October 2019

Distributed Spatiotemporal Control and Dynamic Information Fusion for Multiagent Systems

Dzung Minh Duc Tran
University of South Florida

Distributed Spatiotemporal Control and Dynamic Information Fusion for Multiagent Systems

by

Dzung Minh Duc Tran

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Mechanical Engineering
Department of Mechanical Engineering
College of Engineering
University of South Florida

Major Professor: Tansel Yucelen, Ph.D.
Eduardo L. Pasiliao Jr., Ph.D.
Rajiv Dubey, Ph.D.
Kyle Reed, Ph.D.
Yasin Yilmaz, Ph.D.

Date of Approval:
October 21, 2019

Keywords: Multiplex Networks, Formation Control, Finite-Time Control,
Nullspace Control, Sensor Networks

Dedication

For my beloved parents, Ngoc Tran and Hoa Nguyen, and my brother and sisters, Thinh Tran, Hai Tran and Hang Tran, who have always encouraged me to pursue my dreams.

Acknowledgments

I would like to take this opportunity to express my gratitude to many people who have helped me to make this dissertation possible. First of all, I would like to thank my advisor, Dr. Tansel Yucelen, for his continuous support, motivation, trust, and patience that he gave me. He taught me not only doing research but also many things outside of our research field. His energetic and serious attitude toward doing research has always motivated me to put the full strength in my works. I also gratefully acknowledge the other members of my Ph.D. committee, Dr. Pasiliao, Dr. Rajiv Dubey, Dr. Kyle Reed, and Dr. Yasin Yilmaz, for their time and helpful comments. I would also wish to thank Dr. Rasim Guldiken for his extensive academic guidance. I would also like to thank Dr. Sarangapani Jagannathan from Missouri University of Science and Technology, Dr. David Casbeer from the Air Force Research Laboratory, Wright-Patterson Air Force Base, Ohio, and Dr. Sriram Chellappan and Dr. Andres Tejada from the University of South Florida, for the opportunities to collaborate.

In addition, I would like to acknowledge the support from the University of Missouri Research Board, the U.S. Air Force Summer Faculty and Student Fellowship Program, the Oak Ridge Associated Universities Ralph E. Powe Junior Faculty Enhancement Awards Program, the Dynamic Data-Driven Applications Systems Program of the Air Force Office of Scientific Research under grant number FA9550-17-1-0303, the Army Research Office under grant W911NF-17-1-0582, and the grant from Florida High Tech Corridor.

My sincere thanks also go to my fellow labmates at the University of South Florida and the Missouri University of Science and Technology. I would like to thank Kemberly Cespedes, Merve Dogan, Dr. Ali Albattat, Dr. Ehsan Arabi, Dr. Benjamin Gruenwald, Dr. Ahmet Koru, Dr. Daniel Peterson, Si Dang, Jesse Jaramillo, Stefan Ristevski, Burak Sarsilmaz, Daniel Wagner, Tyler Wieczorek, and Emre Yildirim. I am very grateful for all the personal and professional discussions, and all the fun we had in the last five years.

I also want to thank my “brothers” in the Nam Phong soccer team, who always stay by my side and cheer me up whenever I’m in need. Last but not least, I would like to thank my family: my parents, my brother, and sisters. Without their constant love, support and encouragement, I would not have completed this works and being who I am today.

Table of Contents

List of Figures

Abstract

The first objective of this dissertation is to develop novel distributed control architectures allowing spatiotemporal control of multiagent systems as applied to formation control. In addition, its second objective is to introduce distributed estimation frameworks for dynamic information fusion for addressing the heterogeneity in sensor networks.

