.1. The details are omitted here for brevity but the leader following with a formation offset will be formally presented and proved in the next section.

Reference state tracking for a group of second integrator dynamics with bounded inputs

We consider a group of 6 followers with second order integrator dynamics

$$\dot{\mathbf{x}}^i = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}} \mathbf{x}^i + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{B}} u^i, \quad -5 \leq u^i \leq 5, \tag{2.15}$$

for $i \in \{1, \ldots, 6\}$. The interaction topology of these followers is shown in Fig. 2.5, where, agent 0 is the virtual leader that is defined more precisely below. Starting at initial conditions $\mathbf{x}^1(0) = [0 \quad 0]^\top$, $\mathbf{x}^2(0) = [2 \quad 0]^\top$, $\mathbf{x}^3(0) = [-2 \quad 0]^\top$, $\mathbf{x}^4(0) = [5 \quad 0]^\top$, $\mathbf{x}^5(0) = [10 \quad 0]^\top$, $\mathbf{x}^6(0) = [-10 \quad 0]^\top$, the leader-following mission for this team is to traverse through the sequence of desired states $\mathbf{x}^{\text{d}} = \{\mathbf{x}_1^{\text{d}}, \mathbf{x}_2^{\text{d}}, \mathbf{x}_3^{\text{d}}, \mathbf{x}_4^{\text{d}}\} = \left\{ \begin{bmatrix} 50 \\ 10 \end{bmatrix}, \begin{bmatrix} -50 \\ 10 \end{bmatrix}, \begin{bmatrix} 20 \\ 10 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$, which for privacy reason are only known to follower 1. The objective is to meet the sequence of desired

states without violating any of the followers' control bounds.

In this problem setting, follower 1 is the super node that knows the initial starting state of all the followers in the team and has computational power to compute the arrival times as follows to meet the team's objective.

- First, we note that by virtue of statement (c) of Theorem 2.3.1 the form of input vector of the followers are known to be (2.4). Since follower 1 knows $\mathbf{x}^i(t_0)$ for $i \in \{1, \cdots, 6\}$, follower 1 can evaluate $u^i(t)$ of all the followers. Starting with $t_0^{\mathrm{d}} = 0$, follower 1 computes the arrival time at desired state $\mathbf{x}_1^{\mathrm{d}}$ from the process below

$$t_1^{\mathrm{d},i} = \operatorname{argmin} \int_{t_0^{\mathrm{d}}}^{t_1^{\mathrm{d},i}} \mathrm{d}\tau \quad \text{subject to } -5 \leq u^i(t) \leq 5, \tag{2.16}$$

  where $u^i(t) = \mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top (t_1^{\mathrm{d},i} - t)} \mathbf{G}_0^{-1} (\mathbf{x}_0^{\mathrm{d}} - \mathrm{e}^{\mathbf{A}T_0} \mathbf{x}^i(0))$ with $T_0 = t_1^{\mathrm{d},i} - t_0^{\mathrm{d}}$. Then, the arrival time so that the followers input do not saturate over $(t_0^d, t_1^{\mathrm{d},i}]$ is set to $t_1^{\mathrm{d}} = \max\{t_1^{\mathrm{d},i}\}$.

- Due to Corollary 2.3.1, after first epoch, the agents inputs are equal to each other. Then, the remaining arrival time t_l^{d}, $l \in \{2, 3, 4\}$. Agent 1 computes these desired times from the optimization problem

$$t_{k+1}^{\mathrm{d}} = \operatorname{argmin} \int_{t_k^{\mathrm{d}}}^{t_{k+1}^{\mathrm{d}}} \mathrm{d}\tau \quad \text{subject to } -5 \leq u(t) \leq 5, \tag{2.17}$$

  where $u(t) = \mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top (t_{k+1}^{\mathrm{d}} - t)} \mathbf{G}_k^{-1} (\mathbf{x}_{k+1}^{\mathrm{d}} - \mathrm{e}^{\mathbf{A}T_k} \mathbf{x}_k^{\mathrm{d}})$ with $T_k = t_{k+1}^{\mathrm{d}} - t_k^{\mathrm{d}}$, for $k \in \{1, 2, 3\}$.

The solution for this set of sequential optimal control problem is $t_1^{\mathrm{d}} = 6.7178$, $t_2^{\mathrm{d}} = 25.2061$, $t_3^{\mathrm{d}} = 30.1592$ and $t_4^{\mathrm{d}} = 40.4885$ seconds. Finally, agent 1 broadcasts the desired arrival times to the network. Broadcasting the reference states is not allowed due to privacy reasons. We note that these processes can be done offline. To match the notation in (2.9), at the implantation stage, we set $\mathbf{x}^0(t_{k-1}) = \mathbf{x}_k^{\mathrm{d}}$, $T_{k-1} = t_k^{\mathrm{d}} - t_{k-1}^{\mathrm{d}}$, and $t_k = t_{k-1} + T_{k-1}$, $k \in$

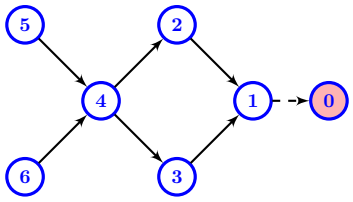

Figure 2.5 – Interaction topology with 6 agents. Agent **0** is the virtual leader.

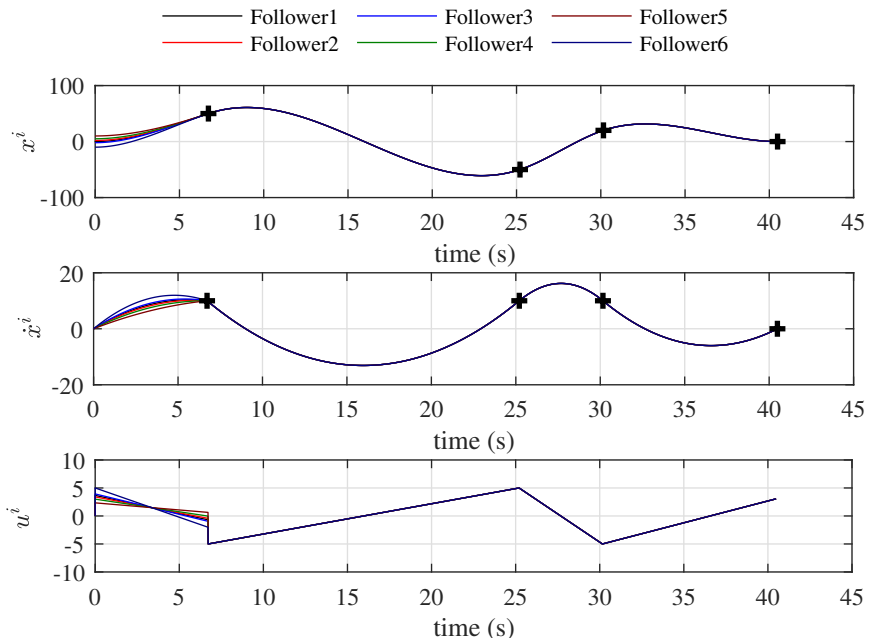

Figure 2.6 – The state and control trajectories of followers of the second numerical example.

$\{1, \ldots, 4\}$, where $t_0^{\mathrm{d}} = 0$. Figures 2.6 shows that all the followers meet the desired reference state of the virtual leader at the specified arrival times without delay (the '+' marks the reference states). Figure 2.6 also shows the control history of the agents. As seen, the control inputs respect the saturation bounds 5 or -5. We can also observe that the followers' states and inputs, as predicted in Corollary 2.3.1, are all synchronized after the first epoch.

Note that, one may wonder the feasibility of the design of the sampling time T_k, $k \in \{0, 1, 2, 3\}$, as described above. The following lemma shows that every T_k, $k \in \{0, 1, 2, 3\}$ is guaranteed to be a finite value.

Lemma 2.3.2. Consider a second order integrator system initialized at $\mathbf{x}(t_0) = \boldsymbol{\chi}(t_0) \in \mathbb{R}^2$

at time $t_0 \in \mathbb{R}_{\geq 0}$. This system implements the minimum energy controller

$$u(t) = \mathbf{B}^\top \mathrm{e}^{\mathbf{A}^\top (t_{k+1}-t)}\, \mathbf{G}(T_k)^{-1}(\boldsymbol{\chi}(t_{k+1}) - \mathrm{e}^{\mathbf{A}T_k}\, \mathbf{x}(t_k)), \quad t \in (t_k, t_{k+1}], \tag{2.18}$$

and $u(t_0) = 0$ to traverse sequentially through a set of $m+1$ points $\{\boldsymbol{\chi}(t_k)\}_{k=0}^m \subset \mathbb{R}^2$, where $t_k \in \mathbb{R}_{\geq 0}$ is the arrival time at point $\boldsymbol{\chi}(t_k)$, $T_k = t_{k+1} - t_k \in \mathbb{R}_{>0}$, $\mathbf{A}$ and $\mathbf{B}$ are given in (2.15), and $\mathbf{G}$ is defined in (2.3). For this system, there always exists a set of finite arrival times $\{t_k\}_0^m$ such that $|u(t)| \leq |u_{max}|$ for any $t \in [t_0, t_m]$, where $u_{max} \in \mathbb{R}_{\geq 0}$ is the known bound on the control input.

Proof. To establish the proof, similar to the proof of Theorem 2.3.1, we relay on the mathematical induction over time intervals $[t_0, t_{k+1}]$, $k \in \{0, \cdots, m-1\}$. The proof of the case for $k = 0$ is similar to the case of $k+1$ and omitted here for brevity. Now let the statements be valid over $[t_0, t_k]$. Next, we show the validity of the statements at $(t_k, t_{k+1}]$, and as a result the validity of the statement over $[t_0, t_{k+1}]$. Let $\boldsymbol{\chi}(t_k) = [\chi_{k,1} \quad \chi_{k,2}]^\top \in \mathbb{R}^2$, $k \in \{0, \cdots, m\}$. Also, given $t \in (t_k, t_{k+1}]$, let $t' = t - t_k \in (0, T_k]$. Disregarding the control bounds, (2.18) results in $\mathbf{x}(t_k) = \boldsymbol{\chi}(t_k)$ for $k \in \{0, \cdots, m-1\}$. Therefore, control (2.18) can also be expressed as

$$u(t') = \begin{bmatrix} \frac{12}{T_k^3}(T_k - t') - \frac{6}{T_k^2} & -\frac{6}{T_k^2}(T_k - t') + \frac{4}{T_k} \end{bmatrix} \begin{bmatrix} \chi_{k+1,1} - \chi_{k,1} + T_k \chi_{k,2} \\ \chi_{k+1,2} - \chi_{k,2} \end{bmatrix}.$$

Since $u(t')$ is an affine function of t', the maximum value of $|u(t')|$ is at either $t' \to 0^+$ or $t' = T_k$. That is, $|u(t')| \leq |u(t' \to 0^+)|$ or $|u(t')| \leq |u(T_k)|$ where $u(t' \to 0^+) = \lim_{t' \to 0^+} u(t')$. Next, we show that there always exists a T_k that makes $|u(t' \to 0^+)| \leq |u_{max}|$ and $|u(T_k)| \leq |u_{max}|$, which means that $|u(t')| \leq |u_{max}|$, $t' \in (0, T_k]$. Note that $|u(t' \to 0^+)| = \left| \frac{6}{T_k^2}(\chi_{k+1,1} - \chi_{k,1}) - \frac{2}{T_k}(\chi_{k+1,2} - 2\chi_{k,2}) \right| \leq \left| \frac{6}{T_k^2}(\chi_{k+1,1} - \chi_{k,1}) \right| + \left| \frac{2}{T_k}(\chi_{k+1,2} - 2\chi_{k,2}) \right|$, and $|u(T_k)| = \left| -\frac{6}{T_k^2}(\chi_{k+1,1} - \chi_{k,1}) + \frac{2}{T_k}(2\chi_{k+1,2} - \chi_{k,2}) \right| \leq \left| \frac{6}{T_k^2}(\chi_{k+1,1} - \chi_{k,1}) \right| + \left| \frac{2}{T_k}(2\chi_{k+1,2} - \chi_{k,2}) \right|$. Since

the upper bounds established for $|u(t' \to 0^+)|$ and $|u(T_k)|$ monotonically decrease when T_k increases, there always exists finite value of T_k such that these upper bounds become equal to u_{max}. Therefore, there always exists a finite value of T_k for which control law (2.18) does not violate the controller saturation bound. □

2.4 Heterogeneous group of followers tracking an active leader in a formation

In this section, we extend the leader following control to adapt a more general scenario. We consider the followers are with different linear dynamics and they aim to follow a uncertain leader in a specified formation. The formation can be view as a offsets of position related states between the followers and the leader. The following section redefines the leader following problem and its objective in this general scenario.

2.4.1 Problem definition

We consider a group of N heterogeneous MAS whose dynamics is described by

$$\dot{\mathbf{x}}^i(t) = \mathbf{A}^i\,\mathbf{x}^i(t) + \mathbf{B}^i\,\mathbf{u}^i(t), \quad i \in \{1, \cdots, N\}, \tag{2.19}$$

where $\mathbf{x}^i \in \mathbb{R}^n$ is the state vector and $\mathbf{u}^i \in \mathbb{R}^{m^i}$ is the control vector. Throughout the paper we assume that the agents' dynamics (2.19) is controllable, i.e., $(\mathbf{A}^i, \mathbf{B}^i)$ for $i \in \{1, \ldots, N\}$ is controllable. These agents (referred hereafter as followers) aim to follow a dynamic signal $\mathbf{x}^0(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ with possibly a locally chosen offset. This signal can be a dynamic reference signal of a virtual leader or the state of an active physical leader with (possibly) a nonlinear dynamics, e.g., $\dot{\mathbf{x}}^0(t) = f^0(\mathbf{x}^0(t), \mathbf{u}^0(t), t)$. The dynamical model and the input

$\mathbf{u}^0 \in \mathbb{R}^{m^0}$ of the leader is not known to the followers. The interaction topology between the followers is described by a acyclic digraph, denoted by $\mathcal{G}$. Only a subset of followers in $\mathcal{G}$, denoted by $\mathcal{N}_{\text{in}}^0 \neq \{\}$, has access to $\mathbf{x}^0(t)$ at the sampling times $t_k \in \mathbb{R}$, $k \in \mathbb{Z}_{\geq 0}$. Throughout the paper we assume that $T_k = t_{k+1} - t_k \in \mathbb{R}_{>0}$ for any $k \in \mathbb{Z}_{\geq 0}$ with $t_0 = 0$. We let $\mathcal{G}_l$ be the digraph consisted of the leader and $\mathcal{N}_{\text{in}}^0$ and the directed edges connecting $\mathcal{N}_{\text{in}}^0$ to the leader. In what follows, we assume that the leader is the global sink of $\overline{\mathcal{G}} = \mathcal{G} \cup \mathcal{G}_l$, so that its information reaches all the agents in an explicit or implicit manner (see Fig. 2.1 for an example). We let $\overline{\mathcal{N}}_{\text{out}}^i$ be the set of the out-neighbors of agent $i \in \{0, 1, \cdots, N\}$ in graph $\overline{\mathcal{G}}$; we note the $\overline{\mathcal{N}}_{\text{out}}^0 = \{\}$.

Give that we only have a limited information about the leader (only the sampled states of the leader $\mathbf{x}^0(t_k)$ is available), we seek a practical solution that enables the followers to arrive at the sampled state of the leader before the next sampling time. Therefore, our objective in this paper is to design a distributed control rule for the input vector $\mathbf{u}^i(t)$ of each follower $i \in \{1, \cdots, N\}$ such that

$$\mathbf{x}^i(t_{k+1}) = \mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k), \quad i \in \{1, \cdots, N\}. \tag{2.20}$$

That is, the follower $i \in \{1, \ldots, N\}$ can steer itself to be in $\mathbf{F}^{i0}(t_k) \in \mathbb{R}^n$ offset with respect to the state $\mathbf{x}^0(t_k)$ of the leader in time before the next sampling time t_{k+1}. We note that the set of offsets $\{\mathbf{F}^{i0}(t_k)\}_{i=1}^N$, when it is related to the position offsets of the agents, defines the formation of the followers around the leader. Here, the term formation refers to *transnational invariant* formation [84, Section 6.1.1]. For scenarios where the objective is to synchronize to the state of the leader, $\mathbf{F}^{i0}(t_k)$ is set to zero for all $i \in \{1, \cdots, N\}$. To form the offset, we assume that at each sampling time t_k, follower $i \in \{1, \cdots, N\}$ knows $\mathbf{F}^{ij}(t_k) = \mathbf{F}^{i0}(t_k) - \mathbf{F}^{j0}(t_k)$ for $j \in \overline{\mathcal{N}}_{\text{out}}^i$; either the follower is given $\mathbf{F}^{ij}(t_k)$ with respect to its out-neighbor j or constructs it locally after agent j sends its $\mathbf{F}^{j0}(t_k)$ to agent i. We note that if the leader is a global sink of $\overline{\mathcal{G}}$, given $\mathbf{x}^0(t_k)$ and a set of $\mathbf{F}^{ij}(t_k)$, $i \in \{1, \ldots, N\}$ and

$j \in \overline{\mathcal{N}}^i_{\text{out}}$, we can show that the state offset $\mathbf{F}^{i0}(t_k)$ for follower i with respect to the leader is unique.

2.4.2 Leader-following control for heterogeneous followers

In this section, we develop a novel distributed solution to solve the leader-following problem stated in Section 2.4.1. We start by using a classical optimal control result to make the following statement.

Lemma 2.4.1. Consider a leader-following formation problem where each follower's dynamics is given by (2.19) with $(\mathbf{A}^i, \mathbf{B}^i)$ controllable. Suppose i is a follower in $\mathcal{G}$ that has access to $\mathbf{x}^0(t)$ of the leader at each sampling time t_k, $k \in \mathbb{Z}_{\geq 0}$, i.e., $i \in \mathcal{N}^0_{\text{in}}$. Also, $\mathbf{F}^{i0}(t_k) \in \mathbb{R}^n$ is the desired state off-set with respect to $\mathbf{x}^0(t_k)$. Starting at an initial condition $\mathbf{x}^i(t_0) \in \mathbb{R}^n$ with $\mathbf{u}^i(t_0) = \mathbf{0}$, for any $i \in \mathcal{N}^0_{\text{in}}$ let

$$\mathbf{u}^i(t) = \mathbf{B}^{i^\top} \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\, \mathbf{G}_k^{i^{-1}} (\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\, \mathbf{x}^i(t_k)), \quad t \in (t_k, t_{k+1}], \tag{2.21}$$

where $T_k = t_{k+1} - t_k \in \mathbb{R}_{>0}$, and

$$\mathbf{G}_k^i = \mathbf{G}^i(T_k) = \int_0^{T_k} \mathrm{e}^{\mathbf{A}^i(T_k-\tau)}\, \mathbf{B}^i \mathbf{B}^{i^\top}\, \mathrm{e}^{\mathbf{A}^{i^\top}(T_k-\tau)} d\tau. \tag{2.22}$$

Then, for every $i \in \mathcal{N}^0_{\text{in}}$ we have $\mathbf{x}^i(t_{k+1}) = \mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k)$ for all $k \in \mathbb{Z}_{\geq 0}$. Moreover, at each time $t \in [t_k, t_{k+1}]$, the control input $\mathbf{u}^i(t)$ of $i \in \mathcal{N}^0_{\text{in}}$ satisfies

$$\mathbf{u}^i(t) = \operatorname{argmin} \int_{t_k}^{t_{k+1}} \mathbf{u}^i(\tau)^\top \mathbf{u}^i(\tau) \mathrm{d}\tau, \quad \text{subject to} \tag{2.23a}$$

$$\dot{\mathbf{x}}^i(t) = \mathbf{A}^i\, \mathbf{x}^i(t) + \mathbf{B}^i\, \mathbf{u}^i(t), \tag{2.23b}$$

$$\mathbf{x}^i(t_k) = \mathbf{x}^i(t_k), \quad \mathbf{x}^i(t_{k+1}) = \mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k). \tag{2.23c}$$

Proof. The proof follows from the classical finite time minimum energy optimal control design [81, page 138]. □

Lemma 2.4.1 essentially states that any follower that samples the leader, in the inter-sampling time interval can use the classical minimum energy control to steer towards the latest sampled state of the leader with an offset if specified. Next, we show that this idea can be extended to a distributed setting in which only a subset of the followers have access to the leader's sampled state. To present our results we first introduce some notations. We denote the adjacency matrix and out-degree matrix of the followers' interaction topology $\mathcal{G}$, respectively, by $\mathbf{A} = [\mathsf{a}_{ij}]$ and $\mathbf{D}_{\text{out}} = \text{Diag}(\mathsf{d}^1_{\text{out}}, \mathsf{d}^2_{\text{out}}, \cdots, \mathsf{d}^N_{\text{out}})$. We let

$$\mathit{1}^i = \begin{cases} 1, & i \in \mathcal{N}^0_{\text{in}}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.24}$$

be the indicator operator that defines the state of connectivity of follower i to the leader. For $i \in \{1, \ldots, N\}$, we also define

$$\mathbf{P}^i(t) = \begin{cases} \mathbf{0} & t = t_k, \\ \overline{\mathbf{G}}_k^{i^{-1}}(t) & t \in (t_k, t_{k+1}], \end{cases} \tag{2.25a}$$

where (2.25b)

$$\overline{\mathbf{G}}_k^i(t) = \int_{t_k}^t \mathrm{e}^{\mathbf{A}^i(t-\tau)}\, \mathbf{B}^i \mathbf{B}^{i^\top}\, \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-\tau)} d\tau, \quad t \in [t_k, t_{k+1}]. \tag{2.25c}$$

We notice that $\overline{\mathbf{G}}_k^i(t) = \mathbf{G}^i(t-t_k)\ \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}$, where $\mathbf{G}^i$ is the controllability Gramian (2.22). Therefore at each finite time $t \in (t_k, t_{k+1}]$, by virtue of controllability of $(\mathbf{A}^i, \mathbf{B}^i)$, $\overline{\mathbf{G}}_k^i(t)$ is invertible. Moreover, note that using the classical control results we can show that $\overline{\mathbf{G}}_k^i(t)$ can be computed numerically from $\overline{\mathbf{G}}_k^i(t) = \mathbf{W}^i(t)\mathbf{\Phi}^i(t)$ where $\mathbf{W}^i(t) = \mathbf{G}^i(t - t_k)$ and

$\mathbf{\Phi}^i(t) = \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}$ for $t \in [t_k, t_{k+1}]$ are obtained from

$$\begin{aligned}
\dot{\mathbf{W}}^i(t) &= \mathbf{A}^i\mathbf{W}^i(t) + \mathbf{W}^i(t)\mathbf{A}^{i^\top} + \mathbf{B}^i\mathbf{B}^{i^\top}, \quad \mathbf{W}^i(t_k) = \mathbf{0}_{n\times n},\\
\dot{\mathbf{\Phi}}^i(t) &= -\mathbf{A}^{i^\top}\mathbf{\Phi}^i(t), \qquad\qquad\qquad\qquad \mathbf{\Phi}^i(t_k) = \mathrm{e}^{\mathbf{A}^{i^\top}T_k}.
\end{aligned}$$

With the proper notations at hand, we present our distributed solution to solve our leader-following problem of interest as follows.

Theorem 2.4.1 (A leader-following in formation algorithm for a group of heterogeneous LTI followers)**.** Consider a leader-following problem where the followers' dynamics are given by (2.19). Suppose the leader's time-varying state is $\mathbf{x}^0 : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$. Let the network topology $\overline{\mathcal{G}} = \mathcal{G} \cup \mathcal{G}_l$ be an acyclic digraph with leader, node 0, as the global sink. Suppose every follower $i \in \mathcal{N}_{\text{in}}^0$ has access to $\mathbf{x}^0(t)$ at each sampling time t_k, $k \in \mathbb{Z}_{\geq 0}$. Let $\mathbf{F}^{i0}(t_k) \in \mathbb{R}^n$ and $\mathbf{F}^{ij}(t_k) \in \mathbb{R}^n$ be the desired state offset (formation) with reference to $\mathbf{x}^0(t_k)$ and $\mathbf{x}^j(t_{k+1})$, respectively. Starting at an initial condition $\mathbf{x}^i(t_0) \in \mathbb{R}^n$ with $\mathbf{u}^i(t_0) = \mathbf{0}$, let for $t \in (t_k, t_{k+1}]$

$$\begin{aligned}
\mathbf{u}^i(t) =& \omega_l^i \Big(\mathbf{B}^{i^\top}\,\mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\,\mathbf{G}_k^{i^{-1}}(\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\,\mathbf{x}^i(t_k))\Big) +\\
&\omega_f^i\Big(\mathbf{B}^{i^\top}\,\mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\,\mathbf{G}_k^{i^{-1}}\sum_{j=1}^{N}\mathsf{a}_{ij}\mathbf{G}_k^j\mathbf{P}^j(t)(\mathbf{x}^j(t) - \mathrm{e}^{\mathbf{A}^j(t-t_k)}\,\mathbf{x}^j(t_k))\\
&+ \mathbf{B}^{i^\top}\,\mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\,\mathbf{G}_k^{i^{-1}}\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathrm{e}^{\mathbf{A}^j T_k}\,\mathbf{x}^j(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\,\mathbf{x}^i(t_k) - \mathbf{F}^{ij}(t_k))\Big), \qquad (2.26)
\end{aligned}$$

where $\mathbf{P}^j(t)$ is given in (2.25b), $\omega_l^i = \frac{1^i}{1^i + \mathsf{d}_{\text{out}}^i}$, and $\omega_f^i = \frac{1}{1^i + \mathsf{d}_{\text{out}}^i}$. Then, the followings hold for $t \in \mathbb{R}_{\geq 0}$ and $k \in \mathbb{Z}_{\geq 0}$:

(a) $\mathbf{x}^i(t_{k+1}) = \mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k)$, moreover, $\mathbf{x}^j(t_{k+1}) - \mathbf{x}^i(t_{k+1}) = \mathbf{F}^{ij}(t_k)$ $i, j \in \{1, \ldots, N\}$ and $i \neq j$;

(b) the trajectory of every follower $i \in \{1, \ldots, N\}$ is

$$\mathbf{x}^i(t) = \mathrm{e}^{\mathbf{A}^i(t-t_k)}\,\mathbf{x}^i(t_k) + \overline{\mathbf{G}}_k^{\,i}(t)\mathbf{G}_k^{i^{-1}}(\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\,\mathbf{x}^i(t_k)); \tag{2.27}$$

(c) the control input $\mathbf{u}^i(t)$ of every agent $i \in \{1, \ldots, N\}$ is equal to (2.21). □

Proof. For $\mathcal{G} \cup \mathcal{G}_l$ an acyclic digraph with 0 as the global sink, the agents can be sorted into a series of hierarchical subsets. Without loss of generality, we sort the agents as follows. Recall that $\mathcal{V} = \{1, \cdots, N\}$ is the set of the followers. We let $\mathcal{V}_0 = \{0\}$. Next, we let $\mathcal{V}_1$ to be the subset of agents in $\mathcal{G}$ that are connected to the leader but they have no out-neighbor in $\mathcal{G}$, i.e., $\mathcal{V}_1 = \{i \in \mathcal{V} |\ 1^i = 1 \text{ and } \mathcal{N}^i_{\text{out}} = \{\}\}$. We sequentially define the lower subset as $\mathcal{V}_k = \{i \in \mathcal{V} \backslash \cup_{j=1}^{k-1} \mathcal{V}_j |\ \mathcal{N}^i_{\text{out}} \subseteq \cup_{j=0}^{k-1} \mathcal{V}_j\}$, where $k \in \{2, \cdots, m\}$, such that $\cup_{j=1}^{m} \mathcal{V}_j = \mathcal{V}$. In short, in this hierarchy, the agents in the lower subset only receive information from the agents in the higher subsets.

We use mathematical induction over time intervals $[t_0, t_{k+1}]$, $k \in \mathbb{Z}_{>0}$ for our proof. That is we show that the theorem statements hold for $k = 0$. Then assuming that the theorem statements hold for k, we show the validity of the statement over $k + 1$. The proof of the case for $k = 0$ is very similar to the case of $k + 1$ and omitted here of brevity. Now let the theorem statements be valid over $[t_0, t_k]$ and we show the validity of the statements at $(t_k, t_{k+1}]$ and as a result the validity of the statement over $[t_0, t_{k+1}]$. For our proof we use as the mathematical induction over $\mathcal{V}_l$ where $l \in \{1, \cdots, m\}$.

Consider first the dynamics of the followers in $\mathcal{V}_l$. For $l = 1$, the control (2.26) reduces to (2.21), since $\omega_l = 1$ and $\sum_{j=1}^{N} \mathsf{a}_{ij} = 0$. Hence statement (c) holds. The trajectory of $\mathbf{x}^i(t)$ after substituting for the control input $\mathbf{u}^i$ is

$$\mathbf{x}^i(t) = \mathrm{e}^{\mathbf{A}^i(t-t_k)}\,\mathbf{x}^i(t_k) + \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}^i(t-\tau)}\,\mathbf{B}^i\mathbf{u}^i(\tau)d\tau$$

$$
\begin{aligned}
&= \mathrm{e}^{\mathbf{A}^i(t-t_k)}\,\mathbf{x}^i(t_k) + \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}^i(t-\tau)}\,\mathbf{B}^i\mathbf{B}^{i^\top}\,\mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-\tau)}\,\mathbf{G}_k^{i^{-1}} \\
&\quad \cdot (\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\,\mathbf{x}^i(t_k))d\tau \\
&= \mathrm{e}^{\mathbf{A}^i(t-t_k)}\,\mathbf{x}^i(t_k) + \overline{\mathbf{G}}^{\,i}_{\,k}(t)\mathbf{G}_k^{i^{-1}}(\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\,\mathbf{x}^i(t_k)).
\end{aligned}
$$

Then given (2.25c), the trajectories of agents $i \in \mathcal{V}_1$ is given by (2.27) for $t \in \mathbb{R}_{\geq 0}$, confirming Statement (b). Moreover, when $t = t_{k+1}$, the final state of the end of this period is

$$
\mathbf{x}^i(t_{k+1}) = \mathrm{e}^{\mathbf{A}^i T_k}\,\mathbf{x}^i(t_k) + \overline{\mathbf{G}}^{\,i}_{\,k}(t_{k+1})\mathbf{G}_k^{i^{-1}}(\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\,\mathbf{x}^i(t_k)) = \mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k).
$$

Also, the relative state with respect to follower $j \in \overline{\mathcal{N}}^i_{\text{out}}$, is $\mathbf{x}^j(t_{k+1}) - \mathbf{x}^i(t_{k+1}) = \mathbf{x}^0(t_k) - \mathbf{F}^{j0}(t_k) - \mathbf{x}^0(t_k) + \mathbf{F}^{i0}(t_k) = \mathbf{F}^{ij}(t_k)$. Therefore, statement (a) holds.

Next, let statements (a), (b) and (c) be true for $i \in \mathcal{V}_s$, $s \in \{1, \cdots, l-1\}$. Then, for the follower $i \in \mathcal{V}_l$ we have:

$$
\begin{aligned}
\mathbf{x}^i(t) &= \mathrm{e}^{\mathbf{A}^i(t-t_k)}\,\mathbf{x}^i(t_k) + \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}^i(t-\tau)}\,\mathbf{B}^i\mathbf{u}^i(\tau)d\tau \\
&= \mathrm{e}^{\mathbf{A}^i(t-t_k)}\,\mathbf{x}^i(t_k) + \frac{1^i}{1^i + \mathsf{d}^i_{\text{out}}} \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}^i(t-\tau)}\,\mathbf{B}^i\mathbf{B}^{i^\top}\,\mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-\tau)}\,\mathbf{G}_k^{i^{-1}} \\
&\quad \cdot (\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\,\mathbf{x}^i(t_k))d\tau \\
&\quad + \frac{1}{1^i + \mathsf{d}^i_{\text{out}}} \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}^i(t-\tau)}\,\mathbf{B}^i\mathbf{B}^{i^\top}\,\mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-\tau)}\,\mathbf{G}_k^{i^{-1}} \\
&\quad \cdot \sum_{j=1}^{N} \mathsf{a}_{ij}\mathbf{G}_k^j\mathbf{P}^j(\tau)(\mathbf{x}^j(\tau) - \mathrm{e}^{\mathbf{A}^j(\tau-t_k)}\,\mathbf{x}^j(t_k))d\tau \\
&\quad + \frac{1}{1^i + \mathsf{d}^i_{\text{out}}} \int_{t_k}^{t} \mathrm{e}^{\mathbf{A}^i(t-\tau)}\,\mathbf{B}^i\mathbf{B}^{i^\top}\,\mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-\tau)}\,\mathbf{G}_k^{i^{-1}} \\
&\quad \cdot \sum_{j=1}^{N} \mathsf{a}_{ij}(\mathrm{e}^{\mathbf{A}^j T_k}\,\mathbf{x}^j(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\,\mathbf{x}^i(t_k) - \mathbf{F}^{ij}(t_k))d\tau \\
&= \mathrm{e}^{\mathbf{A}^i(t-t_k)}\,\mathbf{x}^i(t_k) + \frac{1^i}{1^i + \mathsf{d}^i_{\text{out}}}\overline{\mathbf{G}}^{\,i}_{\,k}(t)\mathbf{G}_k^{i^{-1}}(\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\,\mathbf{x}^i(t_k))
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}\int_{t_k}^{t} \mathrm{e}^{\mathbf{A}^i(t-\tau)}\,\mathbf{B}^i\mathbf{B}^{i^\top}\,\mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-\tau)}\,\mathbf{G}_k^{i-1}\\
&\quad\cdot\sum_{j=1}^{N}\mathsf{a}_{ij}\mathbf{G}_k^j\mathbf{P}^j(\tau)(\mathbf{x}^j(\tau)-\mathrm{e}^{\mathbf{A}^j(\tau-t_k)}\,\mathbf{x}^j(t_k))d\tau\\
&+\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}\overline{\mathbf{G}}^{\,i}_{\,k}(t)\mathbf{G}_k^{i-1}\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathrm{e}^{\mathbf{A}^jT_k}\,\mathbf{x}^j(t_k)-\mathrm{e}^{\mathbf{A}^iT_k}\,\mathbf{x}^i(t_k)-\mathbf{F}^{ij}(t_k)).
\end{aligned}
$$

Since $j\in\mathcal{V}_s$, where $s<l$, the trajectory $\mathbf{x}^j(\tau)$ of agent j is assume to follow (2.27). Therefore, we can put (2.27) into $\mathbf{x}^j(\tau)$.

$$
\begin{aligned}
\mathbf{x}^i(t)=&\,\mathrm{e}^{\mathbf{A}^i(t-t_k)}\,\mathbf{x}^i(t_k)+\frac{1^i}{1^i+\mathsf{d}^i_{\text{out}}}\overline{\mathbf{G}}^{\,i}_{\,k}(t)\mathbf{G}_k^{i-1}(\mathbf{x}^0(t_k)-\mathrm{e}^{\mathbf{A}^iT_k}\,\mathbf{x}^i(t_k)-\mathbf{F}^{i0}(t_k))\\
&+\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}\int_{t_k}^{t}\mathrm{e}^{\mathbf{A}^i(t-\tau)}\,\mathbf{B}^i\mathbf{B}^{i^\top}\mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-\tau)}\,\mathbf{G}_k^{i-1}\sum_{j=1}^{N}\mathsf{a}_{ij}\mathbf{G}_k^j\mathbf{P}^j(\tau)(\mathrm{e}^{\mathbf{A}^j(\tau-t_k)}\,\mathbf{x}^j(t_k)\\
&\qquad+\overline{\mathbf{G}}^j_k(\tau)\mathbf{G}_k^{j^{-1}}(\mathbf{x}^0(t_k)-\mathbf{F}^{j0}(t_k)-\mathrm{e}^{\mathbf{A}^jT_k}\,\mathbf{x}^j(t_k))-\mathrm{e}^{\mathbf{A}^j(\tau-t_k)}\,\mathbf{x}^j(t_k))d\tau\\
&\quad+\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}\overline{\mathbf{G}}^{\,i}_{\,k}(t)\mathbf{G}_k^{i-1}\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathrm{e}^{\mathbf{A}^jT_k}\,\mathbf{x}^j(t_k)-\mathrm{e}^{\mathbf{A}^iT_k}\,\mathbf{x}^i(t_k)-\mathbf{F}^{ij}(t_k))\\
=&\,\mathrm{e}^{\mathbf{A}^i(t-t_k)}\,\mathbf{x}^i(t_k)+\frac{1^i}{1^i+\mathsf{d}^i_{\text{out}}}\overline{\mathbf{G}}^{\,i}_{\,k}(t)\mathbf{G}_k^{i-1}(\mathbf{x}^0(t_k)-\mathbf{F}^{i0}(t_k)-\mathrm{e}^{\mathbf{A}^iT_k}\,\mathbf{x}^i(t_k))\\
&+\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}\overline{\mathbf{G}}^{\,i}_{\,k}(t)\mathbf{G}_k^{i-1}\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathbf{x}^0(t_k)-\mathbf{F}^{j0}(t_k)-\mathrm{e}^{\mathbf{A}^jT_k}\,\mathbf{x}^j(t_k))\\
&+\frac{1}{1^i+\mathsf{d}^i_{\text{out}}}\overline{\mathbf{G}}^{\,i}_{\,k}(t)\mathbf{G}_k^{i-1}\sum_{j=1}^{N}\mathsf{a}_{ij}(\mathrm{e}^{\mathbf{A}^jT_k}\,\mathbf{x}^j(t_k)-\mathrm{e}^{\mathbf{A}^iT_k}\,\mathbf{x}^i(t_k)-\mathbf{F}^{ij}(t_k))\\
=&\,\mathrm{e}^{\mathbf{A}^i(t-t_k)}\,\mathbf{x}^i(t_k)+\frac{1^i}{1^i+\mathsf{d}^i_{\text{out}}}\overline{\mathbf{G}}^{\,i}_{\,k}(t)\mathbf{G}_k^{i-1}(\mathbf{x}^0(t_k)-\mathbf{F}^{i0}(t_k)-\mathrm{e}^{\mathbf{A}^iT_k}\,\mathbf{x}^i(t_k))\\
&+\frac{\mathsf{d}^i_{\text{out}}}{1^i+\mathsf{d}^i_{\text{out}}}\overline{\mathbf{G}}^{\,i}_{\,k}(t)\mathbf{G}_k^{i-1}(\mathbf{x}^0(t_k)-\mathbf{F}^{i0}(t_k)-\mathrm{e}^{\mathbf{A}^iT_k}\,\mathbf{x}^i(t_k))\\
=&\,\mathrm{e}^{\mathbf{A}^i(t-t_k)}\,\mathbf{x}^i(t_k)+\overline{\mathbf{G}}^{\,i}_{\,k}(t)\mathbf{G}_k^{i-1}(\mathbf{x}^0(t_k)-\mathbf{F}^{i0}(t_k)-\mathrm{e}^{\mathbf{A}^iT_k}\,\mathbf{x}^i(t_k)).
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{x}^i(t_{k+1})=&\,\mathrm{e}^{\mathbf{A}^iT_k}\,\mathbf{x}^i(t_k)+\overline{\mathbf{G}}^{\,i}_{\,k}(t_{k+1})\mathbf{G}_k^{i-1}(\mathbf{x}^0(t_k)-\mathbf{F}^{i0}(t_k)-\mathrm{e}^{\mathbf{A}^iT_k}\,\mathbf{x}^i(t_k))\\
=&\,\mathbf{x}^0(t_k)-\mathbf{F}^{i0}(t_k).
\end{aligned}
$$

Similarly, the relative state with respect to agent $j \in \overline{\mathcal{N}}^{\,i}_{\text{out}}$, is $\mathbf{x}^j(t_{k+1}) - \mathbf{x}^i(t_{k+1}) = \mathbf{x}^0(t_k) - \mathbf{F}^{j0}(t_k) - \mathbf{x}^0(t_k) + \mathbf{F}^{i0}(t_k) = \mathbf{F}^{ij}(t_k)$. Thereby, statement (a) and (b) also hold for the case $l = l$. Then we show that control (2.26) is equivalent to (2.21) as follows

$$
\begin{aligned}
\mathbf{u}^i(t) &= \frac{1^i}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^{i^\top} \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\, \mathbf{G}_k^{i^{-1}}(\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\, \mathbf{x}^i(t_k))] \\
&+ \frac{1}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^{i^\top} \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\, \mathbf{G}_k^{i^{-1}} \sum_{j=1}^{N} \mathsf{a}_{ij} \mathbf{G}_k^j \mathbf{P}^j(t)(\mathbf{x}^j(t) - \mathrm{e}^{\mathbf{A}^j(t-t_k)}\, \mathbf{x}^j(t_k)) \\
&\quad + \mathbf{B}^{i^\top} \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\, \mathbf{G}_k^{i^{-1}} \sum_{j=1}^{N} \mathsf{a}_{ij}(\mathrm{e}^{\mathbf{A}^j T_k}\, \mathbf{x}^j(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\, \mathbf{x}^i(t_k) - \mathbf{F}^{ij}(t_k))] \\
&= \frac{1^i}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^{i^\top} \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\, \mathbf{G}_k^{i^{-1}}(\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\, \mathbf{x}^i(t_k))] \\
&+ \frac{1}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^{i^\top} \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\, \mathbf{G}_k^{i^{-1}} \sum_{j=1}^{N} \mathsf{a}_{ij} \mathbf{G}_k^j \mathbf{P}^j(t)(\mathrm{e}^{\mathbf{A}^j(t-t_k)}\, \mathbf{x}^j(t_k) \\
&\quad + \overline{\mathbf{G}}_k^j(t) \mathbf{G}_k^{j^{-1}}(\mathbf{x}^0(t_k) - \mathbf{F}^{j0}(t_k) - \mathrm{e}^{\mathbf{A}^j T_k}\, \mathbf{x}^j(t_k)) - \mathrm{e}^{\mathbf{A}^j(t-t_k)}\, \mathbf{x}^j(t_k)) \\
&+ \mathbf{B}^{i^\top} \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\, \mathbf{G}_k^{i^{-1}} \sum_{j=1}^{N} \mathsf{a}_{ij}(\mathrm{e}^{\mathbf{A}^j T_k}\, \mathbf{x}^j(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\, \mathbf{x}^i(t_k) - \mathbf{F}^{ij}(t_k))] \\
&= \frac{1^i}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^{i^\top} \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\, \mathbf{G}_k^{i^{-1}}(\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\, \mathbf{x}^i(t_k))] \\
&+ \frac{\mathsf{d}^i_{\text{out}}}{1^i + \mathsf{d}^i_{\text{out}}}[\mathbf{B}^{i^\top} \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\, \mathbf{G}_k^{i^{-1}}((\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\, \mathbf{x}^i(t_k))] \\
&= \mathbf{B}^{i^\top} \mathrm{e}^{\mathbf{A}^{i^\top}(t_{k+1}-t)}\, \mathbf{G}_k^{i^{-1}}(\mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k) - \mathrm{e}^{\mathbf{A}^i T_k}\, \mathbf{x}^i(t_k)).
\end{aligned}
$$

Therefore, statement (c) holds.

Since both the base case $l = 1$ and the inductive step have been proved, by mathematical induction statement (a), (b) and (c) hold for all $l \in \{1, \cdots, m\}$. □

Several observations and remarks are in order regarding the leader-following formation algorithm of Theorem 2.4.1.

Remark 2.4.1 (Implementation of control law (2.26))**.** To implement (2.26), we note that

the component of (2.26) that multiplies ω_l^i is computed using the local variables of follower i and the sampled state of the leader if ω_l^i is non-zero, i.e., $i \in \mathcal{N}_{\text{in}}^0$. The component of (2.26) that multiplies ω_f^i is computed using the local variables of follower i and variables of its out-neighbors if $\mathsf{d}_{\text{out}}^i \neq 0$. To compute this term, if the follower i knows $(\mathbf{A}^j, \mathbf{B}^j)$ of its out-neighbor j (which is the case e.g., when the group is homogeneous), it can implement control (2.26) by obtaining the state $\mathbf{x}^j$ of its out-neighbor j and computing $\mathbf{P}^j$ and $\mathrm{e}^{\mathbf{A}^j(t-t_k)}$ locally. Otherwise, each follower needs to obtain $\mathbf{z}^j(t) = \mathbf{G}_k^j \mathbf{P}^j(t)(\mathbf{x}^j(t) - \mathrm{e}^{\mathbf{A}^j(t-t_k)}\,\mathbf{x}^j(t_k))$ and $\mathrm{e}^{\mathbf{A}^j T_k}\,\mathbf{x}^j(t_k)$ of its each out-neighbor j. We note here that since the interval $(t_k, t_{k+1}]$ is open from the left and the dynamics of all the followers is controllable, $\mathbf{P}^i(t)$ is well defined. However, for $t \to t_k^+$ from the right, $\mathbf{P}^i(t)$ goes to infinity. But, since $(\mathbf{x}^j(t) - \mathrm{e}^{\mathbf{A}^j(t-t_k)}\,\mathbf{x}^j(t_k))$ goes to zero as $t \to t_k^+$ from the right, the product $\mathbf{P}^j(t)(\mathbf{x}^j(t) - \mathrm{e}^{\mathbf{A}^j(t-t_k)}\,\mathbf{x}^j(t_k))$ goes to zero as $t \to t_k^+$ from the right. The "high-gain-challenge" observed here is often a common feature in any approach that is geared towards regulation in prescribed finite time. For example, finite-horizon optimal controls with a terminal constraint inevitably yield gains that go to infinity (see e.g., [86–88]). In practice, the multiplication of very large and very small values can create numerical problems. To address the problem, one way proposed in the literature is equivalent to employ a deadzone on $\mathbf{P}^j(t)(\mathbf{x}^j(t) - \mathrm{e}^{\mathbf{A}^j(t-t_k)}\,\mathbf{x}^j(t_k))$ at the beginning of each time interval. Another approach is equivalent to using a lager interval $(t_k - \delta, t_{k+1}]$, where $\delta \in \mathbb{R}_{>0}$ is small positive number, to compute $\mathbf{P}^j(t)$ such that $\mathbf{P}^j(t)$ for $t \in (t_k, t_{k+1}]$ is no longer goes unbounded when t goes to t_k^+ from the right. These approaches of course result in somewhat sacrifices in the accuracy at each arrival value at t_{k+1} at the end of time interval $(t_k, t_{k+1}]$. However, as discussed in Remark 4.1, the errors will not accumulate. As interestingly discussed in [86], the high-gain-challenge in the finite-time control can be contrasted with non-smooth feedback in the sliding mode control where the gain approaches infinity near sliding surface $x = 0$ [89] (but the total control input is zero). However, unlike the sliding mode control, where the practical implementation of the high-gain leads to persistent chattering on the sliding surface [89], in our case the concern arises

only at the start of each transition from one sampling time to the other. The aforementioned practical measures to handle the high-gain-challenge in our setting indeed can be compared to the boundary layer approach [89] in the sliding mode control to eliminate chattering. □

Remark 2.4.2 (Minimum energy control in $[t_k, t_{k+1}]$)**.** From statement (c) of Theorem 2.4.1 it follows that at each time interval $[t_k, t_{k+1}]$, $k \in \mathbb{Z}_{\geq 0}$, the control input $\mathbf{u}^i$ of each follower $i \in \{1, \ldots, N\}$ is the minimum energy controller that transfers the follower from its current state $\mathbf{x}^i(t_k)$ to their desired state $\mathbf{x}^i(t_{k+1}) = \mathbf{x}^0(t_k) - \mathbf{F}_k^{i0}(t_k)$. □

Remark 2.4.3 (Extension of results to output tracking for a special class of MAS)**.** The design methodology of the state formation algorithm of Theorem 2.4.1 can be used in output tracking for a special class of MAS. Let the network topology be as described in Theorem 2.4.1 and the system dynamics of the followers be (2.19) where $\mathbf{x}^i \in \mathbb{R}^{n^i}$ and $\mathbf{u}^i \in \mathbb{R}^{m^i}$ (the state and input dimensions of the followers are not necessarily the same). Let the objective be that the output $\mathbf{y}^i = \mathbf{C}^i\mathbf{x}^i \in \mathbb{R}^n$, $n \leq n^i$, of each follower should satisfy

$$\mathbf{y}^i(t_{k+1}) = \mathbf{x}^0(t_k) - \mathbf{F}^{i0}(t_k), \quad i \in \{1, \cdots, N\}. \tag{2.28}$$

If $\mathbf{C}^i\mathbf{B}^i$ is full row rank, we can use the control $\mathbf{u}^i = \mathbf{B}^{i^\top}\mathbf{C}^{i^\top}(\mathbf{C}^i\mathbf{B}^i\mathbf{B}^{i^\top}\mathbf{C}^{i^\top})^{-1} \cdot$
$(\mathbf{v}^i - \mathbf{C}^i\mathbf{A}^i\mathbf{x}^i)$, $i \in \{1, \cdots, N\}$, to write the output dynamics of each follower i as $\dot{\mathbf{y}}^i = \mathbf{v}^i$. Then the method of Theorem 2.4.1 can be used to design $\mathbf{v}^i \in \mathbb{R}^n$, which can then be used to obtain the appropriate $\mathbf{u}^i$ that will make the followers meet (2.28). □

Finally, we note that if the followers are homogeneous, the followers can achieve full synchronization in the sense stated below.

Corollary 2.4.1 (Full synchronization for homogeneous followers)**.** Let the state offset be constant i.e., $\mathbf{F}^{i0}(t_k) = \mathbf{F}^{i0} \in \mathbb{R}^n$ for all $i \in \{1, \cdots, N\}$ or (equivalently $\mathbf{F}^{ij}(t_k) = \mathbf{F}^{ij} \in \mathbb{R}^n$ for $i, j \in \{1, \cdots, N\}$), and assume that the followers are homogeneous. Then, it follows from statements (b) and (c) of Theorem 2.4.1 that the followers' trajectories and inputs satisfy

$\mathbf{x}^j(t) = \mathbf{x}^i(t) + \mathbf{F}^{ij}$ for $t \in [t_1, \infty)$ and $\mathbf{u}^i(t) = \mathbf{u}^j(t)$ for $t \in (t_1, \infty)$, for every $i, j \in \{1, \ldots, N\}$. One can easily verify this point by shifting the state coordinate with $\mathbf{F}^{ij}$. Moreover, if the agents are initially in the specified offset i.e., $\mathbf{x}^j(0) = \mathbf{x}^i(0) + \mathbf{F}^{ij}$ for all $i, j \in \{1, \ldots, N\}$, then these qualities also hold for $t \in [0, t_1]$. □

Remark 2.4.4 (A sufficient condition for heterogeneous followers to achieve synchronization)**.** *Assume that there exists* $\mathbf{K}^i \in \mathbb{R}^{m^i \times n}$, $\mathbf{W}^i \in \mathbb{R}^{m^i \times m^i}$ *for* $i \in \{1, \cdots, N\}$ *and a controllable pair* $(\mathbf{A}, \mathbf{B})$ *known to all followers, such that using* $\mathbf{u}^i = \mathbf{K}^i\mathbf{x}^i + \mathbf{W}^i\mathbf{v}^i$, $i \in \{1, \cdots, N\}$, *makes the followers dynamic homogeneous, i.e.,* $\dot{\mathbf{x}}^i = \mathbf{A}\mathbf{x}^i + \mathbf{B}\mathbf{v}^i$, $\mathbf{A} = \mathbf{A}^i + \mathbf{B}^i\mathbf{K}^i$ *and* $\mathbf{B} = \mathbf{B}^i\mathbf{W}^i$, $i \in \{1, \cdots, N\}$. *Then, it is also possible to achieve full state synchronization by implementing* (2.26) *to* $\mathbf{v}^i$ *for heterogeneous followers. One sufficient condition for the existence of* $\mathbf{K}^i$ *and* $\mathbf{W}^i$, $i \in \{1, \cdots, N\}$, *is that* $\mathbf{B}^i$ *of each follower* $i \in \{1, \cdots, N\}$ *is full row rank. Then,* $\mathbf{K}^i = \mathbf{B}^{i^\top}(\mathbf{B}^i\mathbf{B}^{i^\top})^{-1}(\mathbf{A} - \mathbf{A}^i)$, $\mathbf{W}^i = \mathbf{B}^{i^\top}(\mathbf{B}^i\mathbf{B}^{i^\top})^{-1}\mathbf{B}$ *and* $(\mathbf{A}, \mathbf{B})$ *can be any controllable pair.*

2.4.3 Demonstrative examples

In this section, we demonstrate our results via numerical examples.

A nonlinear-leader tracking problem for a group of heterogeneous followers

Consider a group of 7 mass-spring-damper system (followers)

$$\dot{\mathbf{x}}^i = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k^i}{m^i} & -\frac{b^i}{m^i} \end{bmatrix}}_{\mathbf{A}^i} \mathbf{x}^i + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m^i} \end{bmatrix}}_{\mathbf{B}^i} u^i, \quad i \in \{1, \ldots, 7\} \tag{2.29}$$

where $\mathbf{x}^i = [x^i \quad \dot{x}^i] \in \mathbb{R}^2$ is the state vector with $x^i \in \mathbb{R}$ and $\dot{x}^i \in \mathbb{R}$ representing the displacement and velocity of the mass, k^i, b^i and m^i are spring constant, damping constant

and mass, respectively, and $u^i \in \mathbb{R}$ is the input force. The system's parameters (k^i, b^i, m^i) for $i \in \{1, \ldots, 7\}$ are $(1, 0.5, 5)$, $(2, 0.5, 15)$, $(2.5, 1.5, 10)$, $(3, 0.8, 8)$, $(3.5, 1.5, 5)$, $(1.2, 1.8, 12)$, and $(0.5, 1, 10)$, respectively. The leader denoted by 0 is a nonlinear mass-spring-damper system

$$\dot{\mathbf{x}}^0 = \begin{bmatrix} \dot{x}^0 \\ \frac{1}{m^0}(u^0 - b^0\dot{x}^0 - k^0 x^0 - 0.6{x^0}^3) \end{bmatrix}, \tag{2.30}$$

where the input u^0 is unknown to the followers and the system parameters $(k^0, b^0, m^0) = (1.2, 2, 5)$. The interaction topology of the systems is shown in Fig. 2.1. Followers 1, 2 and 3 obtain the state of the leader with a sampling rate of 1 per second, i.e., $T_k = 1$ second, $k \in \mathbb{Z}_{\geq 0}$. The followers start at $\mathbf{x}^1(0) = [0 \quad 0]^\top$, $\mathbf{x}^2(0) = [-0.5 \quad 0]^\top$, $\mathbf{x}^3(0) = [-1 \quad 0]^\top$, $\mathbf{x}^4(0) = [-1.5 \quad 0]^\top$, $\mathbf{x}^5(0) = [-2 \quad 0]^\top$, $\mathbf{x}^6(0) = [-2.5 \quad 0]^\top$, $\mathbf{x}^7(0) = [-3 \quad 0]^\top$ in a formation with uniform distance $0.5(m)$ to the previous number of the follower. The objective is for the followers to track the state of the leader while preserving the initial formation of the systems at every sampling time t_k. The follower i only knows the local formation, i.e., $\mathbf{F}^{ij}(0)$ for $j \in \overline{\mathcal{N}}^i_{\text{out}}$. For example, follower 3 knows $\mathbf{F}^{30}(0) = [1 \quad 0]^\top$, $\mathbf{F}^{31}(0) = [1 \quad 0]^\top$, and $\mathbf{F}^{32}(0) = [0.5 \quad 0]^\top$.

The result of implementing the algorithm of Theorem 2.4.1 is shown in Fig. 2.7. The '+' represents the sampled leader states and '$\times$' shows the followers track the leader's state in the desired formation at the next sampled time. In this example interestingly in the transition times similar to what is expected from homogeneous followers the state and input of all the followers are offset-synchronized. However, this property is not necessarily true in general for heterogeneous followers.

To show the practical measures to overcome the high gain challenge mentioned in Remark 2.4.1 when implementing control (2.26). Figures 2.8 and 2.9 are the results of employing the deadzone approach and extended horizon approach, respectively. To observe the implication

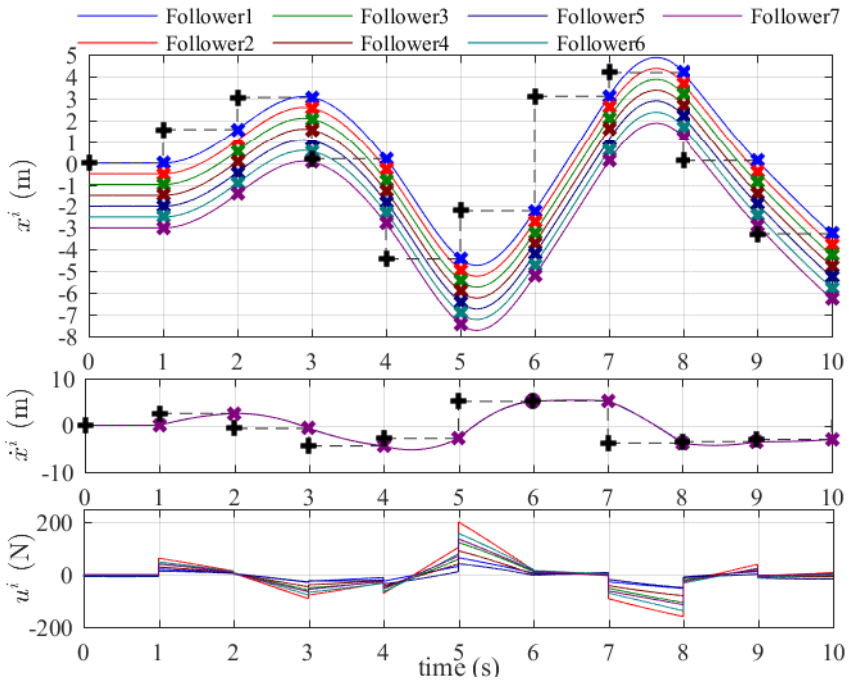

Figure 2.7 – The state and control trajectories of followers of the first numerical example.

of this practical measure more clearly, we simulated a case where the offset of the followers is set to zero (without formation). We can see that with these two solutions, although losing some tracking precision, the followers sill can follow the the sampled state of the leader in a satisfactory level.

Output tracking for a group of aircraft

We consider a group of 7 aircraft whose short-period dynamics is given by (taken from [90, Example 10.1])

$$\underbrace{\begin{bmatrix}\dot{\alpha}\\ \dot{q}\end{bmatrix}}_{\dot{\mathbf{x}}^i} = \underbrace{\begin{bmatrix}-0.0115 & 1\\ -0.0395 & -2.9857\end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix}\alpha\\ q\end{bmatrix}}_{\mathbf{x}^i} + \underbrace{\begin{bmatrix}-0.1601\\ -11.0437\end{bmatrix}}_{\mathbf{B}} \underbrace{\delta_e^i}_{u^i}, \quad y^i(t) = \underbrace{\begin{bmatrix}0 & 1\end{bmatrix}}_{\mathbf{C}} \mathbf{x}^i,$$

where α^i, q^i and δ_e^i are are respectively, angle of attack, pitch rate and elevator angle of aircraft $i \in \{0, \cdots, 6\}$. The interaction topology of these aircraft is shown in Fig. 2.5,

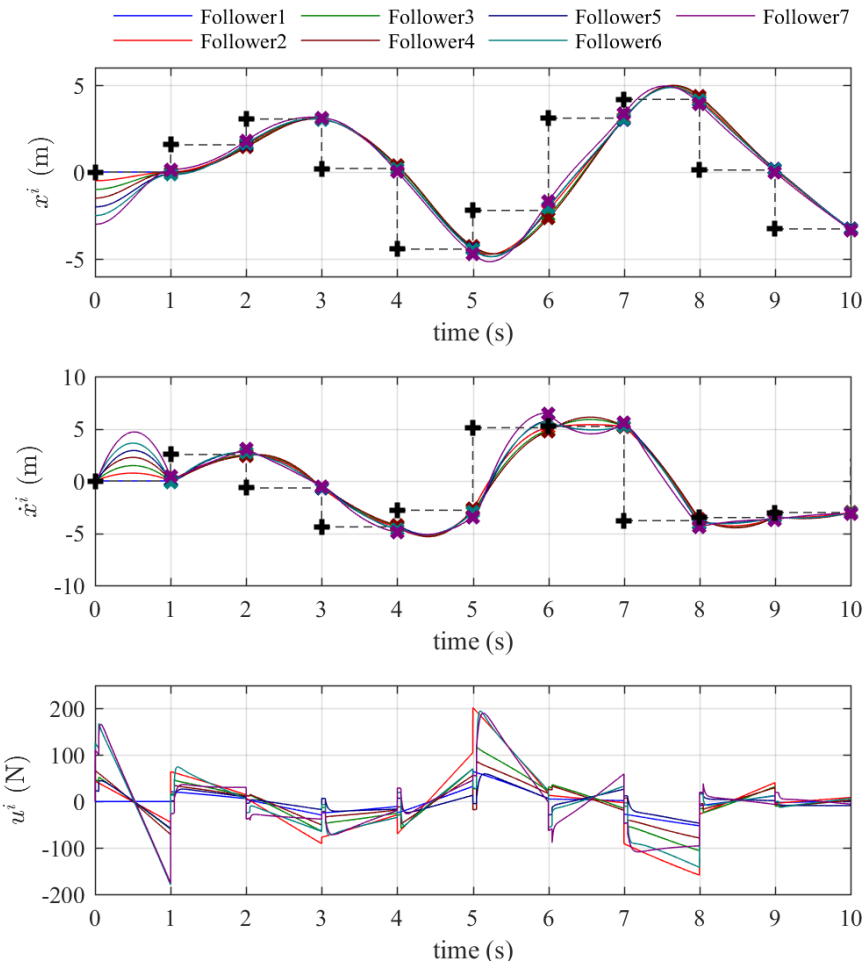

Figure 2.8 – The state and control trajectories of the followers of the first numerical example when we employ the deadzone approach discussed in Remark 4.2 to address the high-gain-challenge. To observe the implication of this practical measure more clearly, we simulated a case where the offset is set to zero.

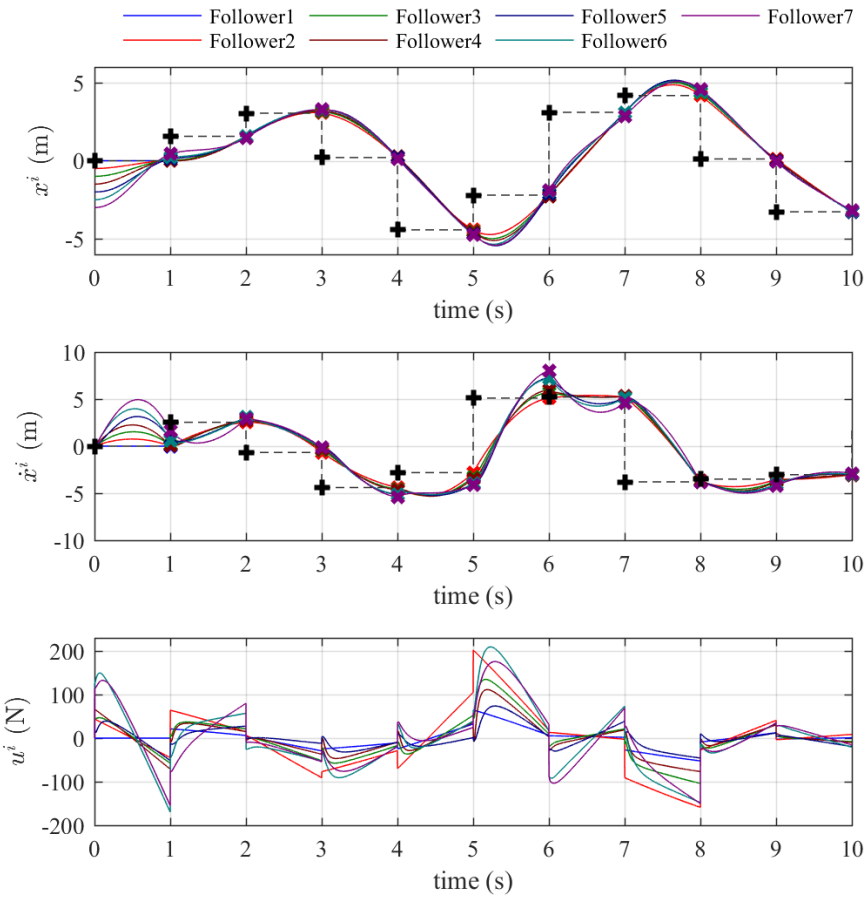

Figure 2.9 – The state and control trajectories of the followers of the first numerical example when we employ the extended horizon approach ($\delta = 0.05$ (s)) discussed in Remark 4.2 to address the high-gain-challenge. To observe the implication of this practical measure more clearly, we simulated a case where the offset is set to zero.

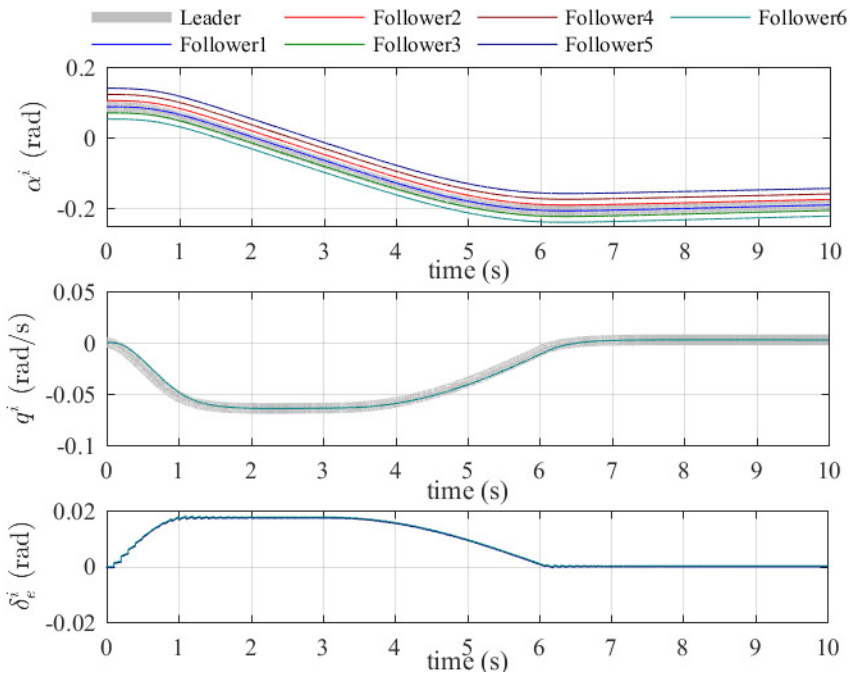

Figure 2.10 – The state and control trajectories of followers of the second numerical example.

where, agent 0 is the leader. For this system $\mathbf{CB} = -11.0437$, therefore the condition of Remark 2.4.3 is satisfied and we can design a distributed algorithm to synchronize the pitch rate of the follower aircraft $\{1, \cdots, 6\}$ to the pitch rate of the leader aircraft when only sampled pitch rate of the leader at every 0.1 seconds is available to the follower aircraft 1. Figure 2.10 demonstrates the results.

2.5 Conclusion

In this chapter, we have proposed the distributed leader-following algorithms for homogeneous and heterogeneous multi-agent systems with an active leader with unknown input. We have proved that our distributed leader-following algorithm for the linear followers steers the group to be at the sampled states of the leader at the specified arrival times in a specified formation. We showed that the control input of each follower agent between the sampling times is a minimum energy control. We also showed that after the first sampling epoch, the states

of all the homogeneous follower agents are synchronized with each other. We demonstrated our results via homogeneous mobile agents with second order integrator dynamics and unicycle dynamics, leader-following problems of heterogeneous mass-spring-damper systems and aircraft.

Chapter 3

Distributed containment control

3.1 Introduction

In this chapter, we propose a distributed containment control algorithm for a group of agents, which communicate over a strongly connected and weight-balanced directed graph. The agents jointly detect a group of moving leaders in a periodic sampling time. Each agent detects a subset (could be empty) of the leaders and computes the geometric center of the subset. Then, the discrete-time containment control algorithm based on the dynamic average consensus algorithm is developed to track the average of the geometric centers which we show is in the convex hull of the observed leaders. Therefore, even though some of the agents do not observe any leaders, they can still track the convex hull. Note that, there is no assumption about the leaders' dynamics (they can be heterogeneous) and the agents only measure the positions of the leaders. Next, we show our proposed containment control algorithm can be further apply to a group of unicycle mobile robots to track the convex hull of the moving leaders. The control scheme combines the discrete-time containment control algorithm as a observer to estimate the average position of the geometry centers in the convex

hull and a local finite-time control to track their estimate, such that the control scheme can drive the robots to the convex hull in time. In our proposed framework, the unicycle robots have continuous-time dynamics but communicate with each other in discrete-time fashion. A numerical example is demonstrated to show the efficiency of this scheme.

The rest of this chapter is organized as follows. Section 3.2 introduces our basic notation, graph-theoretic definitions and notions and reviews the dynamic average consensus algorithm, which we use in our developments. Section 3.3 gives our problem definition and objective statement. Section 3.4 presents our main result on design of a distributed containment control algorithm. Section 3.5 applies the containment control scheme for unicycle robots. Section 3.6 gives a numerical example to demonstrate our results. Section 3.7 concludes the results of this chapter.