Changing the spatial and temporal properties of agent teams in a distributed manner and in real-time is an open problem in the control system literature as multiagent systems are often required to complete tasks with ever-increasing complexity in adverse conditions and dynamic environments. Motivated by this standpoint, this dissertation aims to address challenges related to spatiotemporal control of multiagent systems by proposing three novel tools and methods: The multiplex information networks; the nullspace control; and the time transformation method. First, existing distributed control algorithms utilize only a single layer information exchange rule leading to the multiagent systems have fixed spatial and temporal properties (e.g., the size, orientation and spatial evolution rate of a formation are fixed). To this end, we introduce multiplex information networks with multiple information exchange layers comprising both intralayer and interlayer communication links to allow the spatial and temporal properties of multiagent systems (e.g., the formation's size, orientation, and bandwidth) being manipulated in a distributed manner. Moreover, tools and methods from differential potential fields are used for connectivity maintenance and collision avoidance between agents. Second, complex cooperative behaviors in multiagent systems are restricted under existing control architectures since the local interactions between agents are encoded in the standard Laplacian matrix, which has the nullspace spanning the vector of ones. A novel method proposed in this dissertation defines a more general version for Laplacian matrix, whose nullspace can be manipulated as desired, and reveals a better understanding of the local interactions as well as allows a broader range of complex cooperative behaviors in multiagent systems. Third, time-critical applications, where a task is required to be completed at a user-defined convergence time, is another challenge. Distributed control algorithms for such applications are also developed and generalized. Specifically, a novel time

transformation method is employed to transform the system from the prescribed time interval $t \in [0, T)$ to an equivalent system over the stretched infinite-time interval $s \in [0, \infty)$ for analysis purposes.

One additional challenge in multiagent system is the heterogeneity in sensor networks, which prevents dynamics information correctly being fused. Therefore, another major contribution of this dissertation is to introduce and analyze new distributed input and state estimation architectures for addressing the heterogeneity in sensor networks stemming from different in sensor modalities, quality of sensing information (value of information), and information roles of nodes (active and passive). Both fixed and time-varying information roles of nodes are investigated. Furthermore, some existing literature (see, for example, [2]) implicitly considers nodes with different information roles, yet they require global sufficient stability conditions for analysis while our proposed architectures only utilize local measurements and information both in execution and design stages to guarantee the stability and performance of the overall sensor network.

Finally, the stability of the proposed architectures are theoretically analyzed and their efficacy is illustrated on numerical examples as well as verified with experiments on various mobile robot platforms.

Chapter 1: Introduction

With the rapid development of technology in computing, sensing and communication in the last two decades, exploiting large numbers of interconnected agents such as low-cost and small-in-size autonomous vehicles and micro-sensors has become feasible. This supports a wide array of civilian and military operations ranging from environment monitoring, surveillance to guidance, navigation, and control of autonomous underwater, ground, aerial, and space vehicles. Such operations lead to a demand for developing advanced distributed control architectures for allowing these multiagent systems' spatial and temporal properties to evolve adaptively in adverse conditions and dynamic environments (e.g., applications related to swarms of robots) as well as allowing information fusion (e.g., applications in sensor networks). Motivated by this standpoint, this dissertation first focuses on developing control architectures for manipulating spatiotemporal properties of multiagent systems and then focuses on developing dynamic information fusion architectures for sensor networks.

One of the main challenges in multiagent systems is to control the spatial and temporal properties of agent teams in real-time. To elucidate this point, consider a fleet of ground vehicles as an example that is commanded to form and maintain a formation while simultaneously tracking a dynamic target. The distance between each agent in the formation can be large under ideal conditions to maximize the sensing ability of the overall multiagent system as a whole. Still, when agents pass through a narrow passage, it is then necessary for the formation to scale down (i.e., the spatial property) in real-time to fit in. In addition, depending on the speed of the dynamic target, agents need to adjust their bandwidth (i.e., the temporal property) in real-time for maintaining a desired tracking distance with the target. Yet, current distributed control methods have a lack of information exchange infrastructure and tools to enable such spatial and temporal evolution in a decentralized manner. This is due to the fact that these methods are designed based on information exchange rules for a network having a single layer (see, for example, [1]–[3] and references therein), which leads to multiagent formations with fixed, non-evolving spatial properties. For situations where capable agents (or leaders) have to control the resulting formation using these methods, they can only

do so if such vehicles have global information exchange ability. However, this is not practical for cases with large numbers of agents and low-bandwidth peer-to-peer communications. As another reason for restrictions in increasingly complex multiagent systems' tasks, most distributed control results are predicated on the benchmark consensus algorithm, which is built on the well-known Laplacian matrix whose nullspace spans the vector of ones. In addition, for time-critical applications where the multiagent systems are required to complete a given task over a specific time interval, most standard algorithms either depend on the initial conditions or focus on finding the upper bound for the convergence time. As a result, the convergence time cannot be assigned arbitrarily by the users.