3.2 Notations and Preliminaries

Notation: We let $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Z}$, and $\mathbb{Z}_{\geq a}$ denote the set of real, positive real, non-negative real, integer, and integer numbers greater than $a \in \mathbb{Z}$, respectively. The transpose of a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ is $\mathbf{A}^\top$. For $\mathbf{s} \in \mathbb{R}^d$, $\|\mathbf{s}\| = \sqrt{\mathbf{s}^\top \mathbf{s}}$ denotes the standard Euclidean norm. For a given set of points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$ in a Euclidean space, their convex hull is $\mathrm{Co}(\mathcal{X}) = \{\mathbf{q} \in \mathbb{R}^n | \mathbf{q} = \sum_{j=1}^M \alpha_j \mathbf{x}_j, \alpha \geq 0, \sum_{j=1}^M \alpha_j = 1\}$, which is the smallest convex set containing all the points in $\mathcal{X}$. In a network of N agents, to distinguish and emphasis that a variable is local to an agent $i \in \{1, \ldots, N\}$, we use superscripts, e.g., $\mathbf{x}^i$ is the local variable of agent i. Moreover, if $\mathbf{r}^i \in \mathbb{R}^{n^i}$ is a variable of agent $i \in \mathcal{V} = \{1, \cdots, N\}$, the aggregated $\mathbf{r}^i$'s of the network is the vector $\mathbf{r} = [\{\mathbf{r}^i\}_{i\in\mathcal{V}}] = [{\mathbf{r}^1}^\top, \cdots, {\mathbf{r}^N}^\top]^\top \in \mathbb{R}^m$, $m = \sum_{i=1}^N n^i$.

Graph theoretic notations and definitions: Here we review our graph related notations and relevant definitions and concepts from graph theory following [49]. A *digraph*, is a triplet

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{1, \dots, N\}$ is the *node set* and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the *edge set*, and $\mathbf{A} = [\mathsf{a}_{ij}] \in \mathbb{R}^{N \times N}$ is the *adjacency* matrix of the graph defined according to $\mathsf{a}_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $\mathsf{a}_{ij} = 0$, otherwise. An edge (i, j) from i to j means that agent j can send information to agent i. Here, i is called an *in-neighbor* of j and j is called an *out-neighbor* of i. A *directed path* is a sequence of nodes connected by edges. The *out-degree* of a node i is $\mathsf{d}^i_{\text{out}} = \Sigma_{j=1}^N \mathsf{a}_{ij}$. We let $\mathsf{d}^{\max} = \max\{\mathsf{d}^i_{\text{out}}\}_{i=1}^N$. The out-degree matrix of a graph is $\mathbf{D}_{\text{out}} = \text{Diag}(\mathsf{d}^1_{\text{out}}, \mathsf{d}^2_{\text{out}}, \cdots, \mathsf{d}^N_{\text{out}})$. The *(out-) Laplacian* matrix is $\mathbf{L} = \mathbf{D}_{\text{out}} - \mathbf{A}$. Note that $\mathbf{L}\mathbf{1}_N = 0$. A weighted digraph $\mathcal{G}$ is weight-balanced if and only if $\mathbf{1}_N^\top \mathbf{L} = 0$. Based on the structure of $\mathbf{L}$, at least one of the eigenvalues of $\mathbf{L}$ is zero and the rest of them have nonnegative real parts.

Average dynamic consensus algorithm: In our developments we will use a dynamic average consensus algorithm (see [1]) as described in the lemma below .

Lemma 3.2.1 (Dynamic average consensus algorithm)**.** *Let* $\mathcal{G}(\mathcal{V}, \mathcal{E})$ *be a strongly connected and weight-balanced digraph of* N *agents. Assume each agent* $i \in \mathcal{V}$ *has access to a dynamic input* $\mathbf{r}^i(k) = \mathbf{r}^i(t_k)$ *at time* $t_k = k\delta$, $\delta \in \mathbb{R}_{>0}$, $k \in \mathbb{Z}_{\geq 0}$. *For* $\delta \in (0, \beta^{-1}(\mathsf{d}^{\max})^{-1})$ *, where* $\beta \in \mathbb{R}_{>0}$, *if each agent* $i \in \mathcal{V}$ *implements*

$$\mathbf{p}^i(k+1) = \mathbf{p}^i(k) + \delta\beta \sum_{j=1}^N \mathsf{a}_{ij}(\mathbf{x}^i(k) - \mathbf{x}^j(k)), \tag{3.1a}$$

$$\mathbf{x}^i(k) = \mathbf{r}^i(k) - \mathbf{p}^i(k), \tag{3.1b}$$

starting at $\mathbf{p}^i(0) = \mathbf{0}$, *then the trajectory* $k \mapsto \mathbf{p}^i(k)$ *of each agent* $i \in \mathcal{V}$ *is bounded and satisfies*

$$\lim_{k \to \infty} \left\| \mathbf{x}^i(k) - \frac{1}{N} \sum_{j=1}^N \mathbf{r}^j(k) \right\| \leq \frac{\gamma(\infty)\, \delta}{\beta \hat{\lambda}_2}, \tag{3.2}$$

where $\sup_{\bar{k} \in \mathbb{Z}_{\bar{k} \geq k}} \|(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top)(\mathbf{r}(\bar{k}+1) - \mathbf{r}(\bar{k}))\| = \gamma(k) < \infty$, *and* $\hat{\lambda}_2$ *is second smallest*

eigenvalue of $\mathbf{L}$.

3.3 Problem definition

In this section, we formalize our distributed containment control problem of interest. We assume that a group of M (M can change with time) mobile leaders are moving with a bounded velocity on a $\mathbb{R}^2$ or $\mathbb{R}^3$ space. We let $\mathbf{x}_{L,j}(t)$ be the position vector of leader $j \in \{1,\dots,M\}$ at time $t \in \mathbb{R}_{\geq 0}$. In our setting, a set of networked mobile agents $\mathcal{V} = \{1,\cdots,N\}$ monitor the leaders. The communication topology $\mathcal{G}$ of the agents is a strongly connected and weight-balanced digraph and the agents can communicate at discrete-times $t_k = k\delta_c$, $k \in \mathbb{Z}_{\geq 0}$, $\delta_c \in \mathbb{R}_{>0}$. The agents sample the leaders at sampling times $t_k^s = k\delta_s$, $k \in \mathbb{Z}_{\geq 0}$, $\delta_s \in \mathbb{R}_{>0}$. We let $\mathcal{V}_L^i(t_k^s)$ be the set of leaders observed by agent $i \in \mathcal{V}$ at sampling time t_k^s. Between each sampling time, agent $i \in \mathcal{V}$ uses $\mathbf{x}_{L,j}(t) = \mathbf{x}_{L,j}(t_k^s)$ and $\mathcal{V}_L^i(t) = \mathcal{V}_L^i(t_k^s)$, $t \in [t_k^s, t_{k+1}^s)$, $k \in \mathbb{Z}_{\geq 0}$, $j \in \mathcal{V}_L^i(t_k^s)$. At every sampling time $t_k^s \in \mathbb{R}_{\geq 0}$, we let $\mathcal{V}_L(t_k^s)$ be the set of the mobile leaders that are observed jointly by the agents $\mathcal{V}$, i.e., $\mathcal{V}_L(t_k^s) = \cup_{i=1}^N \mathcal{V}_L^i(t_k^s)$ (see Fig. 3.1). We let $\mathcal{V}_a(t_k^s) \subset \mathcal{V}$ be the set of the agents that observe at least one leader at t_k^s, $k \in \mathbb{Z}_{\geq 0}$; we assume that $\mathcal{V}_a(t_k^s) \neq \emptyset$. Our objective in this paper is to design a distributed control algorithm that enables each agent $i \in \mathcal{V}$ to derive its local state $\boldsymbol{\chi}^i$ to asymptotically track $\mathrm{Co}(\mathcal{V}_L(t_k^s))$, the convex hull of the set of the location of the leaders $\mathcal{V}_L(t_k^s)$ with a bounded error $e \geq 0$ (to simplify notation, we wrote $\mathrm{Co}(\{\mathbf{x}_{L,j}(t)\}_{j\in\mathcal{V}_L(t)})$ as $\mathrm{Co}(\mathcal{V}_L(t))$). We state our objective as

$$\|\boldsymbol{\chi}^i(t_k) - \bar{\mathbf{x}}_L(t_k)\| \leq e, \quad i \in \mathcal{V}. \tag{3.3}$$

where $\bar{\mathbf{x}}_L(t_k) \in \mathrm{Co}(\mathcal{V}_L(t_k))$. We assume that the agents have no knowledge about the motion model of the leaders. As the information of each agent takes some time to propagate through the network, tracking the convex hull of an arbitrarily fast moving leader set with zero error

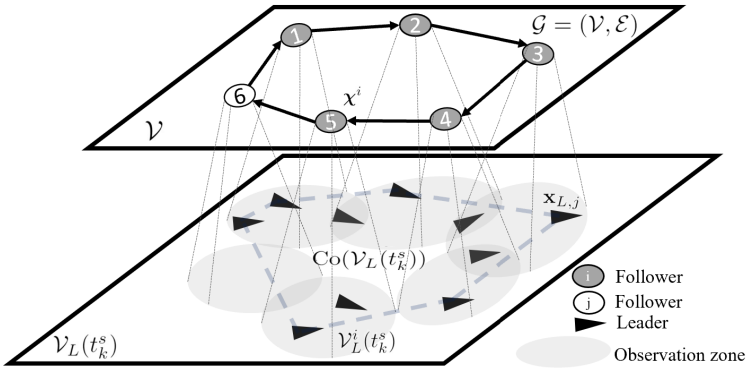

Figure 3.1 – The agent set and the leader set. The ellipsoids show the observation zone of the observing agents.

is not feasible unless agents have some a priori information about the dynamics generating the signals. Also, how fast the information travels across the network $\mathcal{G}$ depends on the connectivity of $\mathcal{G}$. Interestingly, as expected, we will show that the size of the error e is going to be a function of the bound on the velocity of the leaders and $\hat{\lambda}_2$, which is a measure of connectivity of the network. We will also show that when the leader set is stationary and observed by the same set of agents, the agents converge to a point in the convex hull of the leader set, i.e., $e = 0$.

3.4 Containment control algorithm

In this section, we present our solution for the distributed containment problem stated in Section 3.3. If the number of the leaders is equal to the number of the agents, i.e., $|\mathcal{V}_L(t_k^s)| = N$, and $\mathcal{V}_L^i(t_k^s) \cap \mathcal{V}_L^j(t_k^s) = \emptyset$ for any $i, j \in \mathcal{V}$, $i \neq j$, a simple solution to our containment problem of interest is to implement the dynamic average consensus algorithm (3.1) with $\mathbf{r}^i(k) = \sum_{j \in \mathcal{V}_L^i(t_k)} \mathbf{x}_{L,j}(t_k)$ for $i \in \mathcal{V}$ (when $\mathcal{V}_L^i(t_k) = \emptyset$, we use $\mathbf{r}^i(k) = \mathbf{0}$). This is because, in this scenario, as certified by Lemma 3.2.1, the agents track the geometric center of the leader set, which is a point in the convex hull of the leader set, with a bounded error. However, in general, as the agents monitor the leader set, it is likely that the observed leader sets

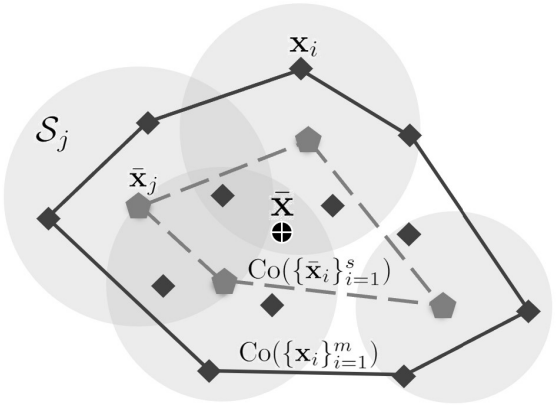

Figure 3.2 – Graphical demonstration of Lemma 3.4.1 for an example case.

$\mathcal{V}_L^i(t_k^s)$, $i \in \mathcal{V}$ of the agents have overlap and also the number of the leaders be different than the number of the agents, see Fig 3.1. In what follows, we present a simple solution for the containment problem that works for the general case.

To present our solution, we first make the following key observation about the convex hull of a set of points $\{\mathbf{x}_i\}_{i=1}^m$ in an Euclidean space.

Lemma 3.4.1 (Auxiliary result on convex hull of a set of points)**.** . *Consider a set of points* $\{\mathbf{x}_i\}_{i=1}^m$ *in* $\mathbb{R}^2$ *or* $\mathbb{R}^3$*. Let* $\mathcal{S}_j \neq \emptyset$*,* $j \in \{1, \cdots, r\}$*, be a subset of* $\{1, \cdots, m\}$*. Let* $\bar{\mathbf{x}}_j = \frac{\sum_{k \in \mathcal{S}_j} \mathbf{x}_k}{|\mathcal{S}_j|}$*,* $j \in \{1, \cdots, r\}$*. Then, the point*

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^r \bar{\mathbf{x}}_i}{r} \tag{3.4}$$

is a point in $\mathrm{Co}(\{\mathbf{x}_j\}_{j=1}^m)$*.*

Proof. It is straightforward to confirm that $\bar{\mathbf{x}}_j \in \mathrm{Co}(\{\mathbf{x}_i\}_{i \in \mathcal{S}_j})$, $j \in \{1, \cdots, s\}$ and $\bar{\mathbf{x}} \in \mathrm{Co}(\{\bar{\mathbf{x}}_i\}_{i=1}^s)$ (recall the definition of the convex hull). Moreover, since $\mathrm{Co}(\{\mathbf{x}_k\}_{k=1}^m)$ is a convex set, we note that $\mathrm{Co}(\{\mathbf{x}_i\}_{i \in \mathcal{S}_j}) \subset \mathrm{Co}(\{\mathbf{x}_k\}_{k=1}^m)$, $j \in \{1, \cdots, s\}$. Thus, for $i \in \{1, \cdots, s\}$, $\bar{\mathbf{x}}_i \in \mathrm{Co}(\{\mathbf{x}_j\}_{j=1}^m)$, and $\mathrm{Co}(\{\bar{\mathbf{x}}_i\}_{i=1}^s) \subset \mathrm{Co}(\{\mathbf{x}_i\}_{i=1}^m)$. As a result, $\bar{\mathbf{x}} \in \mathrm{Co}(\{\mathbf{x}_j\}_{j=1}^m)$. □

We note here that in Lemma 3.4.1, $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$, $i, j \in \{1, \cdots, r\}$, is not required. An example

case that demonstrates the result of Lemma 3.4.1 is shown in Fig. 3.2.

Now for the containment problem, consider the general case when $0 < |\mathcal{V}_a(t_k^s)| \leq N$, and for any two distinct agents $i, j \in \mathcal{V}_a(t_k^s)$, $\mathcal{V}_L^i(t_k^s) \cap \mathcal{V}_L^j(t_k^s)$ is not necessarily empty. Then, in light of Lemma 3.4.1, we know that

$$\bar{\mathbf{x}}_L(t_k^s) = \frac{\sum_{i=1}^N \mathbf{r}^i(t_k^s)}{|\mathcal{V}_a(t_k^s)|} \in \mathrm{Co}(\mathcal{V}_L(t_k^s)), \tag{3.5}$$

where

$$\mathbf{r}^i(t_k^s) = \begin{cases} \frac{\sum_{j \in \mathcal{V}_L^i(t_k^s)} \mathbf{x}_{L,j}(t_k^s)}{|\mathcal{V}_L^i(t_k^s)|}, & i \in \mathcal{V}_a(t_k^s), \\ \mathbf{0}, & i \in \mathcal{V} \backslash \mathcal{V}_a(t_k^s). \end{cases} \tag{3.6}$$

Based on this observation, we propose the following distributed solution for our containment problem of interest. Our solution uses two dynamic average consensus algorithms of the form (3.1), to obtain the numerator and the denominator of $\bar{\mathbf{x}}_L$ in (3.5).

Theorem 3.4.1 (Distributed containment control algorithm)**.** *Let the communication topology of the agents $\mathcal{V}$ be a strongly connected and weight-balanced digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Assume that $\mathcal{V}_a(t_k^s) \neq \emptyset$ at each sampling time t_k^s, $k \in \mathbb{Z}_{\geq 0}$. Let $\mathbf{r}^i(t) = \mathbf{r}^i(t_k^s)$, $i \in \mathcal{V}$, and $\bar{\mathbf{x}}_L(t) = \bar{\mathbf{x}}_L(t_k^s)$ for $t \in [t_k^s, t_{k+1}^s)$, where $\mathbf{r}^i(t_k^s)$ and $\bar{\mathbf{x}}_L(t_k^s)$ are given respectively, in* (3.5) *and* (3.6)*. Moreover, let $\bar{r}^i(t_k) = 1$ if $i \in \mathcal{V}_a(t_k)$, and $\bar{r}^i(t_k) = 0$ if $i \in \mathcal{V} \backslash \mathcal{V}_a(t_k)$. Suppose that $\sup_{\bar{k} \in \mathbb{Z}_{\geq k}} \|\mathbf{r}(t_{\bar{k}+1}) - \mathbf{r}(t_{\bar{k}})\| = \gamma(t_k) < \infty$, and $\sup_{\bar{k} \in \mathbb{Z}_{\geq k}} \|\bar{\mathbf{r}}(t_{\bar{k}+1}) - \bar{\mathbf{r}}(t_{\bar{k}})\| = \bar{\gamma}(t_k) < \infty$. Assume that each agent $i \in \mathcal{V}$ implements the distributed algorithm*

$$\mathbf{p}^i(t_{k+1}) = \mathbf{p}^i(t_k) + \delta_c \beta \sum_{j=1}^N \mathsf{a}_{ij}(\mathbf{w}^i(t_k) - \mathbf{w}^j(t_k)), \tag{3.7a}$$

$$\mathbf{w}^i(t_k) = \mathbf{r}^i(t_k) - \mathbf{p}^i(t_k), \tag{3.7b}$$

$$q^i(t_{k+1}) = q^i(t_k) + \delta_c \beta \sum_{j=1}^N \mathsf{a}_{ij}(z^i(t_k) - z^j(t_k)), \tag{3.7c}$$

$$z^i(t_k) = \bar{r}^i(t_k) - q^i(t_k), \tag{3.7d}$$

$$\boldsymbol{\chi}^i(t_k) = \frac{\mathbf{w}^i(t_k)}{\max\{\epsilon, |z^i(t_k)|\}}, \tag{3.7e}$$

$\beta \in \mathbb{R}_{>0}$, initialized at $\mathbf{p}^i(t_0) = \mathbf{0}$ and $q^i(t_0) = 0$, with a communication stepsize $\delta_c \in (0, \beta^{-1}(\mathsf{d}^{\max})^{-1})$ ($t_k = \delta_c k$, $k \in \mathbb{Z}_{\geq 0}$). Here, $0 < \epsilon < 1/N$ is a small positive real number. Then, there exists a bounded value $e \in \mathbb{R}_{>0}$ such that

$$\|\boldsymbol{\chi}^i(t_k) - \bar{\mathbf{x}}_L(t_k)\| \leq e, \quad t_k \in \mathbb{R}_{\geq 0}, \quad i \in \mathcal{V}, \tag{3.8}$$

Moreover, if for a finite $\bar{k} \in \mathbb{Z}_{\geq 0}$ we have $\bar{\gamma}(t_k) = 0$ for all $k \in \mathbb{Z}_{\geq \bar{k}}$, i.e., the set of observing agents is not changing with time for $t_k \geq t_{\bar{k}}$, we have

$$\lim_{k\to\infty} \|\boldsymbol{\chi}^i(t_k) - \bar{\mathbf{x}}_L(t_k)\| \leq \frac{N\gamma(\infty)\,\delta_c}{|\mathcal{V}_a(t_{\bar{k}})|\,\beta\hat{\lambda}_2}, \quad i \in \mathcal{V}. \tag{3.9}$$

Proof. Given the initial conditions and stated bounds on $\mathbf{r} = [\{\mathbf{r}^i\}_{i\in\mathcal{V}}]$ and $\bar{\mathbf{r}} = [\{\bar{r}^i\}_{i\in\mathcal{V}}]$, by invoking Lemma 3.2.1, we conclude that for $k \in \mathbb{Z}_{\geq 0}$ the trajectories $k \mapsto \mathbf{w}^i(t_k)$ and $k \mapsto z^i(t_k)$, $i \in \mathcal{V}$ are bounded and satisfy

$$\lim_{k\to\infty} \frac{|\mathcal{V}_a(t_k)|}{N} \left\| \mathbf{w}^i(t_k) / \frac{N}{|\mathcal{V}_a(t_k)|} - \bar{\mathbf{x}}_L(t_k) \right\| =$$

$$\lim_{k\to\infty} \left\| \mathbf{w}^i(t_k) - \frac{1}{N} \sum_{j\in\mathcal{V}_a(t_k)} \mathbf{r}^j(t_k) \right\| \leq \frac{\gamma(\infty)\,\delta_c}{\beta\hat{\lambda}_2}, \tag{3.10a}$$

$$\lim_{k\to\infty} \left| z^i(t_k) - \frac{|\mathcal{V}_a(t_k)|}{N} \right| \leq \frac{\bar{\gamma}(\infty)\,\delta_c}{\beta\hat{\lambda}_2}. \tag{3.10b}$$

Therefore, we can conclude that $\|\boldsymbol{\chi}^i(t_k)\|$ is finite for all $k \in \mathbb{Z}_{\geq 0}$, which confirms also (3.8). Next, if for a finite $\bar{k} \in \mathbb{Z}_{\geq 0}$ we have $\bar{\gamma}(t_k) = 0$ for all $k \in \mathbb{Z}_{\geq \bar{k}}$, we note that from (3.10b) we obtain $\lim_{k\to\infty} z^i(t_k) = \frac{|V_o(t_{\bar{k}})|}{N}$, $i \in \mathcal{V}$. Consequently, $\lim_{k\to\infty} \boldsymbol{\chi}^i(t_k) = \frac{N}{|V_o(t_{\bar{k}})|} \lim_{k\to\infty} \mathbf{w}^i(t_k)$, which along with (3.10a) confirms (3.9).

□

It is worth nothing that if in addition to the set of observing agents not changing with time for $t_k \geq t_{\bar{k}}$ for all $k \geq \bar{k} \geq 0$, we also have $\gamma(t_k) = 0$ for $k \geq \hat{k} \geq \bar{k}$ (every $i \in \mathcal{V}_a(t_k)$ observes the same leaders for $t \geq t_{\hat{k}}$ and the leaders converge to a stationary configuration), it follows from (3.9) that $\lim_{k\to\infty} \boldsymbol{\chi}^i(t_k) = \bar{\mathbf{x}}_L(\infty)$, $i \in \mathcal{V}$. We also note here that use of a dynamic average consensus algorithm in constructing the distributed containment controller (3.7) results in a better tracking performance than use of a static average consensus ([7]) that gets re-initialized at each sampling time with the new observed inputs. For more details see the example scenario discussed in Figure 2 of [1], which compares performance of a dynamic average consensus algorithm and a static average consensus algorithm for sampled time-varying input signals.

3.5 An application example: containment problem for a group of networked unicycle robots

In this section, we demonstrate how a group of N unicycle robots with strongly connected and weight-balanced topology can track the convex hull of a set of leaders that they observe. Let the dynamics of each robot be expressed by

$$\mathbf{x}^i = \begin{bmatrix} \dot{x}^i \\ \dot{y}^i \\ \dot{\theta}^i \end{bmatrix} = \begin{bmatrix} v^i \cos\theta^i \\ v^i \sin\theta^i \\ \omega^i \end{bmatrix}, \qquad i \in \mathcal{V}, \tag{3.11}$$

where x^i, $y^i \in \mathbb{R}$ are the coordinates in 2 dimensional space and $\theta^i \in \mathbb{R}$ is the heading angle of the robot. v^i, $\omega^i \in \mathbb{R}$ are, respectively, the linear velocity and angular velocity of robot

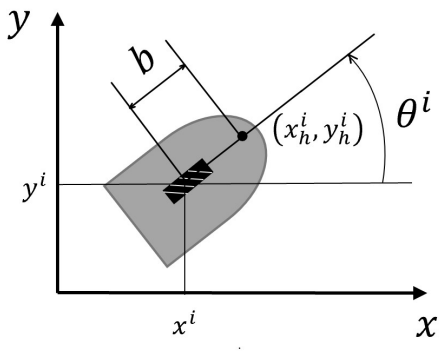

Figure 3.3 – A unicycle robot and its corresponding state variables.

$i \in \mathcal{V}$. We define a head point

$$\mathbf{x}_h^i = \begin{bmatrix} x_h^i \\ y_h^i \end{bmatrix} = \begin{bmatrix} x^i + b\cos\theta^i \\ y^i + b\sin\theta^i \end{bmatrix}, \qquad i \in \mathcal{V}. \tag{3.12}$$

as a position at a distance b along the main axis of the robot i as shown in Fig. 3.3. The head point can be the place where the robot observation sensor (e.g., camera) is located or a point that carry sensitive goods. There is another group of mobile robots called leaders, which have the ability to avoid obstacles. These unicycle robots want to track the leaders and keep their head points inside the convex hull formed by the leaders. The unicycle robots communicate to their neighbors at the instant t_k, $k \in \mathbb{Z}_{\geq 0}$ with the time period δ_c and jointly detect the position of the leaders $\mathbf{x}_{L,j}$, $j \in \mathcal{V}_L(t_k^s)$ at the instant t_k^s with time period δ_s, but they do not have the information of the leaders' dynamics.

To control the unicycle robots to stay in the convex hull $\mathrm{Co}(\mathcal{V}_L(t_k^s))$ of the moving leaders, we propose a two-layer containment control scheme. The first layer is a distributed observer with the discrete-time process of (3.7) proposed in Theorem 3.4.1 which produces a estimated position $\boldsymbol{\chi}^i(t_k)$ of a pin $\bar{\mathbf{x}}_L \in \mathrm{Co}(\mathcal{V}_L(t_k^s)))$ at every discrete communication time t_k. By this distributed observer, the robots can estimate the position of the pin in the convex hull even though some of them could not detect any leader. Then, in order to track the discrete estimate $\boldsymbol{\chi}^i(t_k)$ in time, we use a local finite-time converging controller which drives the head

point of the continuous-time unicycle robot to track $\boldsymbol{\chi}^i(t_k)$ before the next communication time t_{k+1}. That is

$$\mathbf{x}_h^i(t_{k+1}) = \boldsymbol{\chi}^i(t_k), \qquad i \in \mathcal{V}. \tag{3.13}$$

The following result gives the local tracking control that realizes the objective (3.13).

Theorem 3.5.1 (Finite-time converging controller to track the convex hull of the leaders)**.** *Consider a group of N unicycle robots with dynamics described by* (3.11) *and the head point defined by* (3.12)*. Let the communication topology of the robots be a strongly connected and weight-balanced digraph and suppose the robots are implementing the containment controller* (3.7)*. Starting at an initial condition $\mathbf{x}^i(t_0) \in \mathbb{R}^3$, let for $t \in [t_k, t_{k+1})$*

$$v^i(t) = u_1^i \cos\theta^i + u_2^i \sin\theta^i, \tag{3.14a}$$

$$\omega^i(t) = \frac{u_2^i \cos\theta^i - u_1^i \sin\theta^i}{b}, \tag{3.14b}$$

$$\mathbf{u}^i(t) = \begin{bmatrix} u_1^i \\ u_2^i \end{bmatrix} = \frac{1}{\delta_c}(\boldsymbol{\chi}^i(t_k) - \mathbf{x}_h^i(t_k)). \tag{3.14c}$$

Then, for every robot $i \in \mathcal{V}$, we have $\lim_{t \to t_{k+1}^-} \mathbf{x}_h^i(t) = \boldsymbol{\chi}^i(t_k)$ for all $k \in \mathbb{Z}_{\geq 0}$.

Proof. Note that

$$\dot{\mathbf{x}}_h^i = \begin{bmatrix} \dot{x}^i - b\dot{\theta}^i \sin\theta^i \\ \dot{y}^i + b\dot{\theta}^i \cos\theta^i \end{bmatrix}, \qquad i \in \mathcal{V}.$$

Substituting for robot dynamics from (3.11) and using the velocity inputs (3.14a) and (3.14b),

we arrive at (see [91])

$$\dot{\mathbf{x}}_h^i = \begin{bmatrix} u_1^i \\ u_2^i \end{bmatrix}. \tag{3.15}$$

Then, using the control input (3.14c), we obtain

$$\mathbf{x}_h^i(t) = \mathbf{x}_h^i(t_k) + \frac{t - t_k}{\delta_c}(\boldsymbol{\chi}^i(t_k) - \mathbf{x}_h^i(t_k)), \quad t \in [t_k, t_{k+1}),$$

for $i \in \mathcal{V}$, which confirms that $\lim_{t \to t_{k+1}^-} \mathbf{x}_h^i(t) = \boldsymbol{\chi}^i(t_k)$ for all $k \in \mathbb{Z}_{\geq 0}$.

□

3.6 Numeral demonstration

In this section, we use a numerical example to demonstrate the performance of the distributed containment control algorithm (3.7) and the local tracking controller (3.14). We consider a group of 6 unicycle robots with dynamics (3.11) whose communication topology is given in Fig. 3.4. A position defined by (3.12) is the head point for each robot. The robots aim to make their head points to track the convex hull that formed by 10 mobile leaders with unknown dynamics, moving in a 2 dimensional flat space. The robots detect the leaders at the frequency 0.5 Hz, i.e., $\delta_s = 2$ seconds. The observed leaders by the robots are time varying as described below (time intervals are in second):

- $0 \leq t_k^s < 5$: $\mathcal{V}_L^1(t_k^s) = \{1, 4, 6, 8\}$, $\mathcal{V}_L^2(t_k^s) = \{2, 4, 7, 8, 10\}$, $\mathcal{V}_L^3(t_k^s) = \{3, 4, 5, 9\}$, $\mathcal{V}_L^4(t_k^s) = \emptyset$, $\mathcal{V}_L^5(t_k^s) = \{1, 3, 9\}$ and $\mathcal{V}_L^6(t_k^s) = \emptyset$,

- $5 \leq t_k^s < 10$: $\mathcal{V}_L^1(t_k^s) = \{3, 5, 6, 8\}$, $\mathcal{V}_L^2(t_k^s) = \{1, 2, 7, 9, 10\}$, $\mathcal{V}_L^3(t_k^s) = \{3, 4, 5, 9\}$, $\mathcal{V}_L^4(t_k^s) = \emptyset$, $\mathcal{V}_L^5(t_k^s) = \{1, 3, 9\}$ and $\mathcal{V}_L^6(t_k^s) = \{2, 5, 7, 9\}$,

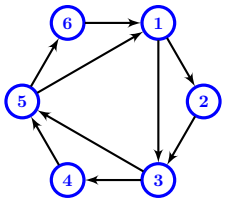

Figure 3.4 – A strongly connected and weight-balance topology with edge weights of 0 and 1.

- $10 \leq t_k^s \leq 20$: $\mathcal{V}_L^1(t_k^s) = \{1, 2, 5, 8\}$, $\mathcal{V}_L^2(t_k^s) = \{2, 3, 6, 7, 10\}$, $\mathcal{V}_L^3(t_k^s) = \{3, 4, 5, 9\}$, $\mathcal{V}_L^4(t_k^s) = \{3, 10\}$, $\mathcal{V}_L^5(t_k^s) = \{1, 3, 9\}$ and $\mathcal{V}_L^6(t_k^s) = \{2, 5, 7, 9\}$.

The robots implement their distributed containment observer (3.7) at the communication frequency of 2 Hz, i.e., $\delta_c = 0.5$ seconds, to estimate $\bar{\mathbf{x}}_L(t_k)$ which is in the convex hull formed by the leaders. The time varying convex hull of the observed leaders are shown by the red closed curves in Fig. 3.5 and Fig. 3.6. These Figures along Fig. 3.7 also show that for the given scenario our proposed tracking control scheme achieves its tracking goal satisfactorily. Fig. 3.8 and Fig. 3.9 show the performance of our containment controller when robots communicate in a higher frequency with $\delta_c = 0.1$ seconds. As we can see, the containment observer converges faster in every sampling interval and the tracking error start to diminish over time. It is very likely that the δ_s is much bigger than δ_c. For such cases, the containment observer results shown in Fig. 3.7 and Fig. 3.9 suggest that instead of deriving the local states to satisfy (3.13) we can use a lower tracking frequency to avid the transient perturbation at the beginning of each sampling time.

3.7 Conclusion

In this chapter, we first propose the algorithm to solve the containment control problem for a group of agents that are communicating in discrete-time over a strongly connected and weight-balance graph. By this containment control algorithm, the group of the agents can track the convex hull of the dynamic leaders detected in the sampling times. Then, the archi-

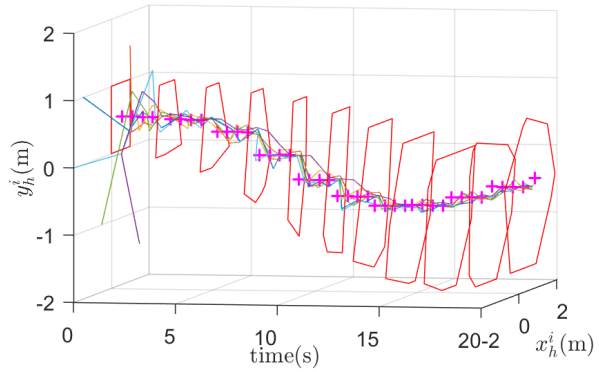

Figure 3.5 – The tracking performance of the robots while implementing the distributed containment observer (3.7) and the local control (3.14) with $\delta_c = 0.5$ seconds: the lines show the trajectory of (x_h^i, y_h^i) vs. time, while "+" show the location of $\bar{\mathbf{x}}_L(t_k)$ of the leaders. The red polygons indicate the convex hull formed by the moving leaders at each sampling time.

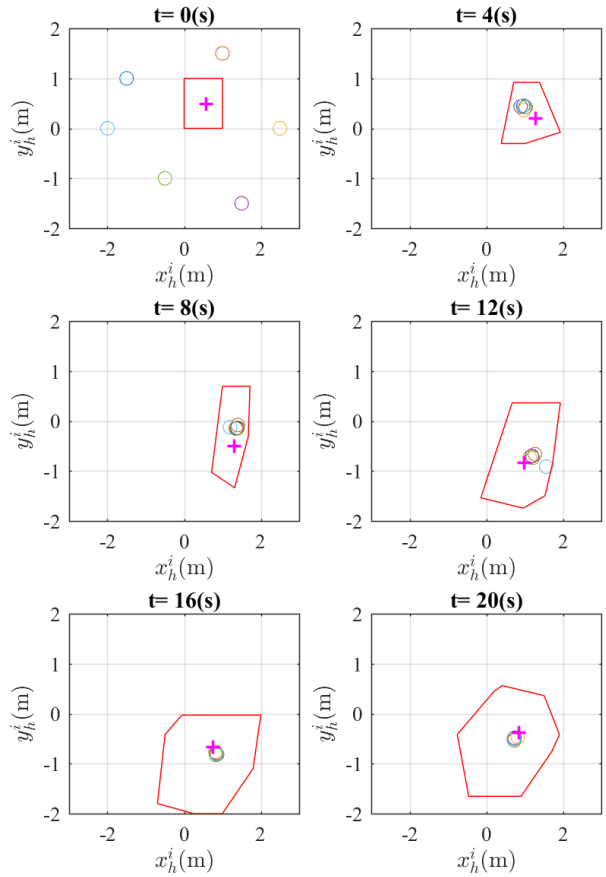

Figure 3.6 – The snapshots showing the leaders convex hull (red polygons), the location of the head point (x_h^i, y_h^i) of the robots ("o" markers) and the location of $\bar{\mathbf{x}}_L(t_k)$ ("+" marker), when robots implement the distributed containment observer (3.7) and the local control (3.14) with $\delta_c = 0.5$ seconds.

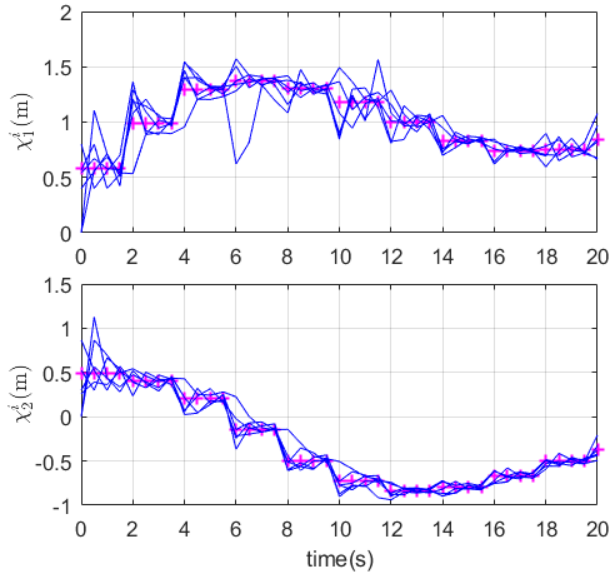

Figure 3.7 – Time history of the output of the containment observer (3.7) with $\delta_c = 0.5$ seconds: the blue curves show $\boldsymbol{\chi}^i$ and the markers "+" show the location of the coordinates of $\bar{\mathbf{x}}_L(t_k)$. The jumps in the location of $\bar{\mathbf{x}}_L(t_k)$ is due to the motion of the leaders and also the changes in the set of the observed leaders by each agent.

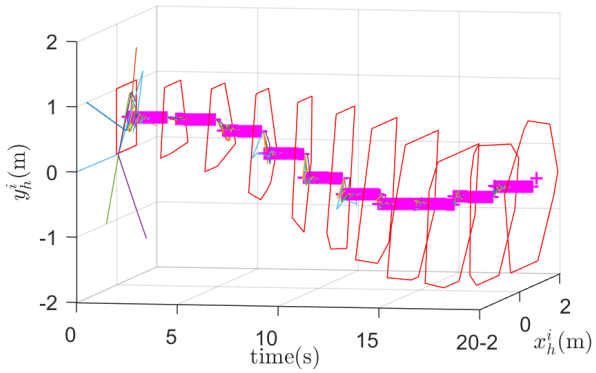

Figure 3.8 – The tracking performance of the robots while implementing the distributed containment observer (3.7) and the local control (3.14) with $\delta_c = 0.1$ seconds: the lines show the trajectory of (x_h^i, y_h^i) vs. time, while "+" show the location of $\bar{\mathbf{x}}_L(t_k)$ of the leaders. The red polygons indicate the convex hull formed by the moving leaders at each sampling time.

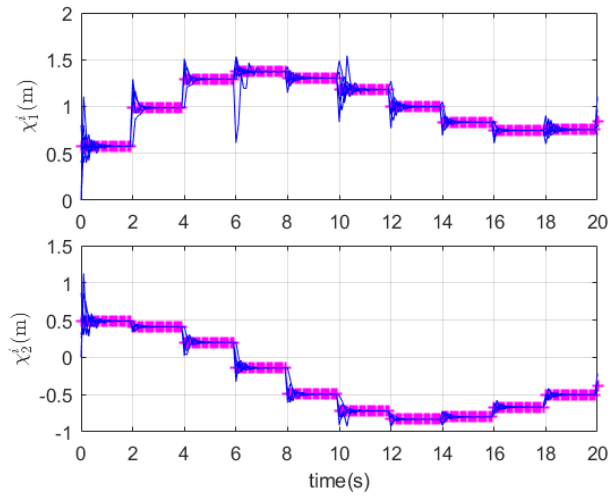

Figure 3.9 – Time history of the output of the containment observer (3.7) with $\delta_c = 0.1$ seconds: the blue curves show $\boldsymbol{\chi}^i$ and the markers "+" show the location of the coordinates of $\bar{\mathbf{x}}_L(t_k)$. The jumps in the location of $\bar{\mathbf{x}}_L(t_k)$ is due to the motion of the leaders and also the changes in the set of the observed leaders by each agent.

tecture for applying this containment control algorithm to unicycle robots is constructed by a two layer controller. The first layer used the distributed containment control algorithm as the observer to estimate the location of a point in the convex hull of the moving leaders. The second layer uses a the local finite-time controller to drive the unicycle robots to track their estimate by the containment observer in finite time. In this framework, the unicycle robots have continuous-time dynamics but they only communicate with each other in discrete-time fashion. Numerical results demonstrates the efficiency of our proposed algorithms.

Chapter 4

Dynamic active weighted average consensus and its application in containment control

4.1 Introduction

In this chapter, we consider a dynamic active weighted average consensus problem. In this problem, a group of agents interacting over a connected undirected graph should track the weighted average of their 'measured' local signals. However, at any time, only a subset of these agents are active, meaning that only a subset of agents collects measurements. The objective then in the active weighted average consensus problem is to obtain the weighted average of the collected measurements, which is the sum of the collected weighted measurements divided by the sum of the weights of the active agents. An example application is the containment problem shown in Fig. 4.1.

An intuitive solution to the active weighted average consensus problem may be to run two

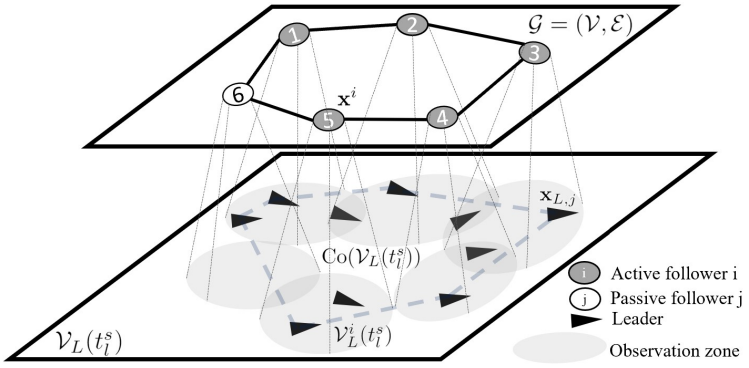

Figure 4.1 – A containment control scenario where a set of $\mathcal{V} = \{1, \cdots, 6\}$ followers should track the convex hull of a set of the leader agents that they observe: Here, followers $\mathcal{V}_{\mathrm{a}} = \{1, \cdots, 5\}$ are active agents that each observes a subset of the leaders, while follower 6 is the passive agent that should still follow the convex hull of the leaders despite having no measurement. The average of the geometric centers of the position of the leaders at each active agent is a point in the convex hull of the leaders (see Lemma 3.4.1). Thus, this containment problem can be formulated as an active weighted average consensus problem with the homogeneous weights of active agents.

conventional average consensus algorithms [1] in parallel. In one, the reference input of active agents is the weighted measurement and the reference input of the passive agents is set to 0. Then, in another, the number of the sum of the weights of active agents divided by the total number of the agents is obtained by using the weight as reference input for active agents and 0 for passive agents. Then, the active average weighted consensus solution is to divide the output of the first algorithm by the output of the second one. The previous chapter follows this approach to solve the containment control problem described in Fig. 4.1. However, besides extra computation and communication costs, this method is prone to zero-crossing for the output of its second average consensus algorithm leading to infinity tracking error so introducing the zero crossing protection measure (the small positive number of ϵ in Theorem 3.4.1) is necessary.

In this chapter, we propose a solution for dynamic active weighted average consensus over connected graphs that has the same computational and communication complexity of typical dynamic average consensus algorithms. In our proposed algorithm the agents can switch between active and passive modes or switch their piece-wise constant weights instantaneously,

as long as a dwell time exists between the switching incidences. Abrupt switching is usually the case for practical problems where agents are observing dynamic activities that can enter or leave the observation zone of the agents and thus change the agents' role from active to passive or vice versa in a non-smooth fashion. We model our algorithm as a switched linear system and study its convergence properties carefully by taking into account the piece-wise access of the agents to the reference signals. Our study employs the concept of distributional derivatives [92] to model the derivative of piece-wise continuous functions and characterize the transient error at the switching times. We also show that the bound of steady state tracking error is only proportional to the maximum rate of change of reference inputs. Then we derive a discrete-time implementation of our proposed dynamic active weighted average consensus algorithm and study its convergence. We demonstrate the application of our proposed algorithm in solving a containment problem. We show that the containment problem can be formulated and solved as an active average consensus problem.

4.2 Notations and Preliminaries

We let $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Z}$, $\mathbb{Z}_{>0}$ and $\mathbb{Z}_{\geq 0}$ denote the set of real, positive real, non-negative real, integer, positive integer, and non-negative integer, respectively. For $\mathbf{s} \in \mathbb{R}^d$, $\|\mathbf{s}\| = \sqrt{\mathbf{s}^\top \mathbf{s}}$ denotes the standard Euclidean norm. We let $\mathbf{1}_n$ (resp. $\mathbf{0}_n$) denote the vector of n ones (resp. n zeros), and $\mathbf{I}_n$ denote the $n \times n$ identity matrix. When clear from the context, we do not specify the matrix dimensions. $\mathrm{H}(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$ is the *Heaviside step* function. $\boldsymbol{\delta}(t) = \begin{cases} \infty, & t = 0 \\ 0, & t \neq 0 \end{cases}$ such that $\int_{-\infty}^{\infty} \boldsymbol{\delta}(t) \mathrm{d}t = 1$ is the *Dirac Delta* function. In a network of N agents, the aggregate vector of local variables $p^i \in \mathbb{R}$, $i \in \{1, \ldots, N\}$, is denoted by $\mathbf{p} = (p^1, \ldots, p^N)^\top \in \mathbb{R}^N$.

Consider the piece-wise continuous function

$$\mathbf{v}(t) = \begin{cases} \mathbf{v}_0(t), & t_0 \leq t < t_1, \\ \mathbf{v}_1(t), & t_1 \leq t < t_2, \\ \quad \vdots & \\ \mathbf{v}_{\bar{k}}(t), & t_{\bar{k}} \leq t \end{cases}, \tag{4.1}$$

where $\mathbf{v}_i \in \mathcal{C}^1, i \in \{1, \cdots, \bar{k}\}$. Using the Heaviside step function, (4.1) reads as $\mathbf{v}(t) = \mathbf{v}_0 + \sum_{k=1}^{\bar{k}} (\mathbf{v}_k - \mathbf{v}_{k-1})\, \mathrm{H}(t - t_k)$. Then, following [92], the distributional derivative of $\mathbf{v}(t)$ is

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v} = \dot{\tilde{\mathbf{v}}} + \sum\nolimits_{k=1}^{\bar{k}} (\mathbf{v}(t_k^+) - \mathbf{v}(t_k^-))\, \boldsymbol{\delta}(t - t_k), \tag{4.2}$$

where $\dot{\tilde{\mathbf{v}}} = \dot{\mathbf{v}}_0 + \sum_{k=1}^{\bar{k}} (\dot{\mathbf{v}}_k - \dot{\mathbf{v}}_{k-1})\, \mathrm{H}(t - t_k)$ or equivalently

$$\dot{\tilde{\mathbf{v}}} = \begin{cases} \dot{\mathbf{v}}_0(t), & t_0 \leq t < t_1, \\ \dot{\mathbf{v}}_1(t), & t_1 \leq t < t_2, \\ \quad \vdots & \\ \dot{\mathbf{v}}_{\bar{k}}(t), & t_{\bar{k}} \leq t. \end{cases}$$

We assume that the piece-wise continuous signals are right-continuous, i.e. $\mathbf{v}(t_k) = \mathbf{v}(t_k^+)$. Hereafter, we use the notation '$\dot{\tilde{}}$' to represent $\dot{\tilde{\mathbf{v}}}(t) = \begin{cases} \dot{\mathbf{v}}(t) & t \neq t_k \\ \dot{\mathbf{v}}(t_k^+) & t = t_k \end{cases}$.

An undirected *graph* is a triplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{1, \ldots, N\}$ is the *node set* and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the *edge set*, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a *adjacency* matrix such that $\mathsf{a}_{\mathsf{ij}} = \mathsf{a}_{\mathsf{ji}} > 0$ if $(i, j) \in \mathcal{E}$ and $\mathsf{a}_{ij} = 0$, otherwise. An edge (i, j) from i to j means that agents i and j can communicate. A *connected graph* is an undirected graph in which for every pair of nodes there is a path connecting them. The *degree* of a node i is $\mathsf{d}^i = \Sigma_{j=1}^N \mathsf{a}_{ij}$. The *Laplacian* matrix is $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \mathrm{Diag}(\mathsf{d}^1, \cdots, \mathsf{d}^N) \in \mathbb{R}^{N \times N}$. For connected graphs, $\mathbf{L}\mathbf{1}_N = \mathbf{0}$ and $\mathbf{1}_N^T \mathbf{L} = \mathbf{0}$. Moreover, $\mathbf{L}$ has one eigenvalue $\lambda_1 = 0$, and the rest of the eigenvalues $\{\lambda_i\}_{i=2}^N$ are positive. $\mathbf{T} = [\mathfrak{r} \quad \mathfrak{R}] \in \mathbb{R}^{N \times N}$ is an orthonormal matrix, where $\mathfrak{r} = \frac{1}{\sqrt{N}} \mathbf{1}_N$ and $\mathfrak{R} \in \mathbb{R}^{N \times (N-1)}$ is any matrix that makes $\mathbf{T}^\top \mathbf{T} = \mathbf{T}\mathbf{T}^\top = \mathbf{I}$. For a connected graph, $\mathbf{T}^\top \mathbf{L} \mathbf{T} = \left[\begin{smallmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathsf{L}^+ \end{smallmatrix}\right]$, where $\mathsf{L}^+ = \mathfrak{R}^\top \mathbf{L} \mathfrak{R}$, is a positive definite matrix with eigenvalues $\{\lambda_i\}_{i=2}^N \subset \mathbb{R}_{>0}$.

Lemma 4.2.1. *Suppose the nonzero matrix $\mathbf{E} \in \mathbb{R}^{N\times N}$ is a diagonal matrix whose diagonal elements are either 0 or of positive real numbers, and L is the Laplacian matrix of a connected graph. Then, $-(\mathbf{E}+\mathsf{L})$ is Hurwitz.*

Proof. Consider the system $\dot{\mathbf{x}} = -(\mathbf{E}+\mathsf{L})\mathbf{x}$. Now consider Lyapnov function $V = \frac{1}{2}\mathbf{x}^\top\mathbf{x}$. Then, $\dot{V} = -\mathbf{x}^\top\mathbf{E}\mathbf{x} - \mathbf{x}^\top\mathsf{L}\mathbf{x} \leq 0$, because $-\mathbf{x}^\top\mathbf{E}\mathbf{x} \leq 0$ and $-\mathbf{x}^\top\mathsf{L}\mathbf{x} \leq 0$. However, $\dot{V} \equiv 0$ happens when $-\mathbf{x}^\top\mathbf{E}\mathbf{x} = 0$ and $-\mathbf{x}^\top\mathsf{L}\mathbf{x} = 0$. But, since $-\mathbf{x}^\top\mathsf{L}\mathbf{x} = 0$ if and only if $\mathbf{x} = \alpha\mathbf{1}$, $\alpha \in \mathbb{R}$ then $\dot{V} \equiv 0$ if $\mathbf{x} \equiv \mathbf{0}$. Therefore, invoking [93, Theorem 4.11], we conclude that the system $\dot{\mathbf{x}} = -(\mathbf{E}+\mathsf{L})\mathbf{x}$ is uniformly exponentially stable. Thus, $-(\mathbf{E}+\mathsf{L})$ is Hurwitz. □

4.3 Problem Definition

Consider a network of N single integrator agents $\dot{x}^i = u^i$, $i \in \mathcal{V}$, interacting over a connected undirected graph $\mathcal{G}$. Suppose each agent $i \in \mathcal{V}$ has access to a measurable locally essentially bounded reference signal $\mathsf{r}^i : \mathbb{R}_{\geq 0} \to \mathbb{R}$ in a possibly intermittent fashion. For every agent $i \in \mathcal{V}$, we let $\eta^i(t)$ be the mode and weight indicator function for the agent $i \in \mathcal{V}$, which is in $\mathbb{R}_{>0}$ if agent i is active and has access to $\mathsf{r}^i(t)$ at time $t \in \mathbb{R}_{\geq 0}$, and 0 otherwise. Let $\mathcal{V}_{\mathrm{a}}(t) \subset \mathcal{V}$ be the set of active agents at time $t \in \mathbb{R}_{\geq 0}$, i.e., $\mathcal{V}_{\mathrm{a}}(t) = \big\{i \in \mathcal{V} \,\big|\, \eta^i(t) > 0\big\}$. In what follows, we assume that $\eta^i(t)$ and $|\mathcal{V}_{\mathrm{a}}(t)|$ are piece-wise constant functions of time, and $\mathcal{V}_{\mathrm{a}}(t) \neq \emptyset$ for all $t \in \mathbb{R}_{\geq 0}$. We refer to an agent in $\mathcal{V}\backslash\mathcal{V}_{\mathrm{a}}(t)$ as the passive agent at time t.

Problem 1 (Active weighted average consensus problem)**.** The active average consensus problem over $\mathcal{G}$ is defined as designing a distributed control input u^i such that the agreement state $x^i(t) \in \mathbb{R}$ of every agent $i \in \mathcal{V}$ tracks

$$\mathsf{avg}^{\mathrm{a}}(t) = \frac{\sum_{i=1}^N \eta^i(t)\,\mathsf{r}^i(t)}{\sum_{i=1}^N \eta^i(t)}. \qquad \square$$

In what follows, we first propose a distributed continuous-time algorithm to solve Problem 1. Then, we present a discrete-time implementation of this active weighted average consensus algorithm in which the agents sample the reference inputs with a rate of $1/\delta_s$ in a zero-order fashion. Lastly, we show how a containment problem can be cast as dynamic active (homogeneously weighted) average consensus problem and solved using our proposed algorithm.

4.4 Continuous-Time Dynamic Active Average Consensus

Our solution to solve Problem 1 over a connected undirected graph $\mathcal{G}$ is

$$\begin{aligned}\dot{x}^i(t) = &-\eta^i(t)(x^i(t)-\mathsf{r}^i(t)) - \sum\nolimits_{j=1}^{N} \mathsf{a}_{ij}(x^i(t)-x^j(t)) \\ &- \sum\nolimits_{j=1}^{N} \mathsf{a}_{ij}(v^i(t)-v^j(t)) + \eta^i(t)\dot{\mathsf{r}}^i(t), \qquad (4.3a)\\ \dot{v}^i(t) = &\sum\nolimits_{j=1}^{N} \mathsf{a}_{ij}(x^i(t)-x^j(t)), \qquad (4.3b)\end{aligned}$$

with $x^i(0), v^i(0) \in \mathbb{R}$, $i \in \mathcal{V}$. Here, $v^i(t) \in \mathbb{R}$ is an internal state that acts as an integral action. Next, we study the convergence properties of (4.3) by modeling it as a switched system and analyzing the collective response of the agents. In what follows, we let $\mathbf{E}(t) = \text{Diag}(\eta^1(t), \cdots, \eta^N(t))$. $\mathbf{E}(t)$ can be considered as switching in the class of non-zero diagonal matrices $\{\mathbf{E}_p\}_{p\in\mathcal{P}}$, $\mathcal{P}$ is the index set, each of which has diagonal elements being either positive real or 0. That is $\mathbf{E}(t) = \mathbf{E}_{\sigma(t)} \neq \mathbf{0}$ with the switching signal $\sigma(t) : \mathbb{R}_{\geq 0} \to \mathcal{P}$. We let $N_\sigma(0,t)$ denote the number of switchings of $\sigma(t)$ on the interval $[0,t)$. In our problem of interest, the following common assumption for switch linear systems holds [94, 95].

Assumption 1. There exist some $N_0 \in \mathbb{Z}_{\geq 0}$ and $\tau_{\text{D}} \in \mathbb{R}_{>0}$ such that, $N_\sigma(0,t) \leq N_0 + \frac{t}{\tau_{\text{D}}}$, $t \in \mathbb{R}_{>0}$, where τ_{D} is called the average dwell time and N_0 is the chatter bound. □

We let $\mathbf{avg}^{\mathrm{a}} = \mathsf{avg}^{\mathrm{a}}\mathbf{1}$, $\Delta\mathbf{avg}^{\mathrm{a}}_k = \mathbf{avg}^{\mathrm{a}}(t_k^+) - \mathbf{avg}^{\mathrm{a}}(t_k^-)$, $\mathbf{w}(t) = \mathbf{E}_{\sigma(t)}(\mathbf{r}(t) - \mathbf{avg}^{\mathrm{a}}(t))$, and $\Delta\mathbf{w}_k = \mathbf{w}(t_k^+) - \mathbf{w}(t_k^-)$, where t_k, $k \in \mathbb{Z}_{\geq 0}$ is the kth switching time of the switching signal $\sigma(t)$. Throughout this paper we assume $t_0 = 0$. Lastly, given a time $t \in \mathbb{R}_{\geq 0}$, $\bar{k} \in \mathbb{Z}_{\geq 0}$ is the largest integer such that $t_{\bar{k}} \leq t$.

For convenience in the correctness analysis of algorithm (4.3), we use the change of variables $\overline{\mathbf{e}} = \mathbf{T}^\top(\mathbf{x} - \mathbf{avg}^{\mathrm{a}})$, $\mathbf{q} = [q_1 \;\; \mathbf{q}_{2:N}^\top]^\top = \mathbf{T}^\top(\mathbf{L}\mathbf{v} - \mathbf{w})$ to write the equivalent compact form of (4.3) as

$$\dot{q}_1 = 0, \tag{4.4a}$$

$$\begin{bmatrix} \dot{\overline{\mathbf{e}}} \\ \dot{\mathbf{q}}_{2:N} \end{bmatrix} = \overline{\mathbf{A}}_{\sigma(t)} \begin{bmatrix} \overline{\mathbf{e}} \\ \mathbf{q}_{2:N} \end{bmatrix} + \overline{\mathbf{B}} \begin{bmatrix} \mathbf{E}\,\dot{\mathbf{r}} - \mathbf{a}\dot{\tilde{\mathbf{v}}}\mathbf{g}^{\mathrm{a}} \\ -\dot{\tilde{\mathbf{w}}} \end{bmatrix} - \overline{\mathbf{B}} \sum\nolimits_{k=1}^{\bar{k}} \begin{bmatrix} \Delta\mathbf{avg}^{\mathrm{a}}_k \\ \Delta\mathbf{w}_k \end{bmatrix} \boldsymbol{\delta}(t - t_k). \tag{4.4b}$$

where $\overline{\mathbf{A}}_{\sigma(t)} = \begin{bmatrix} -\mathbf{T}^\top(\mathbf{E}_{\sigma(t)}+\mathbf{L})\mathbf{T} & -\begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{N-1} \end{bmatrix} \\ \begin{bmatrix} \mathbf{0} & \mathbf{L}^+\mathbf{L}^+ \end{bmatrix} & \mathbf{0} \end{bmatrix}$ and $\overline{\mathbf{B}} = \begin{bmatrix} \mathbf{T}^\top & \mathbf{0} \\ \mathbf{0} & \mathfrak{R}^\top \end{bmatrix}$. Here, we used the facts that $\mathfrak{r}^\top\dot{\tilde{\mathbf{w}}} = 0$ and $\mathfrak{r}^\top\Delta\mathbf{w}_k = 0$. Also, we used $\mathfrak{R}\mathfrak{R}^\top\mathbf{L} = \mathbf{L}$ to write $\mathfrak{R}^\top\mathbf{L}\mathbf{L}\mathfrak{R} = \mathbf{L}^+\mathbf{L}^+$. Lastly, note that since $\mathbf{avg}^{\mathrm{a}}$ and $\mathbf{w}$ are piece-wise continuous functions, we used (4.2) to compute their derivatives that appear in $\dot{\overline{\mathbf{e}}}$ and $\dot{\mathbf{q}}$. Using standard results for linear time-varying systems we can write

$$\begin{bmatrix} \overline{\mathbf{e}}(t) \\ \mathbf{q}_{2:N}(t) \end{bmatrix} = \Phi(t,0) \begin{bmatrix} \overline{\mathbf{e}}(0) \\ \mathbf{q}_{2:N}(0) \end{bmatrix} + \int_0^t \Phi(t,\tau)\overline{\mathbf{B}} \Big(\begin{bmatrix} \mathbf{E}\,\dot{\mathbf{r}} - \mathbf{a}\dot{\tilde{\mathbf{v}}}\mathbf{g}^{\mathrm{a}} \\ -\dot{\tilde{\mathbf{w}}} \end{bmatrix} - \sum\nolimits_{k=1}^{\bar{k}} \begin{bmatrix} \Delta\mathbf{avg}^{\mathrm{a}}_k \\ \Delta\mathbf{w}_k \end{bmatrix} \boldsymbol{\delta}(\tau - t_k) \Big) \mathrm{d}\tau, \tag{4.5}$$

where $\Phi(t,\tau)$ is the transition matrix of linear system (4.4b). The next result shows that the internal dynamics of (4.4b) is uniformly exponentially stable. Therefore, there always exists κ_s, λ_s such that

$$\|\Phi(t,\tau)\| \leq \kappa_s \mathrm{e}^{-\lambda_s(t-\tau)}, \quad t \geq \tau \geq 0. \tag{4.6}$$

Lemma 4.4.1. *Let $\mathcal{G}$ be a connected undirected graph. Then, every subsystem matrix $\overline{\mathbf{A}}_p$, $p \in \mathcal{P}$ of* (4.4b) *is Hurwitz. Furthermore, under Assumption 1 the internal dynamics of* (4.4b) *is uniformly exponentially stable, i.e.,* (4.6) *holds.*

Proof. Consider the radially unbounded quadratic Lyapunov function $V = \frac{1}{2}\mathbf{q}_{2:N}^{\top}(\mathbf{L}^{+}\mathbf{L}^{+})^{-1}\mathbf{q}_{2:N} + \frac{1}{2}\overline{\mathbf{e}}^{\top}\overline{\mathbf{e}}$ (a common Lyapunov function for all the subsystems $\overline{\mathbf{A}}_{p\in\mathcal{P}}$ of the switched system $\overline{\mathbf{A}}_{\sigma(t)}$). Here, note that since $\mathbf{L}^{+} > 0$, then $\mathbf{L}^{+}\mathbf{L}^{+} > 0$. The Lie derivative of V along the trajectories of internal dynamics of (4.4b) is

$$\dot{V} = -\,\overline{\mathbf{e}}^{\top}\mathbf{T}^{\top}(\mathbf{E}_p + \mathbf{L})\mathbf{T}\,\overline{\mathbf{e}} \leq 0, \quad p \in \mathcal{P}. \tag{4.7}$$

To establish negative semi-definiteness of $\dot{V}$, we invoke Lemma 4.2.1. So far we have established that V is a weak Lyapunov function. Next, we use the LaSalle invariant principle and [96, Theorem 4] to establish exponential stability of the internal dynamics of (4.4b). Let $\mathcal{S}_p = \{(\overline{\mathbf{e}}, \mathbf{q}_{2:N}) \in \mathbb{R}^N \times \mathbb{R}^{N-1} | \dot{V} \equiv 0\}$ for all $p \in \mathcal{P}$. Given (4.7), we then have $\mathcal{S}_p = \{(\overline{\mathbf{e}}, \mathbf{q}_{2:N}) \in \mathbb{R}^N \times \mathbb{R}^{N-1} | \overline{\mathbf{e}} = 0\}$, for all $p \in \mathcal{P}$. Then, it is straightforward to observe that the trajectories of the internal dynamics of (4.4b) that belong to $\mathcal{S}_{p\in\mathcal{P}}$, should also satisfy $\mathbf{q}_{2:N} \equiv \mathbf{0}$. Therefore, the largest invariant set of the internal dynamics of (4.4b) in $\mathcal{S}_{p\in\mathcal{P}}$ is the origin. Thus, using [93, Theorem 4.4] all the subsystems $\overline{\mathbf{A}}_{p\in\mathcal{P}}$ of the switched system $\overline{\mathbf{A}}_{\sigma(t)}$ are globally asymptotically stable. Moreover, because the all subsystems of the switched system share the common weak quadratic Lyapunov function and the largest invariant set of $\mathcal{S}_{p\in\mathcal{P}}$ contains only the origin, given Assumption 1, by virtue of [96, Theorem 4] the internal dynamics of (4.4b), which is a switched system, is uniformly exponentially stable. Here, we note that according to [97, Theorem 2.1] the origin being the largest invariant set of $\mathcal{S}_p$, for all $p \in \mathcal{P}$, ensures that the observability condition in [96, Theorem 4] is satisfied. □

Given (4.5) and (4.6), we can characterize the tracking performance of active average con-

sensus algorithm (4.3) as follows.

Theorem 4.4.1. *Let $\mathcal{G}$ be a connected undirected graph and suppose Assumption 1 holds. Then, starting from any $x^i(0), v^i(0) \in \mathbb{R}$, $i \in \mathcal{V}$ the trajectories of dynamic active average consensus algorithm* (4.3) *satisfy*

$$\begin{aligned}|x^i(t) - \mathsf{avg}^{\mathrm{a}}(t)| \leq \kappa_s \, \mathrm{e}^{-\lambda_s t} \left\| \begin{bmatrix} \mathbf{x}(0) - \mathbf{avg}^{\mathrm{a}}(0) \\ \mathbf{L}\mathbf{v}(0) - \mathbf{w}(0) \end{bmatrix} \right\| \\ + \kappa_s \sum\nolimits_{k=1}^{\bar{k}} \mathrm{e}^{-\lambda_s (t-t_k)} \left\| \begin{bmatrix} \Delta \mathbf{avg}_k^{\mathrm{a}} \\ \Delta \mathbf{w}_k \end{bmatrix} \right\| \mathrm{H}(t - t_k) \\ + \frac{\kappa_s}{\lambda_s} \sup_{0 \leq \tau \leq t} \left\| \begin{bmatrix} \mathbf{E}_{\sigma(\tau)} \, \dot{\mathbf{r}}(\tau) - \dot{\mathbf{avg}}^{\mathrm{a}}(\tau) \\ -\dot{\tilde{\mathbf{w}}}(\tau) \end{bmatrix} \right\|. \end{aligned} \tag{4.8}$$

Proof. We note that $\|\overline{\mathbf{B}}\| \leq 1$. Then, given (4.5) and (4.6), we can write

$$\begin{aligned}\left\| \begin{bmatrix} \overline{\mathbf{e}}(t) \\ \mathbf{q}_{2:N}(t) \end{bmatrix} \right\| \leq \kappa_s \, \mathrm{e}^{-\lambda_s t} \left\| \begin{bmatrix} \overline{\mathbf{e}}(0) \\ \mathbf{q}_{2:N}(0) \end{bmatrix} \right\| + \kappa_s \int_0^t \mathrm{e}^{-\lambda_s (t-\tau)} \left\| \begin{bmatrix} \mathbf{E} \, \dot{\mathbf{r}} - \dot{\mathbf{avg}}^{\mathrm{a}} \\ -\dot{\tilde{\mathbf{w}}} \end{bmatrix} \right\| \mathrm{d}\tau \\ + \kappa_s \sum\nolimits_{k=1}^{\bar{k}} \int_0^t \mathrm{e}^{-\lambda_s (t-\tau)} \left\| \begin{bmatrix} \Delta \mathbf{avg}_k^{\mathrm{a}} \\ \Delta \mathbf{w}_k \end{bmatrix} \boldsymbol{\delta}(\tau - t_k) \right\| \mathrm{d}\tau. \end{aligned}$$

Then, the Hölder inequality is used to bound the second term of the right hand side to arrive at

$$\begin{aligned}\left\| \begin{bmatrix} \overline{\mathbf{e}}(t) \\ \mathbf{q}_{2:N}(t) \end{bmatrix} \right\| \leq \kappa_s \, \mathrm{e}^{-\lambda_s t} \left\| \begin{bmatrix} \overline{\mathbf{e}}(0) \\ \mathbf{q}_{2:N}(0) \end{bmatrix} \right\| + \frac{\kappa_s}{\lambda_s} \sup_{0 \leq \tau \leq t} \left\| \begin{bmatrix} \mathbf{E}_{\sigma(\tau)} \, \dot{\mathbf{r}}(\tau) - \dot{\mathbf{avg}}^{\mathrm{a}}(\tau) \\ -\dot{\tilde{\mathbf{w}}}(\tau) \end{bmatrix} \right\| \\ + \kappa_s \sum\nolimits_{k=1}^{\bar{k}} \int_0^t \mathrm{e}^{-\lambda_s (t-\tau)} \left\| \begin{bmatrix} \Delta \mathbf{avg}_k^{\mathrm{a}} \\ \Delta \mathbf{w}_k \end{bmatrix} \boldsymbol{\delta}(\tau - t_k) \right\| \mathrm{d}\tau. \end{aligned}$$

Consequently, with integration by parts, the last term is equivalent to $\kappa_s \sum_{k=1}^{\bar{k}} \mathrm{e}^{-\lambda_s (t-t_k)}$

$\left\| \begin{bmatrix} \Delta \mathsf{avg}_k^{\mathrm{a}} \\ \Delta \mathbf{w}_k \end{bmatrix} \right\| \mathrm{H}(t-t_k)$. Then, since $\mathbf{T}$ is an orthonormal matrix, we have $\left\| \begin{bmatrix} \overline{\mathbf{e}}(0) \\ \mathbf{q}_{2:N}(0) \end{bmatrix} \right\| = \left\| \begin{bmatrix} \mathbf{x}(0) - \mathsf{avg}^{\mathrm{a}}(0) \\ \mathbf{L}\mathbf{v}(0) - \mathbf{w}(0) \end{bmatrix} \right\|$ and $\|\mathbf{x} - \mathsf{avg}^{\mathrm{a}}\| = \|\overline{\mathbf{e}}\|$. Finally, (4.8) is derived along with the relation $|x^i - \mathsf{avg}^{\mathrm{a}}| \leq \left\| \begin{bmatrix} \overline{\mathbf{e}}^\top & \mathbf{q}_{2:N}^\top \end{bmatrix}^\top \right\|$. □

We note that the first summand of the tracking error bound (4.8) is the transient response, which vanishes over time. The second summand is due to the agents alternating between active and passive sets or active agents switching their weights. If the average dwell time τ_D is large, this error also disappears after a while. The third summand can result in a steady-state error. This error that is expected in dynamic average consensus algorithms, as tracking an arbitrarily fast average signal with zero error is not feasible unless agents have some priori information about the dynamics generating the signals [1]. However, the size of this error is proportional to the rate of change of the signals and can be limited by limiting the rate. We recall that to provide bounded tracking, previous work in [59–61] require both the reference input signals and their rate of change to be bounded. If the local reference signals are static and the agents do not switch, the agents exponentially converge to $\mathsf{avg}^{\mathrm{a}}$ without steady-state error. Lastly, algorithm (4.3) does not require specific initialization. In other words, the convergence property of algorithm (4.3) uniformly holds for any initialization. Therefore, as long as the graph stays connected, agents can leave and join the network without effecting the convergence guarantees. Figure 4.2 demonstrates the performance of algorithm (4.3) in a numerical example.