On developing dynamic information fusion architectures for sensor networks, a critical roadblock to achieving correct dynamic information fusion is heterogeneity. Heterogeneity in sensor networks is unavoidable in real-world applications and comes from many sources such as a subset of sensors is subject to the observation while the rest is not, sensing ability and qualities of each sensor can be also different. A dynamic information fusion process should consider such heterogeneities, yet existing distributed estimation algorithms only address these challenges partly. For example, [3] and [2] consider nonidentical modalities of sensor nodes in their distributed algorithms. However, [3] ignores the possibility of having passive nodes in the network since it requires all nodes to be active in the sense of receiving observations from a process of interest. Although the authors of [2] implicitly consider nodes that can have time-invariant active and passive information roles, their analysis involve global sufficient stability conditions, which can be impractical for sensor networks having sufficiently large set of nodes.

In what follows, an overview of the proposed tools and methods as well as the contribution of this dissertation toward addressing the challenges in manipulating spatiotemporal properties of multiagent systems and fusing dynamics information in heterogeneous sensor networks are introduced.

1.1 Multiplex Information Based Distributed Control Architecture for Multiagent Systems

Current distributed control methods have a lack of information exchange infrastructure to enable spatial and temporal evolution of multiagent systems. This is due to the fact that the single-layer structure of existing approaches does not provide the necessary flexibility to control such properties through local interactions (e.g., see [4–9] and references therein). For example, the formation control architectures proposed in [7–9] result in a fixed formation; that is, the size and the orientation of the formation are fixed and they cannot be altered distributively once being formed. As another example, the well-known

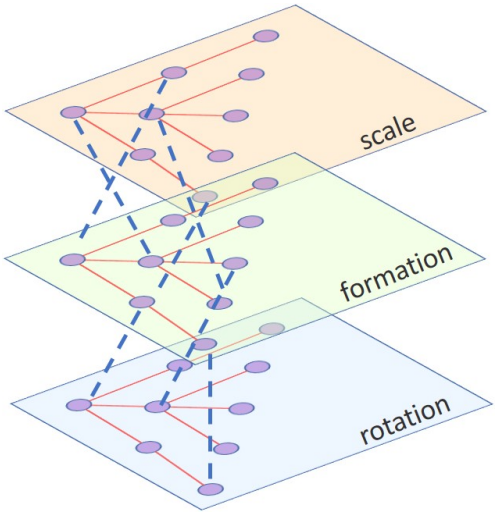


Figure 1.1: An illustration of a multiplex information network with 3 layers.

consensus and consensus-like approaches have a fixed bandwidth, which depends on the Fiedler eigenvalue of the graph Laplacian, and it cannot be changed distributively on the fly.

Multiplex information networks is an appropriate method for altering these fixed properties in real-time. In particular, multiplex information networks describe networks with multiple information exchange layers comprising both intralayer and interlayer communication links. Figure 1.1 illustrates a multiplex information network with 3 layers, where circles denote nodes or agents, solid lines denote intralayer communication links, and dashed lines denote interlayer communication links. The idea of multiplex information networks has emerged in the physics and social science fields (e.g., see [10–14] and references therein). However, these fields mainly focus on studying the system-theoretic characteristics as well as the convergence of the overall network dynamics without focusing on the control design aspect. The contribution of this dissertation is to introduce and system-theoretically utilize multiplex information network for enabling spatial and temporal control of multiagent systems through local interactions. Specifically, the main layer is utilized to regulate the agent team for completing a global task (e.g., forming a desired formation, tracking a dynamics target, and avoiding collision simultaneously) while other layers are utilized to spread out the desired parameters (e.g., the desired scaling factor, rotation angle, or bandwidth commands) from the capable agents (or leaders) to other agents in the network. The key feature of this distributed control architecture is that the parameters updated under secondary layers directly influences the main layer, and hence, the spatial and temporal properties of the multiagent systems can be manipulated in real-time and in a decentralized manner.

1.2 The New Laplacian Matrix with a User-Assigned Nullspace

The common denominator of most notable existing distributed control results is that they utilize the benchmark consensus algorithm, which is built on the well-known Laplacian matrix whose nullspace spans the vector of ones (e.g., see [5, 8, 15–25]). To elucidate this point, consider the benchmark consensus algorithm over undirected and connected graphs with scalar integrator dynamics given by $\dot{x}_i(t) = -\sum_{i\sim j}\big(x_i(t) - x_j(t)\big)$, where $x_i(t)$ denotes the state of agent i, $i = 1,\ldots,N$, and $i \sim j$ indicates that agents i and j are neighbors. Defining $x(t) \triangleq [x_1(t),\ldots,x_n(t)]^{\mathrm{T}}$, one can compactly write the overall dynamics of this multiagent system as $\dot{x}(t) = -\mathcal{L}x(t)$, where $\mathcal{L} \triangleq \mathcal{D} - \mathcal{A}$ is the Laplacian matrix with $\mathcal{D} \in \mathbb{R}^{n\times n}$ denoting its degree matrix and $\mathcal{A} \in \mathbb{R}^{n\times n}$ denoting its adjacency matrix. In particular, the spectrum of the corresponding Laplacian matrix can now be ordered as $0 = \lambda_1(\mathcal{L}) < \lambda_2(\mathcal{L}) \leq \ldots \leq \lambda_n(\mathcal{L})$ ($\lambda_2(\mathcal{L})$ is called as the Fiedler eigenvalue that determines the convergence rate), the nullspace of this Laplacian matrix spans $\mathbf{1}_n = [1,\ldots,1]^{\mathrm{T}}$ ($\mathbf{1}_n$ is the eigenvector corresponding the zero eigenvalue $\lambda_1(\mathcal{L})$), and $\lim_{t\to\infty} x(t) = c\mathbf{1}_n$ with c being a scalar (consensus). Note that the above consensus algorithm is the key building block for a wide array of existing distributed control architectures including but not limited to formation architectures, pinning architectures, containment architectures, and dynamic information fusion architectures. Hence, these extensions are also predicated on this Laplacian matrix with a nullspace spanning the vector of ones. The following question is now immediate: *To pave the way for composing complex cooperative behaviors in multiagent systems, can we generalize the Laplacian nullspace such that it can span any vector with positive elements?*

One contribution of this dissertation is to address the above question, where we introduce a new Laplacian matrix for undirected and connected graphs that generalizes the well-known Laplacian matrix (hereinafter referred to as the standard Laplacian matrix) whose nullspace spans the vector of ones. Specifically, the proposed, new Laplacian matrix is based on a desired, user-assigned nullspace. The key feature of our approach for generating this new Laplacian matrix is that it is simply predicated on keeping the same, standard adjacency matrix and altering the degree matrix. That is, considering a distributed control architecture developed based on the standard Laplacian matrix, one can simply add self-loops to that architecture to achieve convergence to a given user-assigned nullspace based on the results of this dissertation.

1.3 Finite-Time Control with Time Transformation

Many practical applications such as engagement of a guided missile with a target, landing of an aerial vehicle at a non-stationary carrier, and sequential execution of given complex tasks require the tasks to be completed within a specific time interval $[0, \tau)$. While there is a rich literature with regard to finite-time control, the finite-time convergence with the standard algorithms depends on initial conditions of dynamical systems (see, for example, [26–32] and the references therein), and therefore, τ may not be readily assigned by a control designer. To provide a remedy to this problem, several results focus on finding an upper bound on the finite-time convergence (see, for example, [33–39] and references therein). Recently, there are also new results such as [40–54] (and references therein) that have the ability to directly assign a user-defined convergence time τ to the finite-time algorithms utilized in time-critical applications. This dissertation also contributes to finite-time control by proposing a structure predicated on a novel time transformation approach. Concretely, the proposed structure allows the user to assign the convergence time arbitrarily regardless of initial conditions while the time transformation method is utilized to transform a resulting algorithm over the prescribed time interval $[0, \tau)$ to an equivalent algorithm over the stretched infinite-time interval $[0, \infty)$ for stability analysis. With this time transformation technique, standard system-theoretic tools and methods can be used to analyze the systems.