4.5 Discrete-Time Dynamic Active Average Consensus

We consider a scenario where active agents sample their reference inputs at sampling times $t_l^s = l\delta_s \in \mathbb{R}_{\geq 0}$, $l \in \mathbb{Z}_{\geq 0}$, $\delta_s \in \mathbb{R}_{>0}$. The agents can communicate at discrete-times $t_k^c = k\delta_c \in \mathbb{R}_{\geq 0}$, $k \in \mathbb{Z}_{\geq 0}$, $\delta_c \in \mathbb{R}_{>0}$. The objective of every agent $i \in \mathcal{V}$ is to track $\mathsf{avg}^{\mathrm{a}}(k)$ (where k is the shorthand for t_k^c). To solve the active average consensus problem under this scenario,

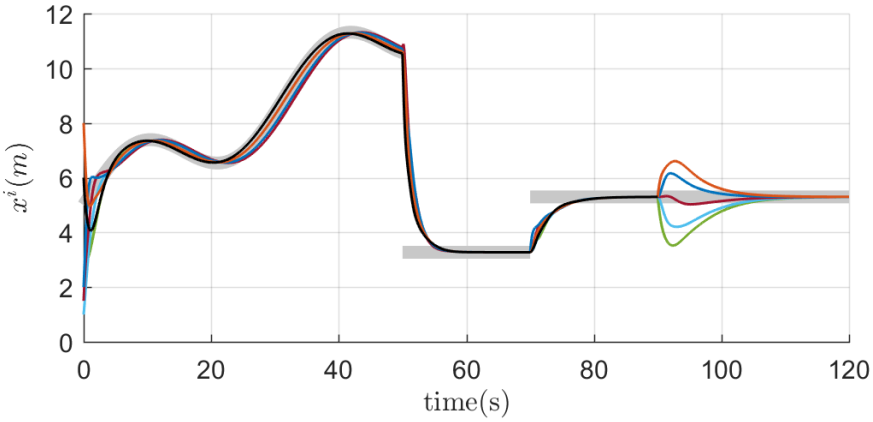

Figure 4.2 – A network of 6 agents with a ring interaction topology executes the active average consensus algorithm (4.3). In time interval $t \in [0, 50)$, the observing agents $\mathcal{V}_{\mathrm{a}}(t) = \{1, 2, 4, 6\}$ all have dynamic inputs. The observing agents at $t \in [50, 70)$ and $t \in [70, 120]$ are, respectively, $\mathcal{V}_{\mathrm{a}}(t) = \{2, 3, 5, 6\}$ $\mathcal{V}_{\mathrm{a}}(t) = \{3, 6\}$ and their observations are static signals. Agent 1 (black line) leaves the network at $t = 90$. The gray thick line represents $\mathsf{avg}^{\mathrm{a}}(t)$. The agents can track the dynamic $\mathsf{avg}^{\mathrm{a}}(t)$ with bounded error in $t \in [0, 50)$, while their tracking error is close to zero for the rest of the time as the reference signals are constant after $t = 50$. The transient tracking error at time $t = 70$ is due to switching of some of agents to the passive mode. This error is captured by the second term in the right-hand side of (4.8). Lastly, agent 1's leaving causes perturbations at $t = 90$ but the network still converge to $\mathsf{avg}^{\mathrm{a}}(t)$.

we propose that every agent $i \in \mathcal{V}$ implements

$$x^i(k) = z^i(k) + \eta^i(k)\mathsf{r}^i(k), \tag{4.9a}$$

$$z^i(k+1) = z^i(k) - \delta_c \eta^i(k)(x^i(k) - \mathsf{r}^i(k)) \tag{4.9b}$$
$$- \delta_c \sum_{j=1}^{N} \mathsf{a}_{ij}(x^i(k) - x^j(k)) - \delta_c \sum_{j=1}^{N} \mathsf{a}_{ij}(v^i(k) - v^j(k)),$$

$$v^i(k+1) = v^i(k) + \delta_c \sum\nolimits_{j=1}^{N} \mathsf{a}_{ij}(x^i(k) - x^j(k)). \tag{4.9c}$$

which is an Euler discretized implementation of the active average algorithm (4.3) with stepszie δ_c. Here, we assume that if $\delta_s \neq \delta_c$, the agents perform a zero-order hold sampling, so that $\mathsf{r}^i(k) = \mathsf{r}^i(\bar{l})$, $i \in \mathcal{V}$, where $\bar{l}$ is the latest sampling time step such that $t_{\bar{l}}^s \leq t_k^c$. We let $\sigma(k) : \mathbb{Z}_{\geq 0} \to \mathcal{P}$ be the switching signal of $\mathbf{E}(k)$, i.e., $\mathbf{E}(k) = \mathbf{E}_{\sigma(k)}$. Then, we implement the same change of variable as for the continuous-time algorithm (4.3) to write the compact

form of (4.9) as

$$q_1(k+1) = q_1(k), \tag{4.10a}$$

$$\begin{bmatrix} \overline{\mathbf{e}}(k+1) \\ \mathbf{q}_{2:N}(k+1) \end{bmatrix} = (\mathbf{I} + \delta_c \overline{\mathbf{A}}_\sigma) \begin{bmatrix} \overline{\mathbf{e}}(k) \\ \mathbf{q}_{2:N}(k) \end{bmatrix} + \overline{\mathbf{B}} \begin{bmatrix} \Delta \mathbf{Er}(k) - \Delta \mathbf{avg}^{\mathrm{a}}(k) \\ -\Delta \mathbf{w}(k) \end{bmatrix}, \tag{4.10b}$$

where $\overline{\mathbf{A}}_\sigma$ and $\overline{\mathbf{B}}$ are defined in (4.4b), $\Delta \mathbf{Er}(k) = \Delta \mathbf{E}(k+1)\mathbf{r}(k+1) - \Delta \mathbf{E}(k)\mathbf{r}(k)$, $\Delta \mathbf{avg}^{\mathrm{a}}(k) = \mathbf{avg}^{\mathrm{a}}(k+1) - \mathbf{avg}^{\mathrm{a}}(k)$, and $\Delta \mathbf{w}(k) = \mathbf{w}(k+1) - \mathbf{w}(k)$. Then, given $|x^i - \mathsf{avg}^{\mathrm{a}}| \leq \left\| \begin{bmatrix} \overline{\mathbf{e}}^\top & \mathbf{q}_{2:N}^\top \end{bmatrix}^\top \right\|$, the tracking performance of (4.9) can be understood by studying the convergence properties of (4.10b). For the discrete-time implementation, the following assumption holds.

Assumption 2. *The switched system* (4.10b) *switches in a finite set of subsystem, i.e.,* $\sigma(k) : \mathbb{Z}_{\geq 0} \to \overline{\mathcal{P}}$*, where* $\overline{\mathcal{P}} \subset \mathcal{P}$ *is a finite subset.*

The first result below shows that with a proper choice for δ_c every subsystem $(\mathbf{I} + \delta_c \overline{\mathbf{A}}_p)$, $p \in \mathcal{P}$ is Schur. However, this is not enough to guarantee that the internal dynamics of (4.10b) is exponentially stable. To provide such guarantee, following [98, Corollary 1], we impose the following standard assumption.

Assumption 3. The average dwell time τ_D of the switching signal $\sigma(k)$ satisfies $\tau_D \geq \tau_D^*$, where τ_D^* is a stable average dwell time of the switched system (4.10b). □

Note that τ_D^* of the switched system (4.10b) can be computed using the methods introduced in [98, 99].

Lemma 4.5.1. *Let* $\mathcal{G}$ *be a connected undirected graph. Then, every subsystem matrix* $(\mathbf{I} + \delta_c \overline{\mathbf{A}}_p)$*,* $p \in \overline{\mathcal{P}}$ *of* (4.10b) *is Schur provided* $\delta_c \in (0, \bar{d})$*, where*

$$\bar{d} = \min \left\{ \{ -2 \frac{Re(\mu_{i,p})}{|\mu_{i,p}|^2} \}_{i=1}^{2N-1} \right\}_{p \in \overline{\mathcal{P}}}$$

and $\{\mu_{i,p}\}_{i=1}^{2N-1}$ *are the set of eigenvalues of* $\overline{\mathbf{A}}_p$*. Furthermore, under Assumption 3 the internal dynamics of* (4.10b) *is uniformly exponentially stable, i.e., there always exists* $\kappa_d \in \mathbb{R}_{>0}$ *and* $\omega_d \in (0,1)$*, such that, the state transition matrix* $\Phi(k,j)$ *of* (4.10b) *satisfies*

$$\|\Phi(k,j)\| \leq \kappa_d\, \omega_d^{(k-j)}, \quad k \geq j \geq 0, k, j \in \mathbb{Z}_{\geq 0}. \tag{4.11}$$

Proof. Lemma 4.4.1 ensures that every $\overline{\mathbf{A}}_p$, $p \in \overline{\mathcal{P}} \subset \mathcal{P}$ is a Hurwitz matrix. Then, it follows from [1, Lemma S1] that $(\mathbf{I} + \delta_{c,p}\overline{\mathbf{A}}_p)$, $p \in \overline{\mathcal{P}}$ is Schur if $\delta_{c,p} \in (0, \bar{d}_p)$, where $\bar{d}_p = \min\{-2\frac{\mathrm{Re}(\mu_{i,p})}{|\mu_{i,p}|^2}\}_{i=1}^{2N-1}$. As a result, $(\mathbf{I} + \delta_c\overline{\mathbf{A}}_p)$, $p \in \overline{\mathcal{P}}$ is Schur if $\delta_c \in (0, \bar{d})$, where $\bar{d} = \min\{\bar{d}_p\}_{p\in\overline{\mathcal{P}}}$. Then, given Assumption 3, it follows from [98, Corollary 1] that the zero input dynamics of switched system (4.10b) is uniformly exponentially stable. □

The next result characterizes the tracking performance of (4.9).

Theorem 4.5.1. *Let* $\mathcal{G}$ *be a connected undirected graph and suppose Assumption 2 and 3 hold.. Then, for any* $\delta_c \in (0, \bar{d})$*, starting from any* $x^i(0), v^i(0) \in \mathbb{R}$*,* $i \in \mathcal{V}$*, the trajectories of dynamic active average consensus algorithm* (4.9) *satisfy*

$$\begin{aligned}|x^i(k) - \mathsf{avg}^{\mathrm{a}}(k)| \leq \kappa_d \omega_d^k \left\| \begin{bmatrix} \mathbf{x}(0) - \mathbf{avg}^{\mathrm{a}}(0) \\ \mathbf{L}\mathbf{v}(0) - \mathbf{w}(0) \end{bmatrix} \right\| \\ + \frac{\kappa_d(1-\omega_d^k)}{1-\omega_d} \sup_{0\leq l\leq k-1} \left\| \begin{bmatrix} \Delta\mathbf{E}\mathbf{r}(l) - \Delta\mathbf{avg}^{\mathrm{a}}(l) \\ -\Delta\mathbf{w}(l) \end{bmatrix} \right\|. \end{aligned} \tag{4.12}$$

Proof. Using standard results for linear systems, trajectories of (4.10b) are given by

$$\begin{bmatrix} \overline{\mathbf{e}}(k) \\ \mathbf{q}_{2:N}(k) \end{bmatrix} = \Phi(k,0) \begin{bmatrix} \overline{\mathbf{e}}(0) \\ \mathbf{q}_{2:N}(0) \end{bmatrix} + \sum\nolimits_{j=0}^{k-1} \Phi(k, j+1)\overline{\mathbf{B}} \begin{bmatrix} \Delta\mathbf{E}\mathbf{r}(j) - \Delta\mathsf{avg}^{\mathrm{a}}(j) \\ -\Delta\mathbf{w}(j) \end{bmatrix}.$$

Then, given that $\|\overline{\mathbf{B}}\| \leq 1$ and (4.11) we can write

$$\left\| \begin{bmatrix} \overline{\mathbf{e}}(k) \\ \mathbf{q}_{2:N}(k) \end{bmatrix} \right\| \leq \kappa_d \, \omega_d^k \left\| \begin{bmatrix} \overline{\mathbf{e}}(0) \\ \mathbf{q}_{2:N}(0) \end{bmatrix} \right\| + \kappa_d \sum\nolimits_{j=0}^{k-1} \omega_d^j \sup_{0 \leq l \leq k-1} \left\| \begin{bmatrix} \Delta \mathbf{Er}(l) - \Delta \mathbf{avg}^{\mathrm{a}}(l) \\ -\Delta \mathbf{w}(l) \end{bmatrix} \right\|.$$

By the sum of geometric sequence, $\kappa_d \sum_{j=0}^{k-1} \omega_d^j = \frac{\kappa_d(1-\omega_d^k)}{1-\omega_d}$. Then, given that $\left\| \begin{bmatrix} \overline{\mathbf{e}}(0) \\ \mathbf{q}_{2:N}(0) \end{bmatrix} \right\| = \left\| \begin{bmatrix} \mathbf{x}(0) - \mathbf{avg}^{\mathrm{a}}(0) \\ \mathbf{Lv}(0) - \mathbf{w}(0) \end{bmatrix} \right\|$ and $|x^i - \mathsf{avg}^{\mathrm{a}}| \leq \left\| \begin{bmatrix} \overline{\mathbf{e}}^\top & \mathbf{q}_{2:N}^\top \end{bmatrix}^\top \right\|$, tracking error (4.12) is established. □

4.6 Distributed containment control via dynamic active average consensus modeling

In this section, we use the discrete-time dynamic active average consensus algorithm to solve a containment control problem. Let's briefly recap the containment control setting. Consider a group of M (M can change with time) mobile leaders that are moving with a bounded velocity on a $\mathbb{R}^2$ or $\mathbb{R}^3$ space. $\mathbf{x}_{L,j}(t)$ represents the position vector of leader $j \in \{1, \ldots, M\}$ at time $t \in \mathbb{R}_{\geq 0}$. A set of networked follower agents $\mathcal{V} = \{1, \cdots, N\}$ interacting over a connected graph $\mathcal{G}$ monitors the leaders. The agents can communicate at discrete-times $t_k^c = k\delta_c \in \mathbb{R}_{\geq 0}$, $k \in \mathbb{Z}_{\geq 0}$, $\delta_c \in \mathbb{R}_{>0}$. The agents sample the leaders at sampling times $t_l^s = l\delta_s \in \mathbb{R}_{\geq 0}$, $l \in \mathbb{Z}_{\geq 0}$, $\delta_s \in \mathbb{R}_{>0}$. We let $\mathcal{V}_L^i(t_l^s)$ be the set of leaders observed by agent $i \in \mathcal{V}$ at sampling time t_l^s. Between each sampling time, agent $i \in \mathcal{V}$ uses $\mathbf{x}_{L,j}(t) = \mathbf{x}_{L,j}(t_l^s)$ and $\mathcal{V}_L^i(t) = \mathcal{V}_L^i(t_l^s)$, $t \in [t_l^s, t_{l+1}^s)$, $l \in \mathbb{Z}_{\geq 0}$, $j \in \mathcal{V}_L^i(t_l^s)$. At every sampling time $t_l^s \in \mathbb{R}_{\geq 0}$, we let $\mathcal{V}_L(t_l^s)$ be the set of the mobile leaders that are observed jointly by the agents $\mathcal{V}$, i.e., $\mathcal{V}_L(t_l^s) = \cup_{i=1}^N \mathcal{V}_L^i(t_l^s)$ (see Fig. 4.1). We let $\mathcal{V}_a(t_l^s) \subset \mathcal{V}$ be the set of the active agents that observe at least one leader at t_l^s, $k \in \mathbb{Z}_{\geq 0}$; we assume that $\mathcal{V}_{\mathrm{a}}(t_l^s) \neq \emptyset$. In what follows, the objective is to design a distributed control that enables each follower $i \in \mathcal{V}$ to derive its local state $\mathbf{x}^i$ to asymptotically track $\mathrm{Co}(\mathcal{V}_L(t_l^s))$, the convex

hull of the set of the location of the observed leaders $\mathcal{V}_L(t_l^s)$, with a bounded error $e \geq 0$. To simplify notation, we wrote $\mathrm{Co}(\{\mathbf{x}_{L,j}(t)\}_{j\in\mathcal{V}_L(t)})$ as $\mathrm{Co}(\mathcal{V}_L(t))$. We state the objective of the containment control as $\|\mathbf{x}^i(t_k^c) - \bar{\mathbf{x}}_L(t_k^c)\| \leq e$, $i \in \mathcal{V}$, where $\bar{\mathbf{x}}_L(t_k^c) \in \mathrm{Co}(\mathcal{V}_L(t_k^c))$. The agents have no knowledge about the motion model of the leaders. Since followers observe the dynamic leaders collaboratively, the tracking error e is expected as the measurement of each active follower needs time to propagate through the network to the rest of the followers.

Our solution builds on the key observation of Lemma 3.4.1. Let

$$\mathbf{r}^i(t_l^s) = \begin{cases} \frac{\sum_{j\in\mathcal{V}_L^i(t_l^s)} \mathbf{x}_{L,j}(t_l^s)}{|\mathcal{V}_L^i(t_l^s)|}, & i \in \mathcal{V}_{\mathrm{a}}(t_l^s), \\ \mathbf{0}, & i \in \mathcal{V}\backslash\mathcal{V}_{\mathrm{a}}(t_l^s). \end{cases} \tag{4.13}$$

Then, in light of Lemma 3.4.1, we know that

$$\bar{\mathbf{x}}_L(t_l^s) = \frac{\sum_{i\in\mathcal{V}_{\mathrm{a}}(t_l^s)} \mathbf{r}^i(t_l^s)}{|\mathcal{V}_{\mathrm{a}}(t_l^s)|} \in \mathrm{Co}(\mathcal{V}_L(t_l^s)). \tag{4.14}$$

Equation (4.14) states that $\bar{\mathbf{x}}_L(t_l^s)$, a point in the convex hull of the leaders, is the average of the input $\mathsf{r}^i(t_l^s)$ of the active agents $\mathcal{V}_{\mathrm{a}}(t_l^s)$. Thus, our containment problem can be solved by implementing the discrete-time active weighted average consensus algorithm (4.9) with homogeneous weights for the active agents as described in the result below.

Lemma 4.6.1. *Let the interaction topology $\mathcal{G}$ of the followers be a connected graph and suppose that the agents communicate at $t_k^c = k\delta_c \in \mathbb{R}_{\geq 0}$, $k \in \mathbb{Z}_{\geq 0}$. Assume that at each sampling time $t_l^s = l\delta_s \in \mathbb{R}_{\geq 0}$, $l \in \mathbb{Z}_{\geq 0}$, we have $\mathcal{V}_{\mathrm{a}}(t_l^s) \neq \emptyset$. Let the local reference inputs of the agents be* (4.13)*. Assume that the leader observation is done is a zero order fashion, i.e., $\mathbf{r}^i(t) = \mathbf{r}^i(t_l^s)$, $i \in \mathcal{V}$, and $\overline{\mathbf{x}}_L(t) = \overline{\mathbf{x}}_L(t_l^s)$ for $t \in [t_l^s, t_{l+1}^s)$, where $\overline{\mathbf{x}}_L(t_l^s)$ is given in* (4.14)*. Moreover, assume $\|\mathbf{x}_{L,j}(t_{l+1}^s) - \mathbf{x}_{L,j}(t_l^s)\|, j \in \{1, \cdots, M\}$, is bounded. Let $\eta^i(t) = 1$ if $i \in \mathcal{V}_{\mathrm{a}}(t_l^s)$, otherwise, $\eta^i(t) = 0$ for $t \in [t_l^s, t_{l+1}^s)$. Suppose that each agent $i \in \mathcal{V}$ implements the distributed algorithm* (4.9)*. Then, the tracking error $\|\mathbf{x}^i(t_k^c) - \overline{\mathbf{x}}_L(t_k^c)\|$ is bounded.*

Proof. It is clear that by letting the local reference inputs of agents be (4.13), the active average of the reference inputs is $\mathsf{avg}^{\mathrm{a}}(t_k^c) = \frac{\sum_{i\in\mathcal{V}_{\mathrm{a}}(t_k^c)} \mathbf{r}^i(t_k^c)}{|\mathcal{V}_{\mathrm{a}}(t_k^c)|} = \bar{\mathbf{x}}_L(t_k^c)$. Then, Theorem 4.5.1 guarantees $\|\mathbf{x}^i(t_k^c) - \overline{\mathbf{x}}_L(t_k^c)\|$ is bounded if each agents implements (4.9). □

Our solution in Lemma 4.6.1 applies to scenarios like in Fig. 4.1 where the observation sets of the followers have overlap. It is interesting to note that in case of overlapping observations, $\overline{\mathbf{x}}_L$ is not the centroid of the leaders. Next, note that by virtue of Theorem 4.5.1, if the leaders are static or move towards a static configuration, the algorithm convergences exactly to $\overline{\mathbf{x}}_L$. Otherwise, to ensure that the followers stay in the convex hull while tracking $\overline{\mathbf{x}}_L$ with some error, we may have to require that the convex hull of the leaders should be sufficiently large.

For demonstration, consider a case that 6 followers with a ring interaction graph aim to follow the convex hull of 10 leaders in a two dimensional space. The followers observe the leaders at 1 Hz according to the scenario described below where the set of active followers changes at $t_l^s = 5$ and $t_l^s = 10$ seconds:

- $0 \le t_l^s < 5$: $\mathcal{V}_L^1(t_l^s) = \{1,4,6,8\}$, $\mathcal{V}_L^2(t_l^s) = \{2,4,7,8,10\}$, $\mathcal{V}_L^3(t_l^s) = \{3,4,5,9\}$, $\mathcal{V}_L^4(t_l^s) = \emptyset$, $\mathcal{V}_L^5(t_l^s) = \{1,3,9\}$ and $\mathcal{V}_L^6(t_l^s) = \emptyset$,
- $5 \le t_l^s < 10$: $\mathcal{V}_L^1(t_l^s) = \{3,5,6,8\}$, $\mathcal{V}_L^2(t_l^s) = \{1,2,7,9,10\}$, $\mathcal{V}_L^3(t_l^s) = \{3,4,5,9\}$, $\mathcal{V}_L^4(t_l^s) = \emptyset$, $\mathcal{V}_L^5(t_l^s) = \{1,3,9\}$ and $\mathcal{V}_L^6(t_l^s) = \{2,5,7,9\}$,
- $10 \le t_l^s \le 20$: $\mathcal{V}_L^1(t_l^s) = \{1,2,5,8\}$, $\mathcal{V}_L^2(t_l^s) = \{2,3,6,7,10\}$, $\mathcal{V}_L^3(t_l^s) = \{3,4,5,9\}$, $\mathcal{V}_L^4(t_l^s) = \{3,10\}$, $\mathcal{V}_L^5(t_l^s) = \{1,3,9\}$ and $\mathcal{V}_L^6(t_l^s) = \{2,5,7,9\}$.

The communication frequency of the followers is 5 Hz. Figure 4.3 shows that the proposed distributed containment control of Lemma 4.6.1 results in a bounded tracking of the convex hull of the observed leaders. The interested reader can also find an application study of use of our solution in Lemma 4.6.1 in solving containment control for a group of unicycle

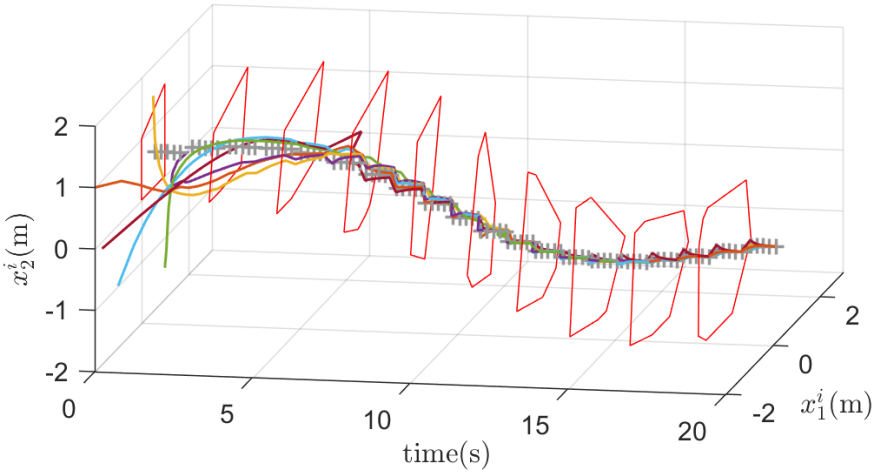

Figure 4.3 – The containment tracking performance of the follower agents while implementing the distributed algorithm (4.9): the solid curves show the trajectory of $\mathbf{x}^i$ vs. time, while "+" show the location of $\bar{\mathbf{x}}_L(t_k^c)$ of the leaders. The red polygons indicate the convex hull formed by the moving leaders.

followers with continuous-time dynamics in the previous chapter. There, the algorithm in Lemma 4.6.1 is used as an observer to generate the tracking points for the followers.

4.7 Conclusion

We proposed a dynamic active weighted average consensus algorithm that makes both active and passive agents track the weighted average of the collected reference inputs. The stability and tracking performance were analyzed in both continuous- and discrete-time implementations. We also showed that a containment control can be formulated as an active average consensus problem and solved using our proposed discrete-time algorithm.

Chapter 5

Distributed coverage control for mobile agents deployment

5.1 Introduction

In this chapter, we propose a novel deployment strategy for mobile agents to cover a collection of dense targets with their heterogeneous anisotropic services. An unknown density distribution of the dense targets is considered as the priority function in our deployment. Our deployment objective is deploying the agents such that the resulting QoS distribution of agents similar to the density distribution of the targets. Hence, the agents' service efficiently covers the targets; that is, the place containing more (fewer) targets is served with higher (lower) QoS.

We model the unknown density distribution of targets by a Gaussian mixture model (GMM). We propose a consensus-based distributed expectation-maximization (EM) algorithm to let the agents learn the parameters of GMM. With the proposed distributed EM, we do not require each agent has to measure the targets locally. Only a subset of the agents can

measure the targets and their information is then propagated through the consensus protocol to all agents. Moreover, the GMM intrinsically partition the area into a set of subregions, each of which represents a Gaussian basis; therefore, after estimating the target density distribution, the agents also finish the area partitioning task. We note that unlike the distributed Voronoi partition requiring the agents to be able to communicate to their Voronoi neighbors, which may be unrealistic because Voronoi neighbors are not definitely inside their communication range, our approach only requires the communication graph among the agents to be connected.

To result in the distributions of QoS and targets similar, we formulate a multi-agent assignment problem under the framework of optimal mass transport to allocate each agent to a subregion based on the similarity measure, Kullback-Leibler divergence (KLD), between the distribution of agent's QoS and the target's density distribution in the subregion. With the virtue of distributed simplex algorithm [100], the agents can solve the distributed assignment problem in a distributed manner and agree on the final assignment plan. Then, a local controller is applied to transport the agents with heterogeneous linear dynamics to their assigned subregions. We first investigate the case where the agents' QoS distributions are different Gaussian distributions. Since the footprint of Gaussian distribution is elliptic, the agents' QoS are heterogeneous and anisotropic. We theoretically show the optimal pose (position and orientation) of an agent that causes its QoS distribution most similar to the Gaussian basis of the subregion once it is assigned to a subregion. Then, we extend to the general case of non-Gaussian QoS. We seek a suboptimal pose for the agent by discretizing the pose space and numerically assesses the minimum KLD with each subregion. We illustrate two applications in the deployments of sensor network and UAV-aided wireless communication network.

5.2 Notations and Preliminaries

We let $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Z}$, $\mathbb{Z}_{>0}$ and $\mathbb{Z}_{\geq 0}$ denote the set of real, positive real, non-negative real, integer, positive integer, and non-negative integer, respectively. For $\mathbf{s} \in \mathbb{R}^d$, $\|\mathbf{s}\| = \sqrt{\mathbf{s}^\top \mathbf{s}}$ denotes the standard Euclidean norm. We let $\mathbf{1}_n$ (resp. $\mathbf{0}_n$) denote the vector of n ones (resp. n zeros), and $\mathbf{I}_n$ denote the $n \times n$ identity matrix. Given two continuous probability density distributions $p(\mathbf{x})$ and $q(\mathbf{x})$, $\mathbf{x} \in \mathbb{X}$, the *Kullback–Leibler divergence* (KLD) is defined as

$$\mathrm{D}_{\mathrm{KL}}\big(p(\mathbf{x})||q(\mathbf{x})\big) = \int_{\mathbf{x}\in\mathbb{X}} p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x}.$$

KLD is a measure of similarity (dissimilarity) between two probability distributions, whose the smaller indicates more similarity. KLD is zero if and only if the two distribution are identical. For two 2-dimensional Gaussian distributions, $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, the KLD has a closed form expression [101, eq. (2)]

$$\mathrm{D}_{\mathrm{KL}}\big(p(\mathbf{x})||q(\mathbf{x})\big) = \frac{1}{2}\Big(\ln \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|} + (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) + \mathrm{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) - 2\Big),$$

where $\{\mu_k\}_{k\in\{0,1\}}$ and $\{\Sigma_k\}_{k\in\{0,1\}}$ are the means and the covariance matrices of the two Gaussian distributions, respectively. For two distributions $p(\mathbf{x})$ and $q(\mathbf{x})$ which are not in the form of any standard distribution, the KLD can be computed by Monte Carlo method [101, eq. (4)]. By drawing M i.i.d. samples $\{\mathbf{x}\}_{n=1}^M$ from the distribution $p(\mathbf{x})$, we have

$$\frac{1}{n}\sum_{n=1}^{M} \ln \frac{p(\mathbf{x}_n)}{q(\mathbf{x}_n)} \to \mathrm{D}_{\mathrm{KL}}\big(p(\mathbf{x})||q(\mathbf{x})\big)$$

as $n \to \infty$. Rényi divergence of order α is a more general divergence measure. The parameter α gives the potential improvement of identifying the similarity of two distributions in application [102]. But tuning the optimal α is not in the scope of this paper, so we use Kullback–Leibler divergence as the similarity measure for simplicity.

We follow [49] for Our graph theoretic notations and definitions. A *graph*, is a triplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{1, \ldots, N\}$ is the *node set* and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the *edge set*, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a *adjacency* matrix such that $\mathsf{a}_{\mathsf{ij}} = 1$ if $(i, j) \in \mathcal{E}$ and $\mathsf{a}_{ij} = 0$, otherwise. An edge (i, j) from i to j means that agents i and j can communicate. A *path* is a sequence of nodes connected by edges. A *connected graph* is an undirected graph in which for every pair of nodes there is a path connecting them.

To develop our distributed density estimator in Section 5.5.1, we rely on the *dynamics active weighted average consensus algorithm* that is shown in Algorithm 1. In a dynamics active weighted average consensus at any time, only a subset of the agents are active, meaning that only a subset of agents collects measurements r^i. The objective then is to enable all the agents, both active and passive, to obtain the weighted average of the collected measurements, $\frac{\sum_i \eta^i(l)\mathsf{r}^i(l)}{\sum_i \eta^i(l)}$, without knowing the set of active agents. Here, $\eta^i(l) = 0$ if i is passive at time step l and $\eta^i(l) \in \mathbb{R}_{>0}$ if i is active. Theorem 4.5.1 shows that Algorithm 1, starting at any $z^i, y^i(0) \in \mathbb{R}$, makes $\mathbf{y}^i(l)$ track the time varying weighted average signal $\frac{\sum_i \eta^i(l)\mathsf{r}^i(l)}{\sum_i \eta^i(l)}$ with a bounded tracking error as $l \to \infty$. Moreover, if the weights and reference signals are static, the tracking error vanishes with time, i.e., $\lim_{l\to\infty} \mathbf{y}^i(l) = \frac{\sum_i \eta^i \mathsf{r}^i}{\sum_i \eta^i}$.

Algorithm 1 Active weighted average consensus algorithm $[\mathbf{y}^i, \mathbf{z}^i, \mathbf{v}^i] \leftarrow \mathrm{Con}(\eta^i, \mathbf{r}^i, \mathbf{z}_0^i, \mathbf{v}_0^i)$

Require: Weight η^i, reference $\mathbf{r}^i$, number of loops L, a small enough number $\delta_c > 0$.
Initialization: $\mathbf{z}^i(1) = \mathbf{z}_0^i$ and $\mathbf{v}^i(1) = \mathbf{v}_0^i$
for $l = 1 : L$ **do**

$$
\begin{aligned}
\mathbf{y}^i(l) &= \mathbf{z}^i(l) + \eta^i(l)\mathbf{r}^i(l), \\
\mathbf{z}^i(l+1) &= \mathbf{z}^i(l) - \delta_c \eta^i(l)(\mathbf{y}^i(l) - \mathbf{r}^i(l)) \\
&\quad - \delta_c \sum_{j=1}^{N} \mathsf{a}_{ij}(\mathbf{y}^i(l) - \mathbf{y}^j(l)) - \delta_c \sum_{j=1}^{N} \mathsf{a}_{ij}(\mathbf{v}^i(l) - \mathbf{v}^j(l)), \\
\mathbf{v}^i(l+1) &= \mathbf{v}^i(l) + \delta_c \sum\nolimits_{j=1}^{N} \mathsf{a}_{ij}(\mathbf{y}^i(l) - \mathbf{y}^j(l)).
\end{aligned}
$$

end for
return $\mathbf{y}^i(l), \mathbf{z}^i(l+1), \mathbf{v}^i(l+1)$

5.3 Problem Definition and Objective

We consider a mobile deployment problem for a group of mobile agents over a set of dense targets $\{\mathbf{x}_t^n\}_{n=1}^M \subset \mathbb{R}^2$ on a planar ground with the objectives such as event detection, wireless communication or monitoring, which we refer to it in general term as providing a 'service'. The probability density distribution $p(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^2$, which is unknown to the agents, represents the density distribution of the targets. The mobile agents communicating over a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ consist of two types. There are a set $\mathcal{V}_\mathrm{a} \subseteq \mathcal{V}$ of *active* agents that have the capability to actively detect the targets and a set $\mathcal{V}_\mathrm{s} = \{1, \cdots, N\} \subseteq \mathcal{V}$ *service* agents that are deployed to provide the targets with a service, see Fig 5.1. Unlike some existing literature like [77], $\mathcal{V}_\mathrm{a}$ and $\mathcal{V}_\mathrm{s}$ do not have to be mutually exclusive. We assume that the active agents have partitioned the area such that each target is detected only by one active agent, i.e., no overlapping detection.

We let $(\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i) \in \mathbb{R}^2 \times [0, 2\pi]$ be the pose (position and orientation) of service agent $i \in \mathcal{V}_\mathrm{s}$. The QoS provided by a service agent $i \in \mathcal{V}_\mathrm{s}$ is given by conditional probability distribution function $Q^i(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i) = z^i(\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i) q^i(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)$, $\mathbf{x} \in \mathbb{R}^2$, where $z^i(\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i) = \int_{\mathbf{x}\in\mathbb{R}^2} Q^i(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i) d\mathbf{x}$ is the normalization constant and $q^i(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)$ is the normalized density distribution of QoS of agent i. We define the collective QoS provided by service agents by the probability density distribution

$$q(\mathbf{x}|\{\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i\}_{i\in\mathcal{V}_\mathrm{s}}) = \frac{\sum_{i\in\mathcal{V}_\mathrm{s}} Q^i}{\int_{\mathbf{x}\in\mathbb{R}^2} \sum_{i\in\mathcal{V}_\mathrm{s}} Q^i d\mathbf{x}} = \frac{\sum_{i\in\mathcal{V}_\mathrm{s}} z^i q^i}{\sum_{i\in\mathcal{V}_\mathrm{s}} z^i} = \sum_{i\in\mathcal{V}_\mathrm{s}} \omega_\mathrm{s}^i q^i, \tag{5.1}$$

where $\omega_\mathrm{s}^i = \frac{z^i}{\sum_{i\in\mathcal{V}_\mathrm{s}} z^i}$ represents the relative service capability of agent i among $\mathcal{V}_\mathrm{s}$.

Our objective in this paper is to first enable all the agents, both active and service agents, obtain an estimate a mixture model $\hat{p}(\mathbf{x})$ of the density distribution of the targets in distributed manner. Then, design a distributed deployment strategy to re-position the service

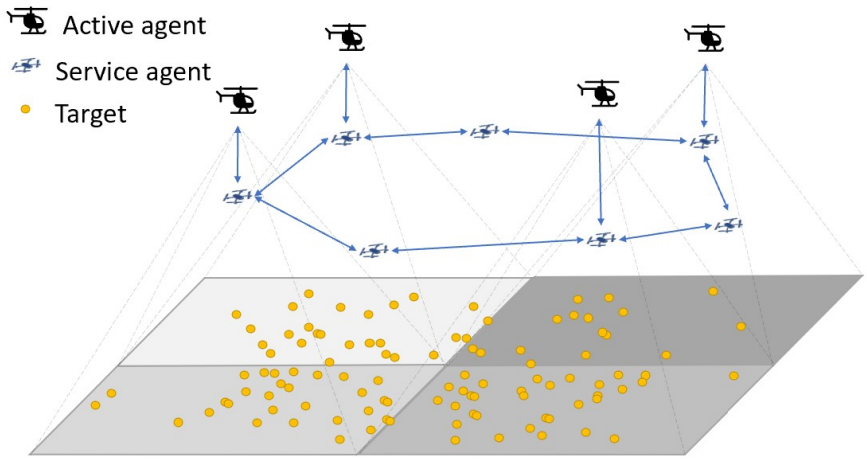

Figure 5.1 – A multi-agent system with active agents and service agents.

agents in a way that their collective QoS serves the targets in an efficient manner. In other words, we seek locations and orientations for service agents such that the collective QoS distribution q is as much similar to as possible to the estimate target density distribution $\hat{p}$. The optimal solution for the deployment objective can be obtained from

$$\{\mathbf{x}_{\mathrm{s}}^{i}, \theta_{\mathrm{s}}^{i}\}_{i\in\mathcal{V}_{\mathrm{s}}} = \arg\min \mathrm{D}_{\mathrm{KL}}\left(\hat{p}(\mathbf{x})||q(\mathbf{x})\right). \tag{5.2}$$

However, we note that $\hat{p}(\mathbf{x})$ and $q(\mathbf{x})$ are mixture distributions, and KLD of mixtures provably does not admit a closed-form formula. In practice these kind of KLDs are usually estimated by using costly Monte-Carlo sampling simulation [103]. Moreover, the collective QoS distribution $q(\mathbf{x})$ contributed by each agent's QoS distribution, $\omega_{\mathrm{s}}^{i}q^{i}$, $i \in \mathcal{V}_{\mathrm{s}}$, is a global information. Accordingly, designing a distributed solver for (5.2) is challenging. Therefore, in this paper, we seek a suboptimal solution for (5.2) that can be implemented in a distributed manner and has low computational complexity.

5.4 Overview of the Proposed Mobile Agent Deployment Solution

Our proposed distributed solution to meet our objective stated in Section 5.3 is the two-stage process depicted in Fig. 5.2. In our first stage, we use a GMM with N Gaussian bases to model the target density distribution. In our setting, the active agents $\mathcal{V}_{\mathrm{a}}$ detect the positions of the targets, considered as the sampled data from the unknown distribution $p(\mathbf{x})$. Then, a distributed EM algorithm, which uses a set of active weighted average consensus algorithm, is used to enable both active and service agents obtain a coherent estimate of the parameters of the N Gaussian bases. As a result, the target density distribution is decomposed into N Gaussian bases. These Gaussian bases partition the target area into N subregions each of which represents a Gaussian basis. Then, our second consecutive stage is an agent allocation process following an optimal mass transport framework. In this allocation process first each service agent $i \in \mathcal{V}_{\mathrm{s}}$ computes the KLD between its QoS distribution, $\omega_{\mathrm{s}}^i q^i$, and each subregion's Gaussian basis estimated from stage 1. Then, a distributed assignment problem is formulated with the KLDs as the cost of deploying the agent to the each respective subregion. As a result, each agent is paired with a subregion and the summation of the divergences corresponding to each paired agent's QoS distribution and subregion's Gaussian basis is minimized. The last step in this stage is a transportation process in which we implement a local controller to drive the agents to their assigned destinations in finite time. We present the details of each stage in the following sections.

Then active agents $\mathcal{V}_{\mathrm{a}}$ detect the targets again and all agents $\mathcal{V}$ return to stage 1 process. This iterative scheme allows the agents to cover the slowly moving targets.

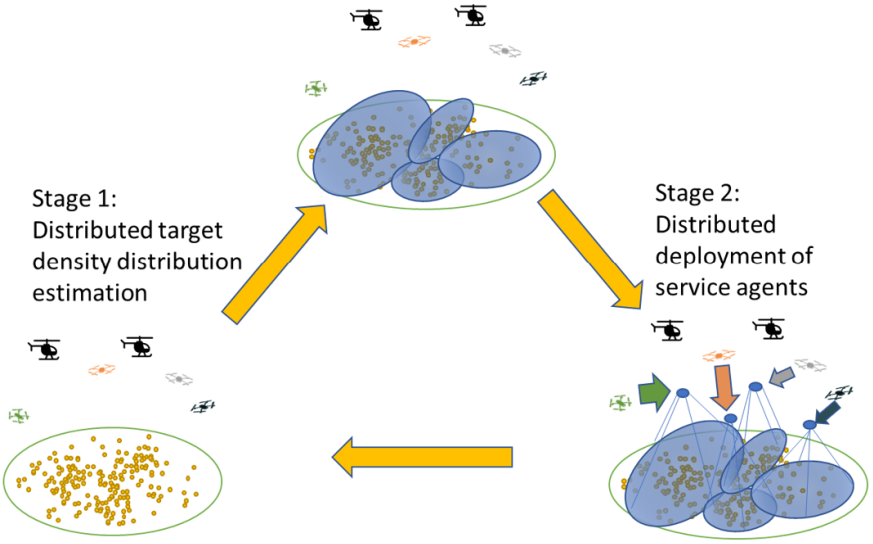

Figure 5.2 – The proposed two-stage distributed deployment solution.

5.5 Stage 1: distributed target density distribution estimation

5.5.1 Consensus-based distributed EM algorithm for GMM

GMM is characterized by finite sum of Gaussian bases with different weights, means and covariance matrices. Let $\mathbf{x} \in \mathbb{R}^2$ be the observed target's position drawn from a mixture of N Gaussian bases with the distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\mu}_k \in \mathbb{R}^2$ is the mean and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{2\times 2}$ is the covariance matrix for $k \in \mathcal{K} = \{1, \cdots, N\}$. Let $z \in \mathbb{R}$ be the indicator which indicates the variable $\mathbf{x}$ belongs to k^{th} Gaussian basis when $z = k$. The variable z is not observed so z is also called hidden variable or latent variable. The probability of drawing a variable from the k^{th} Gaussian basis is denoted as $\pi_k := \Pr(z = k)$. The distribution of $\mathbf{x}$ given the k^{th} mixture basis is Gaussian, i.e., $\hat{p}(\mathbf{x}|z = k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Therefore, the marginal probability distribution for $\mathbf{x}$ is given by

$$\hat{p}(\mathbf{x}) = \sum_{k=1}^{N} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{5.3}$$

The parameters that should be determined to obtain the estimate $\hat{p}(\mathbf{x})$ are the set $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^N$. Next, we employ the EM algorithm to obtain these parameters [104].

The EM algorithm obtains the maximum likelihood estimates of $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^N$ given M independent detected targets' positions $\{\mathbf{x}_t^n\}_{n=1}^M$. It is an iterative method that alternates between an expectation (E) step and a maximization (M) step. Given a detected target $\mathbf{x}_t^n$, $n \in \{1, \cdots, M\}$, E-step computes the posterior probability

$$\gamma_{kn} := \Pr(z = k|\mathbf{x}_t^n) = \frac{\pi_k \mathcal{N}(\mathbf{x}_t^n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^N \pi_j \mathcal{N}(\mathbf{x}_t^n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}, \tag{5.4}$$

using the current value of $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^N$. Then, M-step updates the parameter set $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^N$ by the following equations using the current γ_{kn}:

$$\pi_k = \frac{\sum_{n=1}^M \gamma_{kn}}{M}, \tag{5.5a}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^M \gamma_{kn}\mathbf{x}_t^n}{\sum_{n=1}^M \gamma_{kn}}, \tag{5.5b}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^M \gamma_{kn}(\mathbf{x}_t^n - \boldsymbol{\mu}_k)(\mathbf{x}_t^n - \boldsymbol{\mu}_k)^\top}{\sum_{n=1}^M \gamma_{kn}}, \tag{5.5c}$$

for $k \in \{1, \cdots, N\}$. M-step needs the global information to update the parameter set $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^N$ because the summations in (5.5) are over all detected targets $n \in \{1, \cdots, M\}$. However, the information of the targets' positions $\{\mathbf{x}_t^n\}_{n=1}^M$ is distributed among the active agents $\mathcal{V}_\mathrm{a}$. We observe that the right hand side quantities of (5.5) are in the form of (weighted) average. Therefore, we propose a distributed implementation of the EM algorithm, which invokes a set of active weighted average consensus algorithms such that all the agents, $\mathcal{V} = \mathcal{V}_\mathrm{a} \cup \mathcal{V}_x$ obtain an approximate value of (5.5) by locally exchanging the information with their neighbors. Suppose each agent $i \in \mathcal{V}$ maintains a local copy of the parameter set of the Gaussian bases $\{\pi_k^i, \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i\}_{k=1}^N$. At the E-step, every active agent $i \in \mathcal{V}_a$ computes γ_{kn} for $k \in \{1, \cdots, N\}$ and $n \in \mathcal{V}_t^i$ where $\mathcal{V}_t^i$ is the set of targets detected by active agent

$i \in \mathcal{V}_{\rm a}$. Then, in the M-step, every agent $i \in \mathcal{V}$ executes Consensus Algorithm 1 with proper setting its weight η^i and reference $\mathbf{r}^i$ to estimate the update of $\{\pi_k^i, \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i\}_{k=1}^N$. It is clear that by setting $\eta^i = |\mathcal{V}_t^i|$ and $\mathbf{r}^i = \frac{\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}}{|\mathcal{V}_t^i|}$ if $i \in \mathcal{V}_{\rm a}$, otherwise, $\eta^i = 0$ and $\mathbf{r}^i = 0$, the consensus variable $\mathbf{y}^i$ in Algorithm 1 asymptotically converges to (5.5a). Similarly, letting $\eta^i = \sum_{n\in\mathcal{V}_t^i}\gamma_{kn}$ and $\mathbf{r}^i = \frac{\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}\mathbf{x}_n}{\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}}$ if $i \in \mathcal{V}_{\rm a}$, otherwise, $\eta^i = 0$ and $\mathbf{r}^i = 0$, $\mathbf{y}^i$ converges to (5.5b); letting $\eta^i = \sum_{n\in\mathcal{V}_t^i}\gamma_{kn}$ and $\mathbf{r}^i = \frac{\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}(\mathbf{x}_n-\boldsymbol{\mu}_k^i)(\mathbf{x}_n-\boldsymbol{\mu}_k^i)^\top}{\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}}$ if $i \in \mathcal{V}_{\rm a}$, otherwise, $\eta^i = 0$ and $\mathbf{r}^i = 0$, $\mathbf{y}^i$ converges to (5.5c). The proposed consensus based distributed EM algorithm is summarized in Algorithm 2.

Remark 5.5.1. We emphasize that, in [75], the authors also chose the Gaussian distribution as the basis function to fit the priority function (target density function) but the Gaussian bases are assumed fixed and given. The agents only learn the weight of each Gaussian basis. However, in our GMM, not only the weights but also the parameters (means and covariance matrices) of Gaussian bases are estimated from the proposed EM algorithm so our model has higher degrees of freedom to fit the true distribution of targets more precisely. □

Because the algorithm is terminated in a finite time. It is expected that $\hat{p}^i(x)$ of each active agent i be slightly different than other active agent. In what follows we let,

$$\hat{p}^i(\mathbf{x}) = \sum_{k=1}^{N} \pi_k^i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i), \tag{5.6}$$

be the local final estimate of agent $i \in \mathcal{V}_{\rm a}$.

5.5.2 Numerical demonstration

We demonstrate a numerical simulation to show the performance of the proposed distributed EM algorithm. Consider a group of 6 mobile agents where $\mathcal{V}_{\rm a} = \{1, 2, 3, 6\}$ are the active agents that monitor the targets to enable the service agents $\mathcal{V}_{\rm s} = \mathcal{V} = \{1, 2, 3, 4, 5, 6\}$ to

Algorithm 2 Consensus based distributed EM algorithm for GMM

Require: detected targets set $\{\mathbf{x}_t^n\}_{n\in\mathcal{V}_t^i}$ by agent i, number of Gaussian bases N, number of loops T

Initialization: $\{\pi_k^i, z_{\pi,k}^i, v_{\pi,k}^i\}_{k=1}^N$,$\{\boldsymbol{\mu}_k^i, \mathbf{z}_{\mu,k}^i, \mathbf{v}_{\mu,k}^i\}_{k=1}^N$, $\{\boldsymbol{\Sigma}_k^i, \mathbf{z}_{\Sigma,k}^i, \mathbf{v}_{\Sigma,k}^i\}_{k=1}^N$,

for $t = 1 : T$ **do**

E-step:

if $i \in \mathcal{V}_\mathrm{a}$ **then**

Compute γ_{kn} in (5.4) using the current value of $\{\pi_k^i\, \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i\}$ for $k = \{1, \cdots, N\}$ and $n \in \mathcal{V}_t^i$.

end if

M-step:

for $k = 1 : N$ **do**

if $i \in \mathcal{V}_\mathrm{a}$ **then**

$$[\pi_k^i, z_{\pi,k}^i, v_{\pi,k}^i] \leftarrow \mathrm{Con}(|\mathcal{V}_t^i|, \frac{\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}}{|\mathcal{V}_t^i|}, z_{\pi,k}^i, v_{\pi,k}^i)$$

$$[\boldsymbol{\mu}_k^i, \mathbf{z}_{\mu,k}^i, \mathbf{v}_{\mu,k}^i] \leftarrow \mathrm{Con}(\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}, \frac{\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}\mathbf{x}_n}{\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}}, \mathbf{z}_{\mu,k}^i, \mathbf{v}_{\mu,k}^i)$$

$$[\boldsymbol{\Sigma}_k^i, \mathbf{z}_{\Sigma,k}^i, \mathbf{v}_{\Sigma,k}^i] \leftarrow \mathrm{Con}(\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}, \frac{\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}(\mathbf{x}_n-\boldsymbol{\mu}_k^t)(\mathbf{x}_n-\boldsymbol{\mu}_k^t)^\top}{\sum_{n\in\mathcal{V}_t^i}\gamma_{kn}}, \mathbf{z}_{\Sigma,k}^i, \mathbf{v}_{\Sigma,k}^i)$$

else

$$[\pi_k^i, z_{\pi,k}^i, v_{\pi,k}^i] \leftarrow \mathrm{Con}(0, 0, z_{\pi,k}^i, v_{\pi,k}^i)$$
$$[\boldsymbol{\mu}_k^i, \mathbf{z}_{\mu,k}^i, \mathbf{v}_{\mu,k}^i] \leftarrow \mathrm{Con}(0, 0, \mathbf{z}_{\mu,k}^i, \mathbf{v}_{\mu,k}^i)$$
$$[\boldsymbol{\Sigma}_k^i, \mathbf{z}_{\Sigma,k}^i, \mathbf{v}_{\Sigma,k}^i] \leftarrow \mathrm{Con}(0, 0, \mathbf{z}_{\Sigma,k}^i, \mathbf{v}_{\Sigma,k}^i)$$

end if

end for

end for

return $\{\pi_k^i, \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i\}_{k=1}^N$

obtain an estimate of the density distribution $p(x)$ of a group of $M = 1000$ targets. The agents $\mathcal{V}$ communicate over a connected graph whose adjacency matrix is

$$\mathbf{A} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & -1 \\ -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 \\ -1 & 0 & -1 & 0 & -1 & 0 \end{bmatrix}.$$

The numbers of the targets detected by agent $i \in \mathcal{V}_\mathrm{a}$ are $|\mathcal{V}_\mathrm{t}^1| = 100$, $|\mathcal{V}_\mathrm{t}^2| = 250$, $|\mathcal{V}_\mathrm{t}^3| = 450$, and $|\mathcal{V}_\mathrm{t}^6| = 200$. The agents, both active and service, execute the distributed EM Algorithm 2 to estimate the parameters of the Gaussian bases of the target density distribution. In the simulation, the number of the iteration-loops of the consensus algorithm and EM algorithm are $L = 20$ and $T = 50$, respectively. Agents' estimation results are illustrated in Fig. 5.3 where the black circle's represent the targets, the elliptic footprints are the 3-σ uncertainty ellipses of the 6 Gaussian bases $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i), k \in \{1, \cdots, 6\}$ estimated by agent i and the thickness of the elliptic footprint represents π_k^i. The result shows that with the proposed distributed EM algorithm, the agents successfully estimate the parameter of GMM for the target's distribution though agent 4 and 5 do not detect any target. We note that with the help of the consensus algorithm all agents get the approximately same estimation results. The proposed distributed EM algorithm is a approximate process of the standard centralized EM algorithm. The accuracy of the approximation depends on the number L of loops of consensus algorithm 1. Theoretically, if $L \to \infty$ the approximation is exact because the weight η^i and the reference r^i are static in each M-step. In practice, the choice of finite L is a trade of between the accuracy of the approximation and the consumption of communication among the agents. Figure 5.4 shows the log-likelihood, $\ln \Pr(\{\mathbf{x}_t^n\}_{n=1}^M | \{\pi_k^4\, \boldsymbol{\mu}_k^4, \boldsymbol{\Sigma}_k^4\}_{k=1}^N)$, at the 50th iteration of the distributed EM versus L. We can observe that with a proper large L, the performance of the propose distributed EM algorithm is close to the centralized EM.

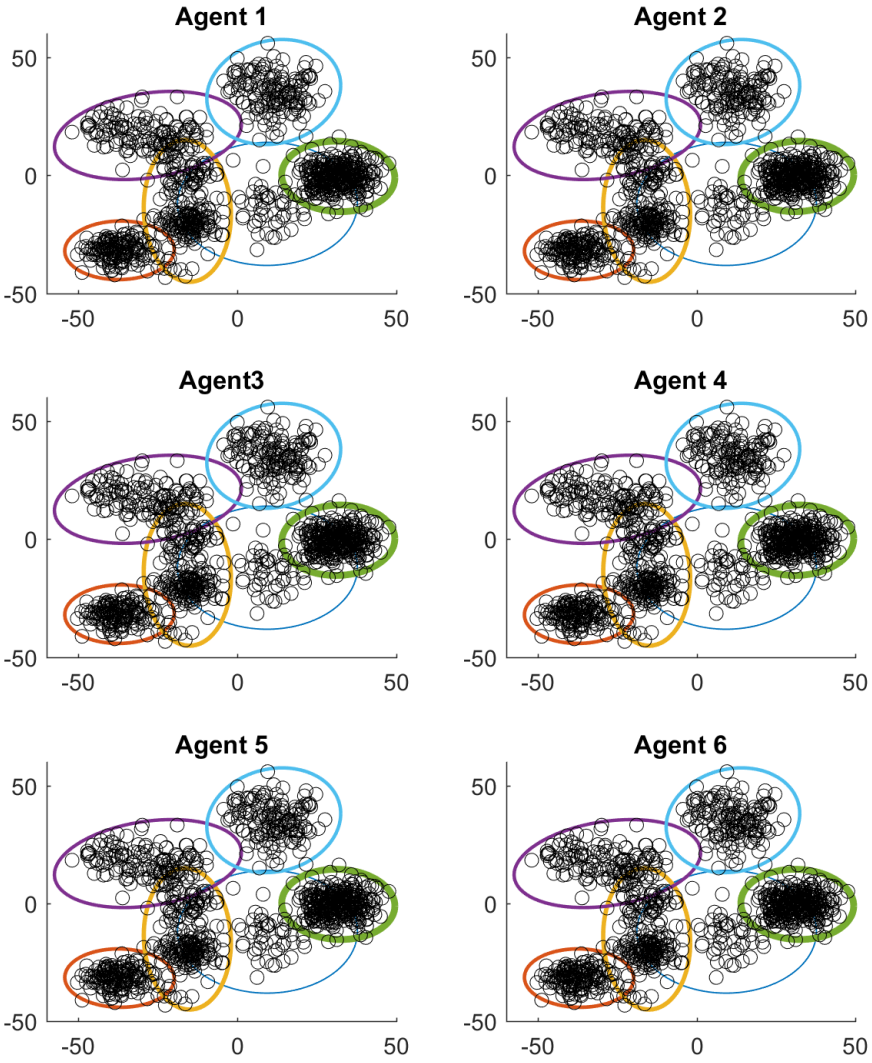

Figure 5.3 – The estimate of GMM of each agents of the demonstration in section 5.5.2

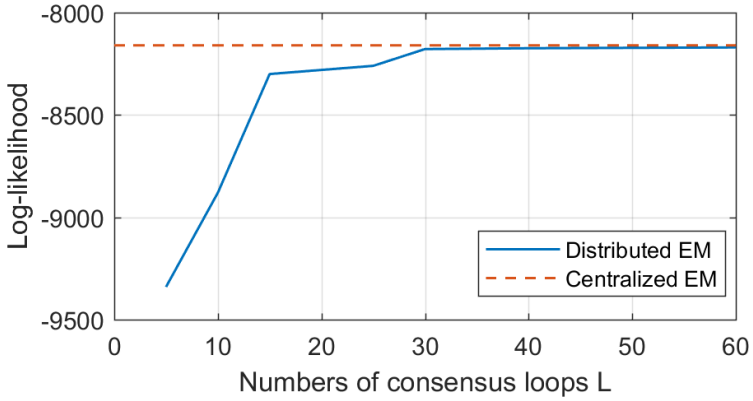

Figure 5.4 – The log-likelihood function for different value of L

5.6 Stage 2: Distributed deployment of service agents

In stage 1, the target density distribution is modeled and estimated by a GMM. The result of GMM intrinsically partitions the area into a set of subregions each of which represents a Gaussian basis. Our suboptimal solution to the deployment problem (5.2) is to deploy each service agent $i \in \mathcal{V}_\mathrm{s} = \{1, \cdots, N\}$ to optimally cover a assigned subregion $k \in \mathcal{K} = \{1, \cdots, N\}$. The service agent assignment is based on the similarity of the agent's QoS distribution, i.e., $\omega_\mathrm{s}^i q^i(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)$, to the Gaussian basis subregion, i.e., $\pi_k^i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)$, such that the summation of the KLD of each assigned agent-subregion pair is minimized. This objective can be formalized as follows. For any service agent $i \in \mathcal{V}_\mathrm{s}$ let

$$
\begin{aligned}
&C_{ik}(\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i) = \mathrm{D}_\mathrm{KL}\left(\pi_k^i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i) || \omega_\mathrm{s}^i q^i(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)\right) \\
=&\int_{\mathbf{x}\in\mathbb{R}^2} \pi_k^i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i) \ln \frac{\pi_k^i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)}{\omega_\mathrm{s}^i q^i(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)} d\mathbf{x} \\
=&\pi_k^i \Big(\ln \frac{\pi_k^i}{\omega_\mathrm{s}^i} + \mathrm{D}_\mathrm{KL}\left(\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i) || q^i(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)\right)\Big),
\end{aligned}
\tag{5.7}
$$

for $k \in \mathcal{K}$. We note that C_{ik} in (5.7) is a continuous function of the service agent's pose $(\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)$. We introduce a binary decision variable $Z_{ik} \in \{0, 1\}$, which is 1 if agent i is assigned to region k and 0 otherwise. With the right notation at hand then, our suboptimal deployment solution is given by

$$
\{\mathbf{x}_\mathrm{s}^{i^\star}, \theta_\mathrm{s}^{i^\star}, \{Z_{ik}^\star\}_{k\in\mathcal{K}}\}_{i\in\mathcal{V}_s} = \arg\min \sum_{i\in\mathcal{V}_\mathrm{s}} \sum_{k\in\mathcal{K}} C_{ik}(\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i) Z_{ik}, \tag{5.8}
$$

$$
\begin{aligned}
&Z_{ik} \in \{0, 1\}, \quad i \in \mathcal{V}_\mathrm{s},\ k \in \mathcal{K}, \\
&\sum_{k\in\mathcal{K}} Z_{ik} = 1, \qquad \forall i \in \mathcal{V}_\mathrm{s}, \\
&\sum_{i\in\mathcal{V}_\mathrm{s}} Z_{ik} = 1, \qquad \forall k \in \mathcal{K}.
\end{aligned}
$$

Next, we introduce a set of manipulations that allows us to arrive at a distributed solution for solving (5.8). For each service agent $i \in \mathcal{V}_{\rm s}$, we start by defining

$$C^{\star}_{ik} = \min_{\mathbf{x}^i_{\rm s}, \theta^i_{\rm s}} C_{ik}(\mathbf{x}^i_{\rm s}, \theta^i_{\rm s}), \quad k \in \mathcal{K}. \tag{5.9}$$

Given (5.9) and the fact that C_{ik} depends on the pose of agent i only, it is straightforward to show that (5.8) can be written in the equivalent form of

$$\begin{aligned} Z^{\star}_{ik} = \arg\min & \sum_{i \in \mathcal{V}_{\rm s}} \sum_{k \in \mathcal{K}} C^{\star}_{ik} Z_{ik}, \\ & Z_{ik} \in \{0,1\}, \quad i \in \mathcal{V}_{\rm s},\ k \in \mathcal{K}, \\ & \sum_{k \in \mathcal{K}} Z_{ik} = 1, \qquad \forall i \in \mathcal{V}_{\rm s}, \\ & \sum_{i \in \mathcal{V}_{\rm s}} Z_{ik} = 1, \qquad \forall k \in \mathcal{K}. \end{aligned} \tag{5.10}$$

where $(\mathbf{x}^{i^\star}_{\rm s}, \theta^{i^\star}_{\rm s})$ for each service agent $i \in \mathcal{V}_{\rm s}$ is equal to minimizer $(\mathbf{x}^{ik^\star}_{\rm s}, \theta^{ik^\star}_{\rm s})$ of the kth (5.9) that corresponds to $Z^{\star}_{ik} = 1$. The equivalent optimization representation (5.10) casts our suboptimal service agent assignment problem in the form of a discrete optimal mass transport problem [105]. In this optimal mass transport problem, the minimum value of (5.7) given in (5.9) can be viewed as the cost of assigning agent i to the kth subregion/basis of the GMM. In Section 5.6.3, we show that the mixed integer programming problem (5.10), in fact can be cast as a linear programming in continuous space, and then solved in a distributed manner using existing optimization algorithms. Once each service agent obtains its assigned pose then in Section 5.6.4, we transport the agents to their assigned region by a finite-time minimum energy control. In what follows, before presenting our equivalent linear programming representation of (5.10), we discuss how we can obtain the minimizers of (5.9). More specifically, in Section 5.6.1 we show that the minimum value for each C_{ik}, when the QoS distributions are Gaussian, happens at $\mathbf{x}^{ik^\star}_{\rm s} = \boldsymbol{\mu}_k$ and the orientation $\theta^{ik^\star}_{\rm s}$ that makes

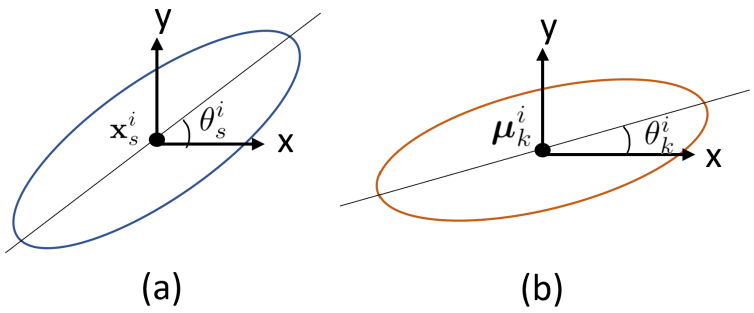

Figure 5.5 – The principal axis angle of (a) agent i's QoS Gaussian distribution and (b) the $k$the subregion/basis of $\hat{p}^i(x)$.

the principal axis of the uncertainty ellipses of the service distribution and the corresponding Gaussian distribution are in parallel. Next, in Section 5.6.2, we discuss how (5.10) can be solved via a numerical approximation method for the case of non-Gaussian QoS distributions.

5.6.1 Gaussian QoS distribution

Let the distribution of QoS provided by agent $i \in \mathcal{V}_\mathrm{s}$ be Gaussian, i.e.,

$$\omega_\mathrm{s}^i q^i(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i) = \omega_\mathrm{s}^i \mathcal{N}(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \boldsymbol{\Sigma}^i(\theta_\mathrm{s}^i)), \tag{5.11}$$

where the mean of the Gaussian distribution is at the agent's location $\mathbf{x}_\mathrm{s}^i$ and the covariance matrix is with principal (major) axis at angle θ_s^i, see Fig. 5.5. Hence, the covariance matrix can be decomposed into $\boldsymbol{\Sigma}^i(\theta_\mathrm{s}^i) = \mathbf{R}(\theta_\mathrm{s}^i)\boldsymbol{\Lambda}^i\mathbf{R}(\theta_\mathrm{s}^i)^\top$, where $\mathbf{R}(\theta_\mathrm{s}^i) = \left[\begin{smallmatrix} \cos\theta_\mathrm{s}^i & -\sin\theta_\mathrm{s}^i \\ \sin\theta_\mathrm{s}^i & \cos\theta_\mathrm{s}^i \end{smallmatrix}\right]$ and $\boldsymbol{\Lambda}^i = \left[\begin{smallmatrix} \sigma_x^i & 0 \\ 0 & \sigma_y^i \end{smallmatrix}\right]$, in which $\sigma_x^i, \sigma_y^i \in \mathbb{R}_{>0}$ with $\sigma_x^i \geq \sigma_y^i$ are known service parameters determines the 'shape' of the service agent i. Similarly, agent i's estimated covariance matrix $\boldsymbol{\Sigma}_k^i$, for the kth subregion/basis of its estimated $\hat{p}(x)$, see (5.6), can be written as $\boldsymbol{\Sigma}_k^i(\theta_k^i) = \mathbf{R}(\theta_k^i)\boldsymbol{\Lambda}_k^i\mathbf{R}(\theta_k^i)^\top$, where $\boldsymbol{\Lambda}_k^i = \left[\begin{smallmatrix} \sigma_{k,x}^i & 0 \\ 0 & \sigma_{k,y}^i \end{smallmatrix}\right]$, in which θ_k^i is the angle of principal (major) axis of the covariance matrix and $\sigma_{k,x}^i, \sigma_{k,y}^i \in \mathbb{R}_{>0}$ with $\sigma_{k,x}^i \geq \sigma_{k,y}^i$ are the variances in the major axis and minor axis direction, respectively, see Fig. 5.5. With the right notation at hand, the theorem below gives a closed-form solution for the minimizer $(\mathbf{x}_\mathrm{s}^{ik\star}, \theta_\mathrm{s}^{ik\star})$ of (5.9).