1.4 Information Fusion Structure

There are two common ways to perform distributed dynamic information fusion. Specifically, one classical way includes decentralized data fusion, for example, see [55–57], where these methods have been shown to work well in practice for some applications without formal stability guarantees. Unlike these methods, system-theoretic dynamic information fusion involves equations of motion to describe time behavior of the information fusion process and they also offer stability guarantees (e.g. [58–60]). The contribution of this dissertation builds on system-theoretic dynamic information fusion approaches.

Although distributed estimation algorithms have had strong appeal owing to their reliability and flexibility as outlined above, a critical roadblock to achieving correct dynamic information fusion with these algorithms is heterogeneity. Heterogeneity in sensor networks is unavoidable in real-world applications. To elucidate this fact, consider a target estimation problem as a motivating example. Specifically, nodes of a given sensor network can have heterogeneous information roles in the target estimation problem such that

a subset of nodes can be subject to observations of this target (active nodes) and the rest can be subject to no observation (passive nodes). Thus, only active nodes have to be taken into account during the dynamic information fusion process. In addition, note that nodes of a sensor network can also have nonidentical sensor modalities; for example, a subset of nodes can sense the target position and others can sense the target velocity. This case also needs to be considered in the dynamic information fusion process. Dealing with these classes of heterogeneity in sensor networks to achieve correct and reliable dynamic information fusion is a challenging task using distributed estimation algorithms. Toward this end, notable contributions in the literature include [2, 3, 58–75]. Specifically, the authors of [58–65] propose dynamic consensus algorithms that are suitable for sensor networks with all nodes being active. However, as discussed above, a subset of nodes in a sensor network can be passive in that they may not be able to sense a process of interest and collect information. While the authors of [66–68] present methods that cover specific applications when a subset of nodes are passive (and the remaining nodes are active), their results are in the context of static consensus, and hence, they are not suitable in their presented form for dynamic data-driven applications.

The contribution of this dissertation is to introduce and analyze a new distributed input and state estimation architecture for heterogeneous sensor networks. Specifically, nodes of a given sensor network are allowed to have heterogeneous information roles in the sense that a subset of nodes can be active (that is, subject to observations of a process of interest) and the rest can be passive (that is, subject to no observation). Both fixed and varying active and passive roles of sensor nodes in the network are investigated. In addition, these nodes are allowed to have nonidentical sensor modalities under the common underlying assumption that they have complementary properties distributed over the sensor network to achieve collective observability (see, for example, [3] and [2], and references therein). The key feature of our framework is that it utilizes local information not only during the execution of the proposed distributed input and state estimation architecture but also in its design unlike the results in [2]; that is, global uniform ultimate boundedness of error dynamics is guaranteed once each node satisfies given local stability conditions independent from the graph topology and neighboring information of these nodes.

1.5 Organization

The organization of this dissertation is as follows. Chapter 2 introduces the multiplex information network as applied to control the spatial properties of a multi-vehicle formation with scalar integrator dynamics and Chapter 3 generalizes the result to higher-order dynamics. Chapter 4 addresses the temporal

control problem of multiagent systems through utilizing the multiplex information network for controlling the bandwidth. Chapter 5 proposes the new Laplacian matrix with user-assigned nullspace for allowing complex cooperative behaviors in multiagent systems. Chapter 6 presents a structure to control multiagent networks as systems with finite-time algorithms, time transformation, and separation principle. Chapter 7 generalizes the time transformation method by introducing a new class of scalar, time-varying gain functions to convert a baseline control algorithm into a time-varying one for time-critical applications. Chapter 8 introduces a preliminary structure for dynamic information fusion in heterogeneous sensor networks. Chapter 9 provides some extensions for improving the performance and simplifies the tunning procedure of the structure proposed in Chapter 8. Chapter 10 proposes another dynamic information fusion framework with the integration of local observers, the value of information, and active-passive consensus filter as well as a layer to monitor the validity of the information. Finally, concluding remarks and future research directions are presented in Chapter 11.