Theorem 5.6.1. *Consider the optimization problem* (5.9) *and recall* (5.6). *Let* $\omega_{\mathrm{s}}^i q^i(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \theta_{\mathrm{s}}^i)$ *be given by* (5.11). *Then, one of the global minimizer of optimization* (5.9) *is* $(\mathbf{x}_{\mathrm{s}}^{ik^\star}, \theta_{\mathrm{s}}^{ik^\star}) = (\boldsymbol{\mu}_k^i, \theta_k^i)$, *where* θ_k^i *is the angle of the principal axis of* $\boldsymbol{\Sigma}_k^i$. *Moreover,*

$$C_{ik}^\star = \pi_k^i \Big(\ln \frac{\pi_k^i}{\omega_{\mathrm{s}}^i} + \frac{1}{2} \big(\ln \frac{\sigma_x^i \sigma_y^i}{\sigma_{k,x}^i \, \sigma_{k,y}^i} + \frac{\sigma_{k,x}^i \sigma_y^i + \sigma_{k,y}^i \sigma_x^i}{\sigma_x^i \sigma_y^i} - 2 \big) \Big). \tag{5.12}$$

Proof. We first note that since π_k^i and ω_{s}^i are fixed parameters, (5.9) is equivalent to minimize $\mathrm{D}_{\mathrm{KL}} \left(\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i) || \mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \boldsymbol{\Sigma}^i(\theta_{\mathrm{s}}^i)) \right)$. Invoking the closed-form expression for the KLD of two Gaussian distributions [101, eq. (2)], we obtain:

$$\begin{aligned}
&\mathrm{D}_{\mathrm{KL}} \left(\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i) || \mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \boldsymbol{\Sigma}^i(\theta_{\mathrm{s}}^i)) \right) \Big) \\
&= \frac{1}{2} \big(\underbrace{\ln \frac{|\boldsymbol{\Sigma}^i(\theta_{\mathrm{s}}^i)|}{|\boldsymbol{\Sigma}_k^i|}}_{(a)} + \underbrace{(\mathbf{x}_{\mathrm{s}}^i - \boldsymbol{\mu}_k^i)^\top \boldsymbol{\Sigma}^i(\theta_{\mathrm{s}}^i)^{-1} (\mathbf{x}_{\mathrm{s}}^i - \boldsymbol{\mu}_k^i)}_{(b)} \\
&\quad + \underbrace{\mathrm{tr}(\boldsymbol{\Sigma}^i(\theta_{\mathrm{s}}^i)^{-1} \boldsymbol{\Sigma}_k^i)}_{(c)} - 2 \big).
\end{aligned} \tag{5.13}$$

We note that, in (5.13),

$$(a) = \ln \frac{|\mathbf{R}(\theta_{\mathrm{s}}^i) \boldsymbol{\Lambda}^i \mathbf{R}^\top(\theta_{\mathrm{s}}^i)|}{|\mathbf{R}(\theta_k^i) \boldsymbol{\Lambda}_k \mathbf{R}^\top(\theta_k^i)|} = \ln \frac{|\boldsymbol{\Lambda}^i|}{|\boldsymbol{\Lambda}_k|} = \ln \frac{\sigma_x^i \sigma_y^i}{\sigma_{k,x}^i \, \sigma_{k,y}^i},$$

thus (a) is a fix term and does not depend on the decision variable θ_{s}^i. Next, we note that (b) is the only term in (5.13) that depends on $\boldsymbol{\mu}_k^i$ and $\mathbf{x}_{\mathrm{s}}^i$. For any value other than $\mathbf{x}_{\mathrm{s}}^i = \boldsymbol{\mu}_k^i$, (b) returns a positive value, which means that the minimum of (5.13) happens at $\mathbf{x}_{\mathrm{s}}^{ik^\star} = \boldsymbol{\mu}_k^i$. Lastly, we note that (c) in (5.13) reads also as

$$\begin{aligned}
(c) &= \mathrm{tr}(\mathbf{R}(\theta_{\mathrm{s}}^i)(\boldsymbol{\Lambda}^i)^{-1} \mathbf{R}(-\theta_{\mathrm{s}}^i + \theta_k^i) \boldsymbol{\Lambda}_k \mathbf{R}(\theta_k^i)) \\
&= \mathrm{tr}(\mathbf{R}(\theta_{\mathrm{s}}^i - \theta_k^i)(\boldsymbol{\Lambda}^i)^{-1} \mathbf{R}(-\theta_{\mathrm{s}}^i + \theta_k^i) \boldsymbol{\Lambda}_k).
\end{aligned}$$

Now, let $\bar{\theta} = \theta^i_{\mathrm{s}} - \theta^i_k$, $\mathrm{s}\bar{\theta} = \sin(\bar{\theta})$ and $\mathrm{c}\bar{\theta} = \cos(\bar{\theta})$. Then, we can write (c) as

$$
\begin{aligned}
(c) &= \mathrm{tr}\Bigg(\begin{bmatrix} \mathrm{c}\bar{\theta} & -\mathrm{s}\bar{\theta} \\ \mathrm{s}\bar{\theta} & \mathrm{c}\bar{\theta} \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma^i_x} & 0 \\ 0 & \frac{1}{\sigma^i_y} \end{bmatrix} \begin{bmatrix} \mathrm{c}\bar{\theta} & \mathrm{s}\bar{\theta} \\ -\mathrm{s}\bar{\theta} & \mathrm{c}\bar{\theta} \end{bmatrix} \begin{bmatrix} \sigma^i_{k,x} & 0 \\ 0 & \sigma^i_{k,y} \end{bmatrix} \Bigg) \\
&= \frac{(\sigma^i_{k,x}\sigma^i_y + \sigma^i_{k,y}\sigma^i_x)\mathrm{c}^2\bar{\theta} + (\sigma^i_{k,x}\sigma^i_x + \sigma^i_{k,y}\sigma^i_y)\mathrm{s}^2\bar{\theta}}{\sigma^i_x\sigma^i_y}.
\end{aligned}
$$

Let $\alpha = \sigma^i_{k,x}\sigma^i_y + \sigma^i_{k,y}\sigma^i_x$ and $\beta = \sigma^i_{k,x}\sigma^i_x + \sigma^i_{k,y}\sigma^i_y$. Then, (c) reduces

$$
(c) = \frac{\alpha + (\beta - \alpha)\mathrm{s}^2\bar{\theta}}{\sigma^i_x\sigma^i_y}.
$$

Because $\sigma^i_{k,x} \geq \sigma^i_{k,y}$ and $\sigma^i_x \geq \sigma^i_y$, we have $\beta \geq \alpha$ and $(\beta - \alpha)\mathrm{s}^2\bar{\theta}$ is non-negative. Hence, the global minimum of (c) is $\frac{\alpha}{\sigma^i_x\sigma^i_y} = \frac{\sigma^i_{k,x}\sigma^i_y + \sigma^i_{k,y}\sigma^i_x}{\sigma^i_x\sigma^i_y}$ which happens at $\bar{\theta}^\star = n\pi, n \in \{0, 1, \cdots\}$, i.e., $\theta^{ik\star}_{\mathrm{s}} = \theta^i_k + n\pi, n \in \{0, 1, \cdots\}$. To complete the proof, we note that $n = 0$ leads to one of the global minimums $\theta^{ik^\star}_{\mathrm{s}} = \theta^i_k$. □

By virtue of Theorem 5.6.1 we now know that that if the assignment optimization problem (5.10), allocates service agent i to the kth subregion/basis of $\hat{p}^i(x)$, the corresponding final pose of agent i will be in the form of $\mathbf{x}^{i^\star}_{\mathrm{s}} = \boldsymbol{\mu}^i_k$, $\theta^{i^\star}_{\mathrm{s}} = \theta^i_k$.

5.6.2 Non-Gaussian QoS distributions

Let the distribution of QoS provided by agent $i \in \mathcal{V}_{\mathrm{s}}$ be $\omega^i_{\mathrm{s}} q^i(\mathbf{x}|\mathbf{x}^i_{\mathrm{s}}, \theta^i_{\mathrm{s}})$, where $q^i(\mathbf{x}|\mathbf{x}^i_{\mathrm{s}}, \theta^i_{\mathrm{s}})$ is a non-Gaussian density function. The close-form solution obtained for the minimizer of (5.9) in Theorem 5.6.1 was obtained by invoking the closed-form expression that exists for the KLD of Gaussian distributions and also the geometric representation of the covariance matrix of the Gaussian distribution. For non-Gaussian distributions, such closed-form expressions do not necessarily exist. Therefore, for non-Gaussian QoS distributions we propose a numerical procedure that enables us to obtain a suboptimal solution for (5.9), which consequently makes

our solution to the assignment optimization problem (5.10) also suboptimal. Our numerical procedure, relies on discretizing the solution space of the pose to finite set of poses and then choosing the point in this discrete space that gives the minimum value of (5.9).

Let $\mathcal{X}_{\mathrm{s}}^{ik} = \{\mathbf{x}_{\mathrm{s},j}^{ik}\}_{j=1}^{N_x}$ and $\vartheta_{\mathrm{s}}^{ik} = \{\theta_{\mathrm{s},j}^{ik}\}_{j=1}^{N_\theta}$, for $N_x, N_\theta \in \mathbb{R}$, be the sets of candidates of poses of agent i deploying to subregion k. Then agent i can compute (5.7) numerically over the finite sets of poses drawn from $\mathcal{X}_{\mathrm{s}}^{ik} \times \vartheta_{\mathrm{s},}^{ik}$, i.e.,

$$C_{ik}^{\star} \approx \min_{\mathbf{x}_{\mathrm{s}}^i \in \mathcal{X}_{\mathrm{s}}^{ik}, \theta_{\mathrm{s}}^i \in \vartheta_{\mathrm{s}}^{ik}} \mathrm{D}_{\mathrm{KL}} \left(\pi_k^i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i) || \omega_{\mathrm{s}}^i q^i(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \theta_{\mathrm{s}}^i) \right). \tag{5.14}$$

Note that the candidates of poses are chosen empirically; for example, $\mathcal{X}_{\mathrm{s}}^{ik}$ can be chosen from points in the $3-\sigma$ uncertainty ellipse of kth basis of $\hat{p}^i(x)$. The size of N_x and N_θ can be decided based on the trade-off between the solutions resolution and the computation cost of solving (5.14).

5.6.3 Distributed multi-agent assignment problem

The assignment optimization problem (5.10) is an integer optimization problem. As it is known in the discrete optimal mass transport literature [105], by the convex relaxation [106], the integer optimization (5.10) can be transferred to the linear programming problem stated as follows:

$$\begin{aligned} &\min_{Z_{ik} \geq 0} \sum_{i \in \mathcal{V}_{\mathrm{s}}} \sum_{k \in \mathcal{K}} C_{ik}^{\star} Z_{ik} && (5.15) \\ &\text{s.t.} \quad \sum_{k \in \mathcal{K}} Z_{ik} = 1, \qquad \forall i \in \mathcal{V}_{\mathrm{s}}, \\ &\qquad \sum_{i \in \mathcal{V}_{\mathrm{s}}} Z_{ik} = 1, \qquad \forall k \in \mathcal{K}. \end{aligned}$$

Since only agent i knows its own cost $C_{ik}^{\star}$ for $k \in \mathcal{K}$, we are interested in solving optimization problem (5.15) in a distributed way. In general, problem (5.15) may exist several optimal solutions ${Z_{ik}}^{\star}$. We also require the agents agreed on the same optimal assignment plan. A distributed simplex algorithm proposed by [100] can achieve this aim. We rewrite (5.15) to the standard form of linear programming

$$\begin{aligned} &\min_{\mathbf{Z}} {\mathbf{C}^{\star}}^{T}\mathbf{Z} \\ &\text{s.t.} \quad \mathbf{A}\mathbf{Z} = \mathbf{b}, \quad \mathbf{Z} \geq 0. \end{aligned} \tag{5.16}$$

where $\mathbf{b} = \mathbf{1}_{2N}$,

$$\begin{aligned} \mathbf{Z} &= [Z_{11}, \cdots, Z_{1N}, Z_{21}, \cdots, Z_{2,N}, \cdots, Z_{N1} \cdots, Z_{NN}]^{\top}, \\ \mathbf{C}^{\star} &= [C_{11}^{\star}, \cdots, C_{1N}^{\star}, C_{21}^{\star}, \cdots, C_{2,N}^{\star}, \cdots, C_{N1}^{\star} \cdots, C_{NN}^{\star}]^{\top}, \\ \mathbf{A} &= [\mathbf{A}_{11}, \cdots, \mathbf{A}_{1N}, \mathbf{A}_{21}, \cdots, \mathbf{A}_{2,N}, \cdots, \mathbf{A}_{N1} \cdots, \mathbf{A}_{NN}], \end{aligned}$$

in which, $\mathbf{A}_{ik} \in \mathbb{R}^{2N}$ is a column vector with i-th and $(N+k)$-th entries are 1, and others are 0. A column of problem (5.16) is a vector $\mathbf{h}_{ik} \in \mathbb{R}^{1+2N}$ defined as $\mathbf{h}_{ik} = [C_{ik}^{\star} \quad \mathbf{A}_{ik}^{\top}]^{\top}$. The set of all columns is denote by $\mathcal{H} = \{\mathbf{h}_{ik}\}_{i \in \mathcal{V}_{\mathrm{s}}, k \in \mathcal{K}}$. Thus, the linear program (5.16) is fully characterized by the pair $(\mathcal{H}, \mathbf{b})$. The information of $\mathcal{H}$ is distributed in the service agents. Let $\mathcal{P}^i = \{\mathbf{h}_{ik}\}_{k \in \mathcal{K}}$ is the problem column set known by agent $i \in \mathcal{V}_{\mathrm{s}}$, which satisfies $\mathcal{H} = \cup_{i=1}^{N} \mathcal{P}^i$ and $\mathcal{P}^i \cap \mathcal{P}^j = \emptyset, \forall (i,j) \in \mathcal{V}_{\mathrm{s}}$. We assume the communication graph $\mathcal{G}_{\mathrm{s}}(\mathcal{V}_{\mathrm{s}}, \mathcal{E}_{\mathrm{s}})$ of the service agents is connected. Hence the tuple $(\mathcal{G}_{\mathrm{s}}, (\mathcal{H}, \mathbf{b}), \{\mathcal{P}^i\}_{i \in \mathcal{V}_{\mathrm{s}}})$ forms a distributed linear program that can be solved by the distributed simplex algorithm [100]. The result of the optimization problem (5.15) is the optimal plan $Z_{ik}^{\star}$, where $Z_{ik}^{\star} = 1$ means assigning the agent i to the kth subregion with the optimal pose $\mathbf{x}_{\mathrm{s}}^{i\star} = \mathbf{x}_{\mathrm{s}}^{ik\star}$ and $\theta_{\mathrm{s}}^{i\star} = \theta_{\mathrm{s}}^{ik\star}$.

5.6.4 Agents transportation

The last step in Stage 2 of our deployment solution is agents transportation to their corresponding assigned pose. In practice, local controllers are expected to complete this task. One such local controller can be the well-known minimum energy control that can transport the agents to their respective assigned pose in finite time $\tau \in \mathbb{R}_{>0}$ while also enabling the agents to save on transportation energy. Let the local dynamics of agent i (linearized) be given by $\dot{\boldsymbol{\chi}}^i(t) = \mathbf{A}^i \boldsymbol{\chi}^i(t) + \mathbf{B}^i \mathbf{u}^i(t)$, where $\mathbf{u}^i(t)$ is the control vector, and $\boldsymbol{\chi}^i$ is the state vector of agent i, which contains the pose and possibly other states. We assume that $(\mathbf{A}^i, \mathbf{B}^i)$ is controllable. Starting at an initial condition $\boldsymbol{\chi}^i(t_0)$, the minimum energy control is given by

$$\mathbf{u}^i(t) = \mathbf{B}^{i^\top} \mathrm{e}^{\mathbf{A}^{i^\top}(t_0+\tau-t)} \, \mathbf{G}^{i^{-1}} (\boldsymbol{\chi}^{i\star} - \mathrm{e}^{\mathbf{A}^i \tau} \, \boldsymbol{\chi}^i(t_0)) \tag{5.17}$$

for $t \in [t_0, t_0+\tau]$, where $\boldsymbol{\chi}^{i\star}$ is agent i's desired state whose pose component is set to $(\mathbf{x}_{\mathrm{s}}^{i\star}, \theta_{\mathrm{s}}^{i\star})$, and

$$\mathbf{G}^i = \int_0^\tau \mathrm{e}^{\mathbf{A}^i(\tau-t)} \, \mathbf{B}^i \mathbf{B}^{i^\top} \, \mathrm{e}^{\mathbf{A}^{i^\top}(\tau-t)} dt.$$

Control (5.17) is a finite time control that guarantees to drive the agent from it initial state $\boldsymbol{\chi}^i(t_0)$ to it desired state $\boldsymbol{\chi}^{i\star}$ in finite transportation time τ, i.e., $\boldsymbol{\chi}^i(t_0+\tau) = \boldsymbol{\chi}^{i\star}$. Stage 2 of our distributed deployment is finished at $t_0 + \tau$. If the targets are dynamic, our two-stage deployment process can repeat to re-position the service agents in accordance to the changes in targets distribution.

5.7 Demonstrations

In this section, we present two case studies to demonstrate the effectiveness of our proposed suboptimal service matching deployment solution.

5.7.1 Sensors deployment for event detection

Follow the same setting in section 5.5.2, a group of 6 agents with the connected communication graph $\mathcal{G}_{\mathrm{s}}$ aim to cover the 1000 targets. The 6 agents are equipped with a wireless sensor which is used to detect events of interest that occurred with targets. A commonly used sensor model is a probabilistic function under the condition of the sensor location and the event location [107, 108]. If a sensor is at $\mathbf{x}_{\mathrm{s}}^i, i \in \mathcal{V}_{\mathrm{s}}$ and a event happens at $\mathbf{x}_t$, the probability of the sensor detecting the event can be expressed as

$$\Pr(\text{Detected}|\mathbf{x}_{\mathrm{s}}^i, \mathbf{x}_t) = \beta^i \mathrm{e}^{-\alpha^i \frac{(\mathbf{x}_{\mathrm{s}}^i - \mathbf{x}_t)^\top (\mathbf{x}_{\mathrm{s}}^i - \mathbf{x}_t)}{{\gamma^i}^2}} \tag{5.18}$$

where $\alpha^i, \beta^i, \gamma^i$ are sensor i's parameters. We define (5.18) as the QoS of the sensor i at the location $\mathbf{x}_t$. Thus the distribution of QoS of sensor i over the 2-D space $\mathbf{x} \in \mathbb{R}^2$ is

$$\begin{aligned} Q(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i) &= \beta^i \, \mathrm{e}^{-\alpha^i \frac{(\mathbf{x}_{\mathrm{s}}^i - \mathbf{x})^\top (\mathbf{x}_{\mathrm{s}}^i - \mathbf{x})}{{\gamma^i}^2}} \\ &= \sqrt{2\pi |\mathbf{\Lambda}^i|} \beta^i \cdot \frac{1}{\sqrt{2\pi |\mathbf{\Lambda}^i|}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}_{\mathrm{s}}^i - \mathbf{x})^\top {\mathbf{\Lambda}^i}^{-1} (\mathbf{x}_{\mathrm{s}}^i - \mathbf{x})} \\ &= z^i \mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \mathbf{\Lambda}^i), \end{aligned} \tag{5.19}$$

where $\mathbf{\Lambda}^i = \left[\begin{smallmatrix} \sigma^i & 0 \\ 0 & \sigma^i \end{smallmatrix}\right]$, $\sigma^i = \frac{{\gamma^i}^2}{2\alpha^i}$. Hence, the distribution of QoS is a weighted Gaussian distribution. Note that, this Gaussian distribution of the sensor model is isotropic and independent to the sensor's orientation θ_{s}^i because the covariance matrix $\mathbf{\Lambda}_{\mathrm{s}}^i$ has the same eigenvalues. In this demonstration, we consider a more general sensor model with anisotropic

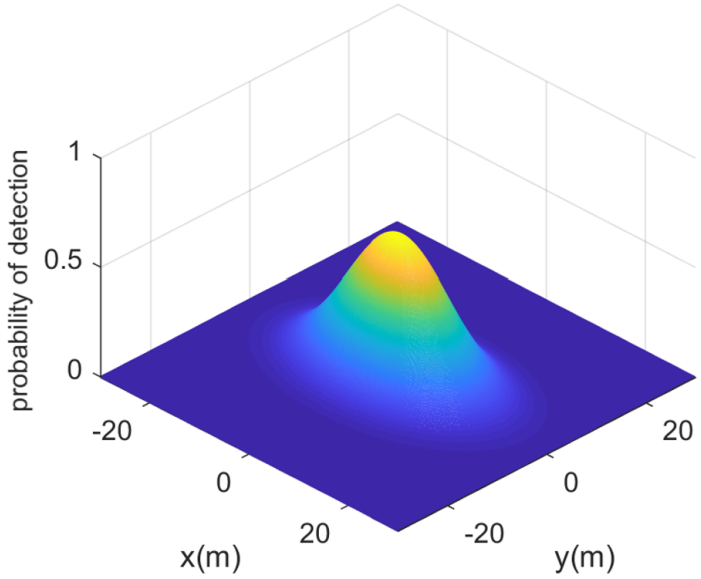

Figure 5.6 – The QoS distribution of a sensor.

sensory capability, i.e. $\mathbf{\Lambda}^i = \begin{bmatrix} \sigma_x^i & 0 \\ 0 & \sigma_y^i \end{bmatrix}$, so the distribution of QoS is expressed as

$$Q(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \theta_{\mathrm{s}}^i) = z^i \mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \mathbf{\Sigma}^i(\theta_{\mathrm{s}}^i)), \tag{5.20}$$

where $\mathbf{\Sigma}^i(\theta_{\mathrm{s}}^i) = \mathbf{R}(\theta_{\mathrm{s}}^i)\mathbf{\Lambda}^i\mathbf{R}^\top(\theta_{\mathrm{s}}^i)$ and θ_{s}^i is the orientation of sensor i. One example is shown in Fig. 5.6. Lastly, the overall density distribution of QoS provided by the 6 sensors is $q(\mathbf{x}|\{\mathbf{x}_{\mathrm{s}}^i, \mathbf{\Sigma}^i(\theta_{\mathrm{s}}^i)\}_{i\in\mathcal{V}_{\mathrm{s}}}) = \sum_{i\in\mathcal{V}_{\mathrm{s}}} \omega_{\mathrm{s}}^i \mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \mathbf{\Sigma}^i(\theta_{\mathrm{s}}^i))$ where $\omega_{\mathrm{s}}^i = \frac{z^i}{\sum_{i=1}^N z^i}$.

The agents estimate the parameters $\{\pi_k^i \boldsymbol{\mu}_k^i, \mathbf{\Sigma}_k^i\}_{k=1}^6$ of the Gaussian bases by the proposed distributed EM algorithm in stage 1. The estimate result is illustrated in Fig 5.3 in Section 5.5.2. The parameters for the sensor's QoS distribution are as follow: $\omega_{\mathrm{s}}^1 = 0.15, \sigma_x^1 = 70, \sigma_y^1 = 25$, $\omega_{\mathrm{s}}^2 = 0.15, \sigma_x^2 = 30, \sigma_y^2 = 15$, $\omega_{\mathrm{s}}^3 = 0.2, \sigma_x^3 = 80, \sigma_y^1 = 30$, $\omega_{\mathrm{s}}^4 = 0.1, \sigma_x^4 = 30, \sigma_y^4 = 30$, $\omega_{\mathrm{s}}^5 = 0.1, \sigma_x^5 = 60, \sigma_y^5 = 40$, and $\omega_{\mathrm{s}}^6 = 0.3, \sigma_x^6 = 30, \sigma_y^6 = 30$. Each agent $i \in \mathcal{V}_{\mathrm{s}}$ evaluates its costs $C_{ik}^\star$ for all $k \in \mathcal{K}$ by (5.12). Then, the agents cooperatively solve the distributed multi-agent assignment problem (5.16) by the means of distributed simplex algorithm [100]. The optimal assignment plan of (5.16) is $Z_{14}^\star = 1$, $Z_{22}^\star = 1$, $Z_{33}^\star = 1$, $Z_{41}^\star = 1$, $Z_{56}^\star = 1$, and $Z_{65}^\star = 1$.

Suppose the agents are with unicycle dynamics

$$\begin{aligned}
\dot{x}^i_{\mathrm{s},x} &= v^i \cos\theta^i_{\mathrm{s}}, \\
\dot{x}^i_{\mathrm{s},y} &= v^i \sin\theta^i_{\mathrm{s}}, \qquad i \in \mathcal{V}_{\mathrm{s}}, \\
\dot{\theta}^i_{\mathrm{s}} &= \omega^i,
\end{aligned} \tag{5.21}$$

where $[x^i_{\mathrm{s},x} \quad x^i_{\mathrm{s},y}]^\top = \mathbf{x}^i_{\mathrm{s}}$, $v^i \in \mathbb{R}$ and $\omega^i \in \mathbb{R}$ are are linear velocity and angular velocity of each agent i, respectively. The agents execute the feedback linearization procedure [85] to achieve the equivalent linear dynamics

$$\begin{bmatrix} \dot{\chi}^i_1 \\ \dot{\chi}^i_2 \\ \dot{\chi}^i_3 \\ \dot{\chi}^i_4 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \chi^i_1 \\ \chi^i_2 \\ \chi^i_3 \\ \chi^i_4 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} u^i_1 \\ u^i_2 \end{bmatrix}, \tag{5.22}$$

by the change of variables

$$\begin{aligned}
\chi^i_1 &= x^i_{\mathrm{s},x}, \\
\chi^i_2 &= v^i \cos\theta^i_{\mathrm{s}}, \\
\chi^i_3 &= x^i_{\mathrm{s},y}, \\
\chi^i_4 &= v^i \sin\theta^i_{\mathrm{s}},
\end{aligned} \tag{5.23}$$

and the compensator

$$\begin{aligned}
\dot{v}^i &= u^i_1 \cos\theta^i_{\mathrm{s}} + u^i_2 \sin\theta^i_{\mathrm{s}}, \\
\omega^i &= \frac{u^i_2 \cos\theta^i_{\mathrm{s}} - u^i_1 \sin\theta^i_{\mathrm{s}}}{v^i}.
\end{aligned} \tag{5.24}$$

Finally, control (5.17) is applied to to drive the agents to their desired state $\boldsymbol{\chi}^{i^\star} = [x^{i^\star}_{\mathrm{s},x} \quad v^{i^\star} \cos\theta^{i^\star}_{\mathrm{s}}$

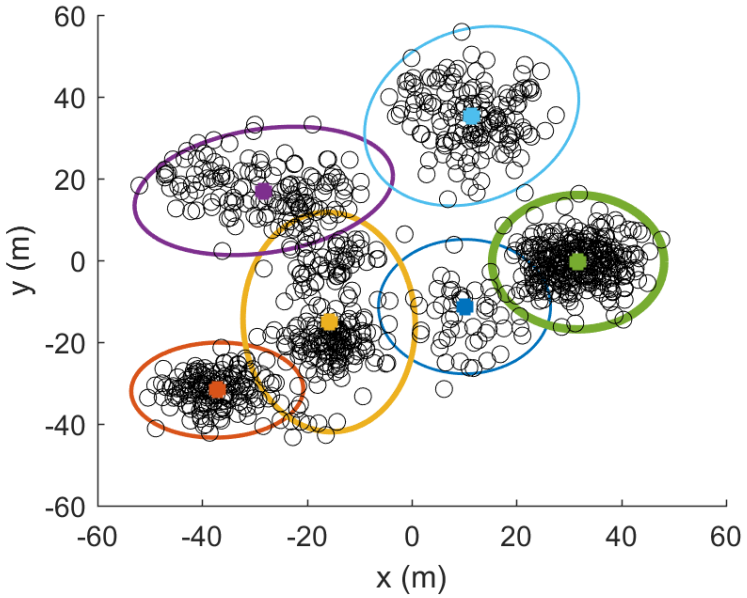

Figure 5.7 – The deployment result of the sensors.

$x_{\mathrm{s},y}^{i^\star} \quad v^{i^\star} \sin\theta_{\mathrm{s}}^{i^\star}]^\top$ corresponding to their assigned subregions and the optimal pose, i.e., $\mathbf{x}_{\mathrm{s}}^{i^\star} = \boldsymbol{\mu}_k^i$ and $\theta_{\mathrm{s}}^{i^\star} = \theta_k^i$ if $Z_{ik}^\star = 1$, where $v^{i^\star} > 0$ is the arrival velocity which we can assign.

Figure 5.7 shows the deployment result where the colored solid circles represent the locations of the sensors while the black circle's represent the targets, the elliptic footprints represent the 3-σ uncertainty ellipses of the 6 sensors' QoS Gaussian distributions, $\mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \boldsymbol{\Sigma}^i(\theta_{\mathrm{s}}^i)), i \in \mathcal{V}_{\mathrm{s}}$ and the thickness of the elliptic footprint represents ω_{s}^i. Moreover, the density distribution of QoS provided the 6 agents (sensors) is illustrated in Fig. 5.8. We can see that with the proposed two-stage deployment strategy, the distribution of QoS efficiently covers the targets.

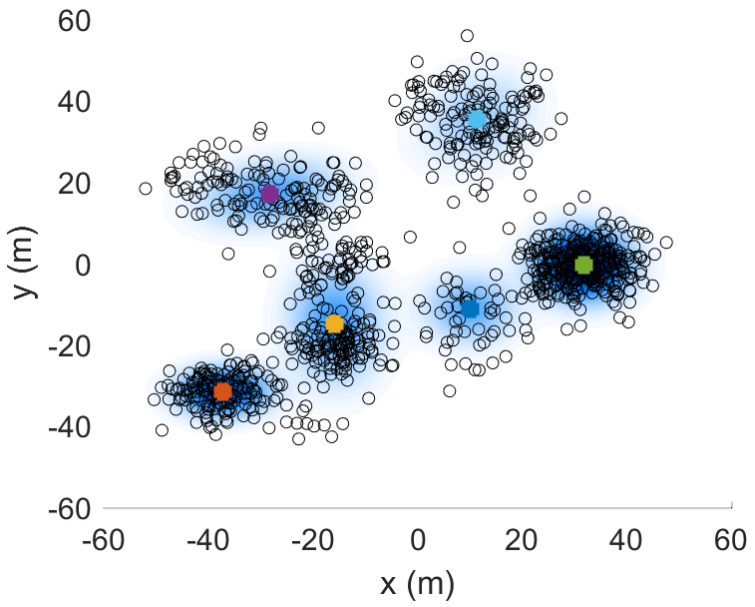

Figure 5.8 – The QoS distribution provided by the 6 sensors.

5.7.2 Deployment of an UAV-aided wireless communication network

The UAV-aided aerial communication has been acknowledged as an alternative to provide wireless service for the ground users in the recent year. In this section, we implement our proposed deployment scheme to deploy the 6 UAVs which provide the wireless communications to the 1000 users on the ground. Assume all UAVs locate at altitude $h = 100$ (m). We focus on the downlink communication channel and the Rician fading model is commonly used to model the communication channel [109]. The instantaneous signal-to-noise ratio (SNR) of the channel from UAV $i \in \mathcal{V}_{\mathrm{s}}$ to a user at $\mathbf{x}_t$ can be expressed as

$$\Gamma = \frac{\gamma^i P_T^i}{N_0^i d^\alpha} \Omega, \tag{5.25}$$

where $d = (\|\mathbf{x}_{\mathrm{s}}^i - \mathbf{x}_t\|^2 + h^2)^{\frac{1}{2}}$ is the distance between the user and the UAV i, $\mathbf{x}_{\mathrm{s}}^i$ is the location of the projection of UAV i onto the ground, P_T^i is the radiated power at the transmitter of UAV i, N_0^i is the noise power, α is the path loss exponent, γ^i is a constant related to the

antenna gain and the operating frequency and $\Omega \in [0, \infty)$ is a random variable follows a non-central chi-square probability distribution given by

$$f_{\Omega}(\omega) = (K+1)\,\mathrm{e}^{-K}\,\mathrm{e}^{-(K+1)\omega}\,\mathrm{I}_0\left(2\sqrt{K(K+1)\omega}\right), \tag{5.26}$$

where $\omega \geq 0$, $\mathrm{I}_0(\cdot)$ is the zero-order modified Bessel function of the first kind and K is the Rician factor. In this application, we define the QoS of UAV i to the user at $\mathbf{x}_t$ as the probability of the channel SNR being greater or equal to a threshold ζ, which can be written as

$$\Pr(\Gamma \geq \zeta) = \mathrm{Q}(\sqrt{2K}, \sqrt{2\zeta(1+K)d^{\alpha}/\beta^i}), \tag{5.27}$$

where Q is the first-order Marcum Q–function and $\beta^i = \frac{\gamma^i P_T^i}{N_0^i}$. As a result, the QoS of UAV i forms a distribution $Q(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i)$ over the 2-D space $\mathbf{x} \in \mathbb{R}^2$. An example is illustrated in Fig. 5.9.

For seeking the suboptimal pose of (5.14) for the UAV, we assume UAV i with a QoS distribution $\omega_{\mathrm{s}}^i q^i(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \theta_{\mathrm{s}}^i)$, shown in Fig. 5.9, is evaluated (5.14) to a subregion k which corresponds to a Gaussian basis $\pi_k^i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i) = 0.2\,\mathcal{N}(\mathbf{x}|\mathbf{0}, \left[\begin{smallmatrix} 2000 & 0 \\ 0 & 1000 \end{smallmatrix}\right])$. Then, the space of $\mathbf{x}_{\mathrm{s}}^i = [x_{\mathrm{s},x}^i \quad x_{\mathrm{s},y}^i]^{\top}$ is discretized to $N_x = 81 \times 81$ a set of candidates of locations $\mathcal{X}_{\mathrm{s}}^{ik} = \{\mathbf{x}_{\mathrm{s}.j}^{ik}\}_{j=1}^{81\times 81}$ around the mean, $\mathbf{0}$, of the Gaussian basis. We numerically compute the KLD of (5.14) over set $\mathcal{X}_{\mathrm{s}}^{ik}$ and show the KLD of each candidate of location in Fig. 5.10. We can observer that the minimal value of KLD is at $\mathbf{x}_{\mathrm{s}}^{ik\star} = \mathbf{0}$, that is the suboptimal location of UAV i is at the mean of the Gaussian basis of subregion. Note that since (5.27) is a function of distance d, the QoS distribution is isotropic and independent to the UAV's orientation θ_{s}^i.

In this demonstration, the communication graphs $\mathcal{G}$ and $\mathcal{G}_{\mathrm{s}}$ of the 6 UAVs, and the set $\mathcal{V}_{\mathrm{a}}$ and $\mathcal{V}_{\mathrm{s}}$ follow the same setting in section 5.5.2. The parameters in (5.27) are $K = 10$ (dB), $\zeta = 5$ (dB) and $\alpha = 2$. The parameters related to the UAVs are $\beta^1 = \beta^2 = 50$

(dB), $\beta^3 = \beta^4 = 51.8$ (dB) and $\beta^5 = \beta^6 = 53$ (dB). In stage 1, the 6 UAVs execute the distributed EM algorithm 2 to estimate the parameters of Gaussian bases of the users' density distribution. The estimation results of the UAVs are shown in Fig. 5.11, in which we can see that all UAVs get almost the same Gaussian bases representing the users' density. Then, in stage 2, each agent $i \in \mathcal{V}_\mathrm{s}$ numerically evaluate $C^\star_{ik}, k \in \mathcal{K}$ in (5.14) with the known suboptimal pose being $\mathbf{x}_\mathrm{s}^{ik\star} = \boldsymbol{\mu}_k^i$. Then, the agents cooperatively solve the distributed multi-agent assignment problem (5.16) by the means of distributed simplex algorithm [100]. The optimal assignment plan of (5.16) is $Z^\star_{11} = 1$, $Z^\star_{24} = 1$, $Z^\star_{33} = 1$, $Z^\star_{46} = 1$, $Z^\star_{52} = 1$, and $Z^\star_{65} = 1$.