Chapter 2: Formation Control with Multiplex Information Networks*

Current distributed control methods have a lack of information exchange infrastructure to enable spatially evolving multiagent formations. Specifically, these methods are designed based on information exchange rules represented by a network having a single layer, where they lead to multiagent formations with fixed, non-evolving spatial properties. For situations where capable agents have to control the resulting formation through these methods, they can often do so if such agents have global information exchange ability. Yet, global information exchange is not practical for cases that have large numbers of agents and low-bandwidth peer-to-peer communications. Motivated from this standpoint, the contribution of this paper is to show how information exchange rules, which are represented by a network having multiple layers (multiplex information networks), can be designed for enabling spatially evolving multiagent formations. In particular, we first consider the formation assignment problem and then the formation tracking problem, and introduce new distributed control architectures that allow capable agents to spatially alter the size and the orientation of the resulting formation without requiring global information exchange ability. In addition, tools and methods from differential potential fields are further utilized in order to generalize the proposed distribute control architecture for the formation tracking problem to allow for connectivity maintenance and collision avoidance needed in real-world applications. Stability of the proposed architectures is theoretically analyzed and their efficacy are illustrated on numerical examples and on multiagent formation experiments.

2.1 Introduction

As advances in VLSI and MEMS technologies have boosted the development of integrated systems that combine mobility, computing, communication, and sensing on a single platform, future civilian and military operations will have the capability to exploit large numbers of interconnected agents such as low-cost and small-in-size autonomous vehicles and microsensors. Such large-scale multiagent systems will

*This chapter is previously published in [76]. Permission is included in Appendix I.

support operations ranging from environment monitoring and military surveillance, to guidance, navigation, and control of autonomous underwater, ground, aerial, and space vehicles. For performing operations with dramatically increasing levels of complexity, multiagent systems require advanced distributed information exchange rules in order to make these systems evolve spatially for adapting dynamic environments and effectively responding to human interventions. Yet, current distributed control methods lack information exchange infrastructures to enable spatially evolving multiagent formations. This is due to the fact that these methods are designed based on information exchange rules for a network having a single layer (see, for example, [5, 77, 78] and references therein), which leads to multiagent formations with fixed, non-evolving spatial properties. For situations where capable agents [1] have to control the resulting formation through these methods, they can often do so if such vehicles have global information exchange ability, but this is not practical for cases with large numbers of agents and low-bandwidth peer-to-peer communications.

2.1.1 Contribution

The contribution of this paper is to introduce and show how information exchange rules, which are represented by a network having multiple layers (multiplex information networks), can be designed for enabling spatially evolving multiagent formations. In particular, after stating necessary mathematical preliminaries in Section 2.2, we first consider the formation assignment problem (i.e., creating a desired formation for the multiagent system in hand) in Section 2.3 and then the formation tracking problem (i.e., formation control while tracking a dynamic, non-stationary target) in Section 2.4, and introduce new distributed control architectures that allow capable agents to spatially alter the size and the orientation of the resulting formation[2] without requiring global information exchange ability. In addition, tools and methods from differential potential fields are further utilized in Section 2.4 in order to generalize the proposed distribute control architecture for the formation tracking problem to allow for connectivity maintenance and collision avoidance needed in real-world applications. Stability of the proposed architectures is theoretically analyzed and their efficacy are illustrated on numerical examples in Sections 2.3 and 2.4 and on multiagent formation experiments in Section 2.5.

[1] Capable agents denote a subset (or at least one) of all agents in a given multiagent system, which have the knowledge of desired parameters used to control resulting formations.

[2] In this paper, spatial size control means to scale the original desired distances between agents through a design parameter only available to capable agents and spatial orientation control means to rotate the original multiagent formation by a rotation matrix constructed with a design parameter only available to these capable agents.

2.1.2 Related Literature

Studies on multiplex information networks have recently emerged in the physics and networks science literatures, where they consider system-theoretic characteristics of network dynamics with multiple layers subject to intralayer and interlayer information exchange [10–14, 79–81] (there also exist studies on multiplex networks that do not consider system-theoretic characteristics; see [82] for a survey). However, these studies mainly consider cases where all layers perform simple consensus algorithms and analyze the convergence of the overall multiagent systems in the presence of not only intralayer but also interlayer information exchange, and hence, they do not deal with controlling spatial properties of multiagent formations. Note that there are also recent studies on networks of networks by the authors of [83–85]. However, these studies deal with large-scale systems formed from smaller factor networks via graph Cartesian products; hence, they are also not related with the contribution of this paper.

Spatial multiagent formation control is considered by the authors of [86–89] using approaches different from multiplex information networks. In particular, the authors of [86–88] assume that some of the formation design parameters are known globally by all agents, and the authors of [89] assume global knowledge of the complete network at the analysis stage. However, as previously discussed, such assumptions may not be practical in the presence of large numbers of agents and low-bandwidth peer-to-peer communications. From a data security point of view, in addition, it should be noted that one may not desire a multiagent system with all agents sharing some global information about an operation of interest. Throughout this paper, we do not make such assumptions in our multiplex information networks-based spatial multiagent formation control approach. Finally, two preliminary conference versions of this paper appeared in [90, 91]. The present paper considerably expands on [90, 91] by providing detailed proofs of all the results with additional motivation, examples, and multiagent formation experiments.

2.2 Mathematical Preliminaries

We now introduce this paper's notation and recall basic notions from graph theory, which are followed by the general setup of consensus and formation problems for multiagent systems that are necessary to establish our main results[3].

[3]For details about graph theory and multiagent systems, see [5, 77, 92].

2.2.1 Notation and Notions from Graph Theory

Throughout this paper, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ denotes the set of $n \times 1$ real column vectors, $\mathbb{R}^{n\times m}$ denotes the set of $n \times m$ real matrices, $\mathbb{R}_+$ (resp., $\overline{\mathbb{R}}_+$) denotes the set of positive (resp., non-negative) real numbers, $\mathbb{R}_+^{n\times n}$ (resp., $\overline{\mathbb{R}}_+^{n\times n}$) denotes the set of $n \times n$ positive-definite (resp., non-negative-definite) real matrices, 0_n denotes the $n \times 1$ vector of all zeros, $\mathbf{1}_n$ denotes the $n \times 1$ vector of all ones, $0_{n\times n}$ denotes the $n \times n$ zero matrix, I_n denotes the $n \times n$ identity matrix, and $\otimes$ denotes the Kronecker product operation. In addition, we write $(\cdot)^{\mathrm{T}}$ for transpose, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ for the minimum and maximum eigenvalue of the Hermitian matrix A, respectively; $\lambda_i(A)$ for the i-th eigenvalue of A, where A is symmetric and the eigenvalues are ordered from least to greatest value, $\det(A)$ for the determinant of A, $\mathrm{diag}(a)$ for the diagonal matrix with the vector a on its diagonal, $[x]_i$ for the entry of the vector x on the i-th row, and $[A]_{ij}$ for the entry of the of the matrix A on the i-th row and j-th column. Furthermore, for given functions $f(t)$ and $g(t)$, $f(t) \to g(t)$ as $t \to \infty$ denotes $\lim_{t\to\infty}\big(f(t)-g(t)\big)=0$.