Assume the UAVs follow the quadcopter small angle approximate linear dynamics [110].

$$\begin{aligned} &\ddot{x}^i_{\mathrm{s},x} = g\psi^i_\mathrm{s}, \quad \ddot{x}^i_{\mathrm{s},y} = -g\phi^i_\mathrm{s}, \quad \ddot{h}^i_\mathrm{s} = \frac{u^i_1 - m^i g}{m^i}, \\ &\ddot{\phi}^i_\mathrm{s} = \frac{L^i}{I^i_x} u^i_2, \quad \ddot{\psi}^i_\mathrm{s} = \frac{L^i}{I^i_y} u^i_3, \quad \ddot{\theta}^i_\mathrm{s} = \frac{1}{I^i_z} u^i_4, \end{aligned} \tag{5.28}$$

where $x^i_{\mathrm{s},x}$ and $x^i_{\mathrm{s},y}$ are the x and y location coordinate, h^i_s is the altitude; ϕ^i_s, ψ^i_s and θ^i_s are the roll, pitch and yaw of the Euler angles, respectively; u^i_1, u^s_2, u^s_3 and u^s_4 are the thrust, roll, pitch, and yaw forces; L^i is the length from the rotors to the center of mass, m^i is the mass; I^i_x, I^i_y and I^i_z are the moments of inertia. Hence the state of the linear system is $\boldsymbol{\chi}^i = [x^i_{\mathrm{s},x} \quad x^i_{\mathrm{s},y} \quad h^i_\mathrm{s} \quad \dot{x}^i_{\mathrm{s},x} \quad \dot{x}^i_{\mathrm{s},y} \quad \dot{h}^i_\mathrm{s} \quad \phi^i_\mathrm{s} \quad \psi^i_\mathrm{s} \quad \theta^i_\mathrm{s} \quad \dot{\phi}^i_\mathrm{s}$
$\dot{\psi}^i_\mathrm{s} \quad \dot{\theta}^i_\mathrm{s}]^\top$. Then, the agents are transported to hovering at the means of their assigned subregions, i.e., $\mathbf{x}^{i^\star}_\mathrm{s} = \boldsymbol{\mu}^i_k$ if $Z^\star_{ik} = 1$, by the control (5.17), in which $\boldsymbol{\chi}^{i^\star} = [x^{i^\star}_{\mathrm{s},x} \quad x^{i^\star}_{\mathrm{s},y} \quad h \quad 0 \quad 0 \quad 0$ $0 \quad 0 \quad \theta^{i^\star}_\mathrm{s} \quad 0 \quad 0 \quad 0]^\top$. Again, since the distribution of QoS of agent i is independent to the orientation (yaw angle) θ^i_s, we can randomly assign a $\theta^{i^\star}_\mathrm{s} \in [0, 2\pi]$.

The density distribution of QoS provided the 6 UAVs after the deployment is illustrated in Fig. 5.12 where the colored solid circles represent the locations of the UAVs and the black circle's represent the users. We can see that with the proposed two-stage deployment

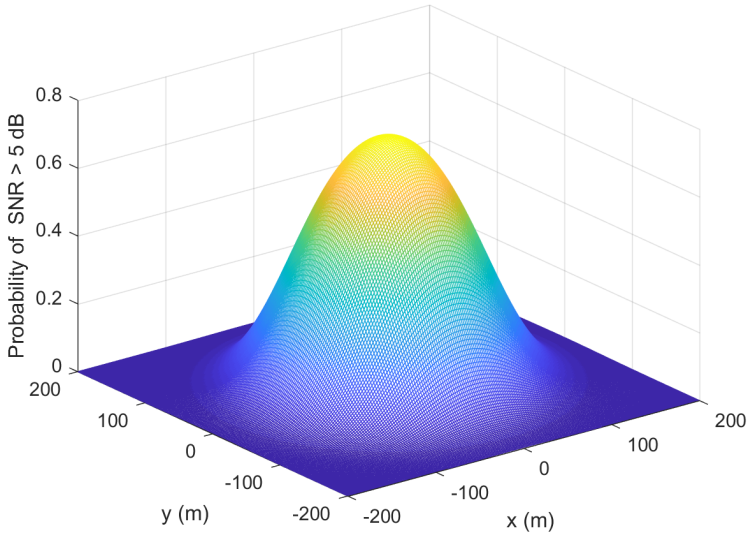

Figure 5.9 – The QoS distribution of an UAV.

strategy, the distribution of QoS efficiently covers the users.

5.8 Conclusion

We proposed a distributed deployment solution for mobile agents to efficiently cover a group of dense targets with their service. In our setting, the exact density distribution of the targets was unknown to the mobile agents. We modeled this unknown density distribution as a GMM. We then proposed a consensus-based EM algorithm that enabled the agents to obtain a local copy of this GMM model. Then we introduced KLD to evaluate the similarity of each pair of agent's QoS distribution and subregion's Gaussian basis. The QoS was first assumed to be a Gaussian distribution and the optimal pose of agent that leads to the smallest KLD was analysed. We then extended the approach to QoS in any distribution so our approach can adapt to wide range of heterogeneous and anisotropic service model. In order to making the distributions of QoS and target density distribution similar, we formulated a distributed multi-agent assignment problem to allocate every agent to a subregion by taking the KLD between the agent and subregion as the assignment cost. Finally, agents with heterogeneous

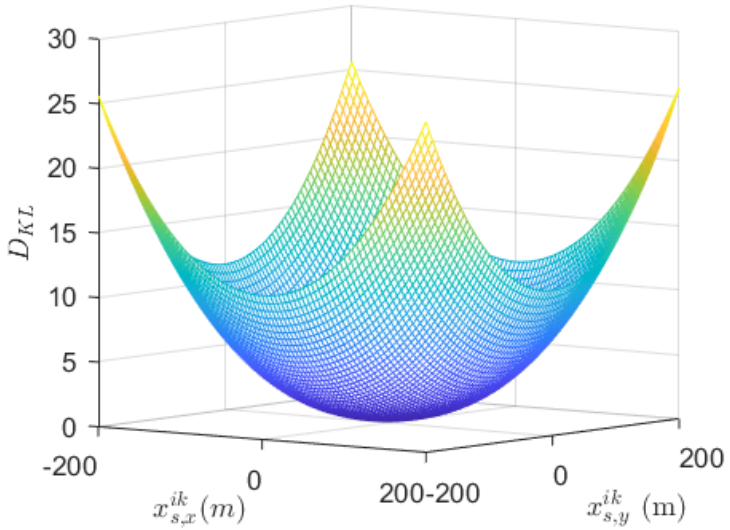

Figure 5.10 – D_{KL} versus candidate locations of $\mathbf{x}_{\mathrm{s}}^{ik} = [x_{\mathrm{s},x}^{ik} \quad x_{\mathrm{s},y}^{ik}]^{\top}$.

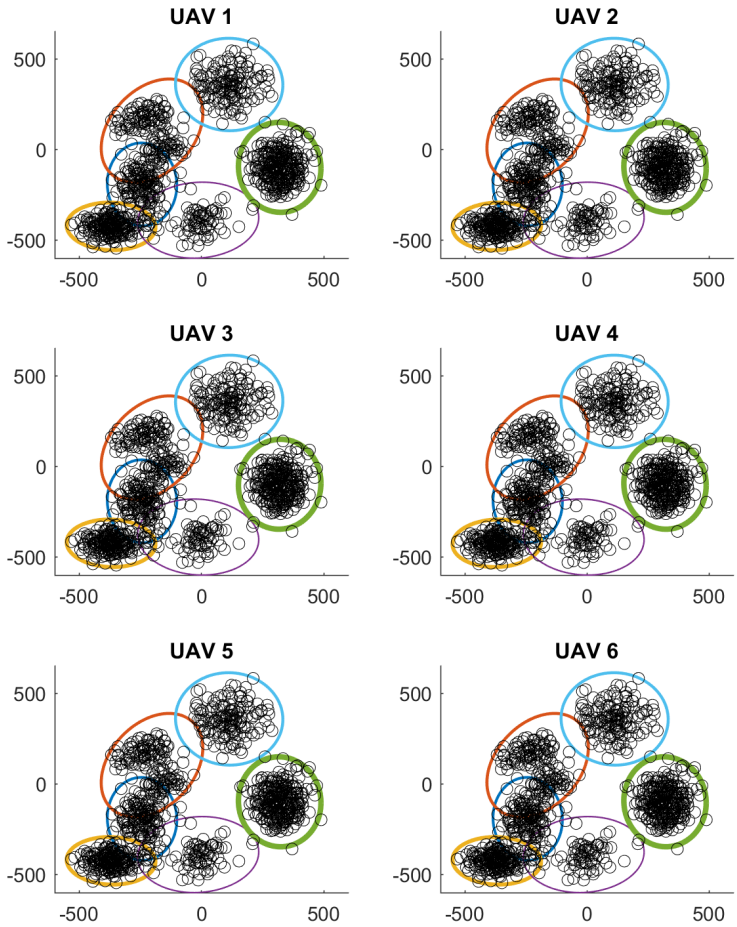

Figure 5.11 – The estimate of GMM of each UAV of the demonstration in section 5.7.2

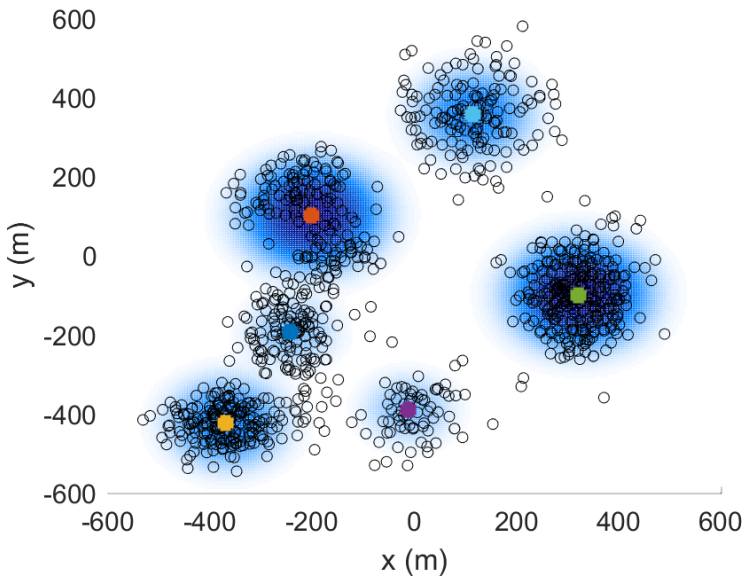

Figure 5.12 – The QoS distribution provided by the 6 UAVs.

linear dynamics was transported to their subregion by the local controller. We illustrated two application examples to show the efficiency of our deployment strategy.

Chapter 6

Conclusion and Future work

6.1 Conclusion

In this dissertation, we investigated four different distributed cooperative tracking problems for a group of autonomous agents that aim to meet a set of leader-following objectives. Chapter 2 considered a single leader-following problem for a group of homogeneous or heterogeneous LTI followers interacting over a directed acyclic graph. Only a subset of the followers has access to the state of the leader in specific sampling times. The dynamics of the leader that generates its sampled states is unknown to the followers. For interaction topologies in which the leader is a global sink in the graph, we proposed a distributed algorithm that allows the followers to arrive at the sampled state of the leader by the time the next sample arrives. Our algorithm is a practical solution for a leader-following problem when there is no information available about the state of the leader except its instantaneous value at the sampling times. Our algorithm also allows the followers to track the sampled state of the leader with a locally chosen offset that can be time-varying. When the followers are mobile agents whose state or part of their state is their position vector, the offset

mechanism can be used to enable the followers to form a transnational invariant formation about the sampled state of the leader. We proved that the control input of the followers to take them from one sampled state to the next one is minimum energy. We also showed in case of the homogeneous followers, after the first sampling epoch the states and inputs of all the followers are synchronized with each other. Four numerical examples demonstrated our results.

Chapter 3 proposed a distributed containment control solution for a group of communicating mobile agents that aim to track the convex hull spanned by a group of moving leaders with unknown dynamics. The communication topology of the mobile agents is described by a strongly connected and weight-balanced directed graph. In our problem setting the agents can communicate in discrete-time and also detect the leaders in specific sampling times. The contribution in this chapter was to show how a group of unicycle robots can use the proposed containment control algorithm to track the convex hull of their jointly monitoring mobile leaders. In the proposed framework, the unicycle robots have continuous-time dynamics but communicate with each other in discrete-time fashion. We demonstrated our results through a numerical example. Then, we observed the containment control problem is a multi-agent system consisted of active agents and passive agents, which motivated us to investigate the dynamic active weighted average consensus problem. Chapter 4 proposed a continuous-time dynamic active weighted average consensus algorithm in which the agents can alternate between active and passive modes depending on their ability to access to their reference input. The algorithm was modeled as a switched linear system whose convergence properties were carefully studied considering the agents' piece-wise constant access to the reference signals and possible piece-wise constant weights. We also studied the discrete-time implementation of this algorithm. Next, we showed how a containment control problem could be cast as an active average consensus problem, and solved efficiently by our proposed dynamic active average consensus algorithm. Numerical examples demonstrated our results.

Finally, chapter 5 proposed a distributed coverage control for mobile agents with heterogeneous and anisotropic QoS distributions. The proposed coverage control deploys the agents to efficiently cover the targets, i.e., making QoS distribution similar to the target density distribution. The target density distribution is unknown in advance. We proposed a consensus-based distributed EM algorithm for agents to estimate the target density distribution which was modeled by a GMM. The GMM not only decomposes the target density distribution to a set of Gaussian bases but also partitions the area into subregions each of which represents a Gaussian basis. We first assumed the QoS distribution of each agent is a Gaussian distribution. KLD was used to evaluate the similarity between each agent's QoS distribution and each subregion's Gaussian basis. Then, a multi-agent assignment problem was formulated under the framework of optimal mass transport to allocate each agent to a subregion by taking the divergence as the cost of assignment. As a result, the summation of the divergences corresponding to each paired agent's QoS distribution and subregion's Gaussian basis is minimal. We then extended this approach to adapt to agents with their QoS in any distribution. Two application examples were demonstrated to show the efficiency of the proposed deployment strategy.

6.2 Future Work

In this dissertation, we proposed four distributed controls: single leader-following control, containment control, dynamic active weighted average control, and coverage control. We outline several directions of this work that can be investigated in the future.

- Discrete-time communication for single leader-following control

  In chapter 2, we have developed the distributed leader-following control to track a single leader. The proposed control needs the follower agents to continuously acquire the state of their neighbors. However, continuous-time communication may not be realistic

in practice. Therefore, an algorithm that only requires discrete-time communication is preferable. Theorem 2.4.1 shows the closed-form of the follower agent's state trajectory when it implements the proposed control (2.26). The closed-form contains the information of the sampled state of the leader. Hence, follower agents are possible to resolve the sampled state of the leader by analyzing their neighbors' state trajectories that sampled in a discrete-time fashion.

- Collision avoidance containment control

  Chapter 3 has provided the containment control to drive the unicycle robots to track a point in the convex hull spanned by the leaders. A practical measure that can enhance this design is to consider collision avoidance and also spreading out the agents in a way that they still stay inside the convex hull.

- Controllable tracking error bounds for active weighted average consensus control

  In chapter 4, we have shown that the tracking error of the proposed control is bounded. It is worth to improve the algorithm to make the size of tracking error bound controllable. Then, users can control the size of the bound based on a trade-off between the desired tracking error and the control effort of their controller. Eq. (4.8) shows the tracking error bound is a function of the convergence parameters (κ_s and λ_s) of switched system $\overline{\mathbf{A}}_{\sigma(t)}$ of (4.4b). Once the relation between the convergence parameters and switched system $\overline{\mathbf{A}}_{\sigma(t)}$ is explicitly resolved (for example relate the parameters to the eigenvalues of $\overline{\mathbf{A}}_p$, $p \in \mathcal{P}$), it is possible to add a coefficient before each term of algorithm (4.4) to control the size of the tracking error bound, as it has been done in [52] for a dynamic average consensus algorithm.

- Using other mixture models for target density distribution in coverage control

  In chapter 5, the first stage of the proposed deployment strategy estimates the unknown target density distribution by a GMM which also partitions the area into subregions. We first investigated the case that the agent's QoS distribution is also a Gaussian

distribution. In the second stage, each agent is paired with a subregion based on the divergence of the agent's QoS Gaussian distribution and subregion's Gaussian basis. We then extended the idea of matching the divergence of two distributions to the case that the agent's QoS is in any form of distribution. However, in this case, the target density distribution was still modeled by a GMM. It is possible to use another distribution as a mixture model's basis. Especially, using the identical agent's QoS distribution as the mixture model's basis may result in a smaller divergence in the second stage. But the mixture model needs a new algorithm to estimate the parameters of the bases (as the EM algorithm for GMM), so developing the algorithm for a mixture model with general basis may be a challenging problem.

- Deployment under constraint

  Our deployment strategy aims to deploy the agents such that their collective QoS is as similar as possible to the spatial distribution of the targets. However, in some applications they may be some constrains such as avoiding overlap in service footprint of the agents in wireless signal coverage that should also be taken into account. In future directions can explore deployment strategy design under such constrains.

Bibliography

[1] S. S. Kia, B. Van Scoy, J. Cortés, R. A. Freeman, K. M. Lynch, and S. Martínez. Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3):40–72, 2019.

[2] J. Lin, A. S. Morse, and B. D. O. Anderson. The multi-agent rendezvous problem. *42nd IEEE International Conference on Decision and Control*, 2:1508–1513, 2003.

[3] W. Ren. Consensus strategies for cooperative control of vehicle formations. *IET Control Theory Applications*, 1(2):505–512, 2007.

[4] H. Su, X. Wang, and Z. Lin. Flocking of multi-agents with a virtual leader. *IEEE Transactions on Automatic Control*, 54(2):293–307, 2009.

[5] Yongcan Cao, Wei Ren, and Magnus Egerstedt. Distributed containment control with multiple stationary or dynamic leaders in fixed and switching directed networks. *Automatica*, 48(8):1586 – 1597, 2012.

[6] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6698–6703, 2005.

[7] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, Sep. 2004.

[8] D. V. Dimarogonas and K. J. Kyriakopoulos. On the rendezvous problem for multiple nonholonomic agents. *IEEE Transactions on Automatic Control*, 52(5):916–922, 2007.

[9] Huiyang Liu, Guangming Xie, and Long Wang. Necessary and sufficient conditions for containment control of networked multi-agent systems. *Automatica*, 48(7):1415 – 1422, 2012.

[10] X. Wang, V. Yadav, and S. N. Balakrishnan. Cooperative uav formation flying with obstacle/collision avoidance. *IEEE Transactions on Control Systems Technology*, 15(4):672–679, July 2007.

[11] S. P. Hou and C. C. Cheah. Can a simple control scheme work for a formation control of multiple autonomous underwater vehicles? *IEEE Transactions on Control Systems Technology*, 19(5):1090–1101, Sep. 2011.

[12] Joseph F Engelberger. *Robotics in practice: management and applications of industrial robots*. Springer Science & Business Media, 2012.

[13] S Sitharama Iyengar and Richard R Brooks. *Distributed Sensor Networks: Sensor Networking and Applications (Volume Two)*. CRC press, 2016.

[14] Manish Kumar, Kelly Cohen, and Baisravan HomChaudhuri. Cooperative control of multiple uninhabited aerial vehicles for monitoring and fighting wildfires. *Journal of Aerospace Computing, Information, and Communication*, 8(1):1–16, 2011.

[15] Anthony Faustine, Aloys N Mvuma, Hector J Mongi, Maria C Gabriel, Albino J Tenge, and Samuel B Kucel. Wireless sensor networks for water quality monitoring and control within lake victoria basin. 2014.

[16] Sara Susca, Francesco Bullo, and Sonia Martinez. Monitoring environmental boundaries with a robotic sensor network. *IEEE Transactions on Control Systems Technology*, 16(2):288–296, 2008.

[17] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Mérouane Debbah. Mobile internet of things: Can uavs provide an energy-efficient mobile architecture? In *2016 IEEE global communications conference (GLOBECOM)*, pages 1–6. IEEE, 2016.

[18] Wei Ren and Randal W. Beard Syrmos. *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer London, 2008.

[19] Wei Ni and Daizhan Cheng. Leader-following consensus of multi-agent systems under fixed and switching topologies. *Systems & Control Letters*, 59:209–217, 2010.

[20] H. Zhang, F. L. Lewis, and A. Das. Optimal design for synchronization of cooperative systems: State feedback, observer and output feedback. *IEEE Transactions on Automatic Control*, 56(8):1948–1952, 2011.

[21] X. Wang and Y. Hong. A distributed control approach to a robust output regulation problem for multi-agent linear systems. *IEEE Transactions on Automatic Control*, 55(12):2891–2895, 2010.

[22] Y. Su and J. Huang. Cooperative output regulation of linear multi-agent systems. *IEEE Transactions on Automatic Control*, 57(4):1062–1066, 2012.

[23] Yiguang Hong, Jiangping Hu, and Linxin Gao. Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica*, 42:1177–1182, 2006.

[24] Yiguang Hong, Guanrong Chen, and Linda Bushnell. Distributed observers design for leader-following control of multi-agent networks. *Automatica*, 44:846–850, 2008.

[25] Zhongkui Li and Wei Ren. Distributed tracking control for linear multiagent systems with a leader of bounded unknown input. *IEEE Transactions on Automatic Control*, 58(2):518–544, 2013.

[26] Yutao Tang, Yiguang Hong, and Xinghu Wang. Distributed output regulation for a class of nonlinear multi-agent systems with unknown-input leaders. *Automatica*, 62:154–160, 2015.

[27] J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

[28] Yongliang Yang and Hamidreza Modares. Leader-follower output synchronization of linear heterogeneous systems with active leader. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2018.

[29] Long Wang and Feng Xiao. Finite-time consensus problems for networks of dynamic agents. *IEEE Transactions on Automatic Control*, 55(4):950–955, 2010.

[30] Shihua Li, Haibo Du, and Xiangze Lin. Finite-time consensus algorithm for multi-agent systems. *Automatica*, 47(8):1706–1712, 2011.

[31] Yongcan Cao and Wei Ren. Distributed Coordinated Tracking With Reduced Interaction via a Variable Structure Approach. *IEEE Transactions on Automatic Control*, 57(1):33–48, 2012.

[32] Michael Defoort, Andrey Polyakov, Guillaume Demesure, Mohamed Djemai, and Kalyana Veluvolu. Leader-follower fixed-time consensus for multi-agent systems with unknown non-linear inherent dynamics. *IET Control Theory & Applications*, 9(14):2165–2170, 2015.

[33] Zongyu Zuo. Nonsingular fixed-time consensus tracking for second-order multi-agent networks. *Automatica*, 54:305–309, 2015.

[34] Y. Zhao, Y. Liu, G. Wen, W. Ren, and G. Chen. Designing distributed specified-time consensus protocols for linear multi-agent systems over directed graphs. *IEEE Transactions on Automatic Control*, pages 1–1, 2018.

[35] Dimos V. Dimarogonas, Magnus Egerstedt, and Kostas J. Kyriakopoulos. A Leader-based Containment Control Strategy for Multiple Unicycles. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 5968–5973. IEEE, 2006.

[36] M Ji, G. Ferrari-Trecate, M Egerstedt, and A Buffa. Containment Control in Mobile Networks. *IEEE Transactions on Automatic Control*, 53(8):1972–1975, sep 2008.

[37] Jianzhen Li, Wei Ren, and Shengyuan Xu. Distributed containment control with multiple dynamic leaders for double-integrator dynamics using only position measurements. *IEEE Transactions on Automatic Control*, 57(6):1553–1559, 2012.

[38] Xiangyu Wang, Shihua Li, and Peng Shi. Distributed Finite-Time Containment Control for Double-Integrator Multiagent Systems. *IEEE Transactions on Cybernetics*, 44(9):1518–1528, sep 2014.

[39] Yongcan Cao, Wei Ren, and Magnus Egerstedt. Distributed containment control with multiple stationary or dynamic leaders in fixed and switching directed networks. *Automatica*, 48(8):1586–1597, aug 2012.

[40] Huiyang Liu, Guangming Xie, and Long Wang. Containment of linear multi-agent systems under general interaction topologies. *Systems & Control Letters*, 61(4):528 – 534, 2012.

[41] Jie Mei, Wei Ren, and Guangfu Ma. Distributed containment control for Lagrangian networks with parametric uncertainties under a directed graph. *Automatica*, 48(4):653–659, 2012.

[42] Luca Galbusera, Giancarlo Ferrari-Trecate, and Riccardo Scattolini. A hybrid model predictive control scheme for containment and distributed sensing in multi-agent systems. *Systems & Control Letters*, 62(5):413 – 419, 2013.

[43] H. Liu, L. Cheng, M. Tan, and Z. Hou. Containment control of double-integrator multi-agent systems with aperiodic sampling: A small-gain theorem based method. In *Proceedings of the 33rd Chinese Control Conference*, pages 1407–1412, July 2014.

[44] Huiyang Liu, Long Cheng, Min Tan, and Zeng-Guang Hou. Containment control of continuous-time linear multi-agent systems with aperiodic sampling. *Automatica*, 57:78–84, jul 2015.

[45] Yuanshi Zheng and Long Wang. Containment control of heterogeneous multi-agent systems. *International Journal of Control*, 87(1):1–8, 2014.

[46] Hamed Haghshenas, Mohammad Ali Badamchizadeh, and Mahdi Baradarannia. Containment control of heterogeneous linear multi-agent systems. *Automatica*, 54:210–216, apr 2015.

[47] Wei Ren and Yongcan Cao. *Distributed coordination of multi-agent networks: emergent problems, models, and issues*. Springer Science & Business Media, 2010.

[48] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*, volume 33. Princeton University Press, 2010.

[49] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009.

[50] Demetri P Spanos, Reza Olfati-Saber, and Richard M Murray. Dynamic consensus on mobile networks. In *IFAC world congress*, pages 1–6. Citeseer, 2005.

[51] Randy A Freeman, Peng Yang, and Kevin M Lynch. Stability and convergence properties of dynamic average consensus estimators. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 338–343. IEEE, 2006.

[52] Solmaz S Kia, Jorge Cortés, and Sonia Martinez. Dynamic average consensus under limited control authority and privacy requirements. *International Journal of Robust and Nonlinear Control*, 25(13):1941–1966, 2015.

[53] He Bai, Randy A Freeman, and Kevin M Lynch. Robust dynamic average consensus of time-varying inputs. In *49th IEEE Conference on Decision and Control (CDC)*, pages 3104–3109. IEEE, 2010.

[54] F. Chen, Y. Cao, and W. Ren. Distributed average tracking of multiple time-varying reference signals with bounded derivatives. *IEEE Transactions on Automatic Control*, 57(12):3169–3174, 2012.

[55] Jemin George, Randy A Freeman, and Kevin M Lynch. Robust dynamic average consensus algorithm for signals with bounded derivatives. In *2017 American Control Conference (ACC)*, pages 352–357. IEEE, 2017.

[56] S. Rahili and W. Ren. Heterogeneous distributed average tracking using nonsmooth algorithms. In *American Control Conference*, pages 691–696, July 2017.

[57] Jorge Cortés. Analysis and design of distributed algorithms for x-consensus. In *Conference on Decision and Control*, pages 3363–3368. IEEE, 2006.

[58] Yilun Shang. Finite-time weighted average consensus and generalized consensus over a subset. *IEEE Access*, 4:2615–2620, 2016.

[59] T. Yucelen and J. D. Peterson. Active-passive networked multiagent systems. In *Conference on Decision and Control*, pages 6939–6944, 2014.

[60] J. D. Peterson, T. Yucelen, G. Chowdhary, and S. Kannan. Exploitation of heterogeneity in distributed sensing: An active-passive networked multiagent systems approach. In *American Control Conference*, pages 4112–4117, 2015.

[61] J. D. Peterson, T. Yucelen, J. Sarangapani, and E. L. Pasiliao. Active-passive dynamic consensus filters with reduced information exchange and time-varying agent roles. *IEEE Transactions on Control Systems Technology*, 28(3):844–856, 2020.

[62] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2):243–255, 2004.

[63] Francesco Bullo, Jorge Cortes, and Sonia Martinez. *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*, volume 27. Princeton University Press, 2009.

[64] Jorge Cortes, Sonia Martinez, and Francesco Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM: Control, Optimisation and Calculus of Variations*, 11(4):691–719, 2005.

[65] Francesco Bullo, Ruggero Carli, and Paolo Frasca. Gossip coverage control for robotic networks: Dynamical systems on the space of partitions. *SIAM Journal on Control and Optimization*, 50(1):419–447, 2012.

[66] Minyi Zhong and Christos G Cassandras. Distributed coverage control in sensor network environments with polygonal obstacles. *IFAC Proceedings Volumes*, 41(2):4162–4167, 2008.

[67] Guiling Wang, Guohong Cao, and Thomas F La Porta. Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5(6):640–652, 2006.

[68] Luciano CA Pimenta, Vijay Kumar, Renato C Mesquita, and Guilherme AS Pereira. Sensing and coverage for a network of heterogeneous robots. In *2008 47th IEEE conference on decision and control*, pages 3947–3952. IEEE, 2008.

[69] Alyssa Pierson, Lucas C Figueiredo, Luciano CA Pimenta, and Mac Schwager. Adapting to performance variations in multi-robot coverage. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 415–420. IEEE, 2015.

[70] Omur Arslan and Daniel E Koditschek. Voronoi-based coverage control of heterogeneous disk-shaped robots. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4259–4266. IEEE, 2016.

[71] María Santos, Yancy Diaz-Mercado, and Magnus Egerstedt. Coverage control for multirobot teams with heterogeneous sensing capabilities. *IEEE Robotics and Automation Letters*, 3(2):919–925, 2018.

[72] Katie Laventall and Jorge Cortés. Coverage control by multi-robot networks with limited-range anisotropic sensory. *International Journal of Control*, 82(6):1113–1121, 2009.

[73] Farsam Farzadpour, Xuebo Zhang, Xiang Chen, and Tong Zhang. On performance measurement for a heterogeneous planar field sensor network. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 166–171. IEEE, 2017.

[74] Azwirman Gusrialdi, Sandra Hirche, Takeshi Hatanaka, and Masayuki Fujita. Voronoi based coverage control with anisotropic sensors. In *2008 American control conference*, pages 736–741. IEEE, 2008.

[75] Mac Schwager, Daniela Rus, and Jean-Jacques Slotine. Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3):357–375, 2009.

[76] Andrea Carron, Marco Todescato, Ruggero Carli, Luca Schenato, and Gianluigi Pillonetto. Multi-agents adaptive estimation and coverage control using gaussian regression. In *2015 European Control Conference (ECC)*, pages 2490–2495. IEEE, 2015.

[77] Farid Sharifi, Mostafa Mirzaei, Youmin Zhang, and Brandon W Gordon. Cooperative multi-vehicle search and coverage problem in an uncertain environment. *Unmanned systems*, 3(01):35–47, 2015.

[78] Minghui Zhu and Sonia Martínez. Discrete-time dynamic average consensus. *Automatica*, 46(2):322 – 329, 2010.

[79] E. Montijano, J. I. Montijano, C. Sagues, and S. Martínez. Step size analysis in discrete-time dynamic average consensus. In *2014 American Control Conference*, pages 5127–5132, June 2014.

[80] Eduardo Montijano, Juan Ignacio Montijano, Carlos Sagüés, and Sonia Martínez. Robust discrete time dynamic average consensus. *Automatica*, 50(12):3131 – 3138, 2014.

[81] F. Lewis, D. Vrabie, and V. Syrmos. *Optimal Control*. Wiley, 2012.

[82] H. G. Tanner, G. J. Pappas, and V. Kumar. Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3):443–455, June 2004.

[83] Wei Ding, Gangfeng Yan, and Zhiyun Lin. Collective motions and formations under pursuit strategies on directed acyclic graphs. *Automatica*, 46(1):174–181, 2010.

[84] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*, volume 33. Princeton University Press, 2010.

[85] Giuseppe Oriolo, Alessandro De Luca, and Marilena Vendittelli. Wmr control via dynamic feedback linearization: Design, implementation, and experimental validation. *Transactions on Control Systems Technology*, 10:835–852, 2002.

[86] Yongduan Song, Yujuan Wang, John Holloway, and Miroslav Krstic. Time-varying feedback for regulation of normal-form nonlinear systems in prescribed finite time. *Automatica*, 83:243–251, 2017.

[87] Tansel Yucelen, Zhen Kan, and Eduardo L Pasiliao. Finite-time cooperative engagement. *IEEE Transactions on Automatic Control*, 2018.

[88] Xiaoyang Liu, Daniel WC Ho, and Chunli Xie. Prespecified-time cluster synchronization of complex networks via a smooth control approach. *IEEE transactions on cybernetics*, 2018.

[89] K. D. Young, V. I. Utkin, and U. Ozguner. A control engineer's guide to sliding mode control. *IEEE Transactions on Control Systems Technology*, 7(3):328–342, May 1999.

[90] M. V. Cook. *Flight Dynamics Principles*. Elsevier, 2007.

[91] Alessandro De Luca, Giuseppe Oriolo, and Marilena Vendittelli. Control of wheeled mobile robots: An experimental overview. In *Ramsete*, pages 181–226. Springer, 2001.

[92] Ram P. Kanwal. *Distributional Derivatives of Functions with Jump Discontinuities*, pages 99–137. Birkhäuser Boston, Boston, MA, 1998.

[93] Hassan K Khalil. *Nonlinear systems*. Prentice-Hall, 2002.

[94] Lixian Zhang and Huijun Gao. Asynchronously switched control of switched linear systems with average dwell time. *Automatica*, 46(5):953–958, 2010.

[95] Joao P Hespanha and A Stephen Morse. Stability of switched systems with average dwell-time. In *Conference on Decision and Control*, volume 3, pages 2655–2660. IEEE, 1999.

[96] Joao P Hespanha. Uniform stability of switched linear systems: Extensions of lasalle's invariance principle. *IEEE Transactions on Automatic Control*, 49(4):470–482, 2004.

[97] Zvi Artstein. Stability, observability and invariance. *Journal of differential equations*, 44(2):224–248, 1982.

[98] Guisheng Zhai, Bo Hu, Kazunori Yasuda, and Anthony N Michel. Qualitative analysis of discrete-time switched systems. In *Proceedings of the 2002 American Control Conference*, volume 3, pages 1880–1885. IEEE, 2002.

[99] José C Geromel and Patrizio Colaneri. Stability and stabilization of discrete time switched systems. *International Journal of Control*, 79(07):719–728, 2006.

[100] Mathias Bürger, Giuseppe Notarstefano, Francesco Bullo, and Frank Allgöwer. A distributed simplex algorithm for degenerate linear programs and multi-agent assignments. *Automatica*, 48(9):2298–2304, 2012.

[101] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE, 2007.

[102] Himesh Bhatia, William Paul, Fady Alajaji, Bahman Gharesifard, and Philippe Burlina. Rényi generative adversarial networks. *arXiv preprint arXiv:2006.02479*, 2020.

[103] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE, 2007.

[104] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[105] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[106] Dimitri P Bertsekas. *Network optimization: continuous and discrete models*. Athena Scientific Belmont, MA, 1998.

[107] Frédéric Bourgault, Tomonari Furukawa, and Hugh F Durrant-Whyte. Optimal search for a lost target in a bayesian world. In *Field and service robotics*, pages 209–222. Springer, 2003.

[108] Hu Xiao, Rongxin Cui, and Demin Xu. A sampling-based bayesian approach for cooperative multiagent online search with resource constraints. *IEEE Transactions on Cybernetics*, 48(6):1773–1785, 2017.

[109] Mohammad Mahdi Azari, Fernando Rosas, Kwang-Cheng Chen, and Sofie Pollin. Optimal uav positioning for terrestrial-aerial communication in presence of fading. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2016.

[110] Zachary T Dydek, Anuradha M Annaswamy, and Eugene Lavretsky. Adaptive control of quadrotor uavs: A design trade study with flight evaluations. *IEEE Transactions on control systems technology*, 21(4):1400–1406, 2012.