In the multiagent systems literature, graphs are broadly adopted to encode interactions between networked agents. An undirected graph $\mathcal{G}$ is defined by a set $\mathcal{V}_\mathcal{G} = \{1,\dots,n\}$ of nodes and a set $\mathcal{E}_\mathcal{G} \subset \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ of edges. If the distance between two arbitrary nodes is less than R, then they are said to be neighbors and the neighboring relation is denoted by $j \in \mathcal{N}_i \triangleq \{j|\ j \in \mathcal{V}_\mathcal{G}, \|x_{ij}\|_2 < R\}$, where $x_{ij} \triangleq x_i - x_j$ with x_i and x_j being the state (position) of nodes i and j, respectively. In addition, if $(i,j) \in \mathcal{E}_\mathcal{G}$, then the nodes i and j are said to be formation neighbors [93, 94] and this relation is denoted by $j \in \mathcal{N}_i^f$, where $\mathcal{N}_i^f$ is a subset of $\mathcal{N}_i$. In general, note that $\mathcal{N}_i$ can be a time-varying set while $\mathcal{N}_i^f$ is a static set, that is, $\mathcal{N}_i^f$ remains unchanged in the presence of node movements. The degree of a node is given by the number of its formation neighbors. In particular, letting d_i be the degree of node i, the degree matrix of a graph $\mathcal{G}$, $\mathcal{D}(\mathcal{G}) \in \mathbb{R}^{n\times n}$, is given by $\mathcal{D}(\mathcal{G}) = \mathrm{diag}(d)$, $d = [d_1,\dots,d_n]^{\mathrm{T}}$. A path $i_0 i_1 \dots i_L$ is a finite sequence of nodes such that $i_{k-1} \in \mathcal{N}_{i_k}^f$ with $k = 1,\dots,L$, and a graph $\mathcal{G}$ is connected if there exists a path between any pair of distinct nodes. The adjacency matrix of a graph $\mathcal{G}$, $\mathcal{A}(\mathcal{G}) \in \mathbb{R}^{n\times n}$, is given by $[\mathcal{A}(\mathcal{G})]_{ij} = 1$ if $(i,j) \in \mathcal{E}_\mathcal{G}$ and $[\mathcal{A}(\mathcal{G})]_{ij} = 0$ otherwise. The Laplacian matrix of a graph, $\mathcal{L}(\mathcal{G}) \in \overline{\mathbb{R}}_+^{n\times n}$, is given by $\mathcal{L}(\mathcal{G}) \triangleq \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$, where the spectrum of the Laplacian for an undirected and connected graph $\mathcal{G}$ can be ordered as $0 = \lambda_1(\mathcal{L}(\mathcal{G})) < \lambda_2(\mathcal{L}(\mathcal{G})) \le \cdots \le \lambda_n(\mathcal{L}(\mathcal{G}))$ with $\mathbf{1}_n$ as the eigenvector corresponding to the zero eigenvalue $\lambda_1(\mathcal{L}(\mathcal{G}))$ and $\mathcal{L}(\mathcal{G})\mathbf{1}_n = 0_n$ and $\mathrm{e}^{\mathcal{L}(\mathcal{G})}\mathbf{1}_n = \mathbf{1}_n$ hold. Here, we assume graph $\mathcal{G}$ is undirected and connected unless noted otherwise.

2.2.2 Consensus and Formation Dynamics

A graph $\mathcal{G}$ can model a given multiagent system with nodes and edges respectively representing agents and interagent information exchange links. Specifically, let $x_i(t) \in \mathbb{R}^m$ denote the state of node i, whose dynamics is described by the single integrator $\dot{x}_i(t) = u_i(t)$, $x_i(0) = x_{i0}$, $i = 1, \cdots, n$, with $u_i(t) \in \mathbb{R}^m$ being the control input of node i. Allowing agent i to have access to the relative state information with respect to its formation neighbors, a solution to the consensus problem can be achieved, for example, by applying $u_i(t) = -\sum_{j \in \mathcal{N}_i^f} \big(x_i(t) - x_j(t)\big)$ to the above single integrator dynamics [5, 77], where the resulting dynamics can be represented by the Laplacian dynamics of the form

$$\dot{x}(t) = -\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_m \, x(t), \quad x(0) = x_0, \tag{2.1}$$

with $x(t) = [x_1^{\mathrm{T}}(t), \cdots, x_n^{\mathrm{T}}(t)]^{\mathrm{T}}$ denoting aggregated state vector of multiagent system. Since the graph $\mathcal{G}$ is undirected and connected, $\lim_{t \to \infty} [x_i(t)]_j = \big([x_1(0)]_j + \cdots [x_n(0)]_j\big)/n$ holds from (2.1) for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. In this paper, we assume that $m = 2$ without loss of generality, which implies that the multiagent system evolves in a planar space.

On the formation problem, define $x_i(t) - \xi_i \in \mathbb{R}^2$ as the displacement of $x_i(t) \in \mathbb{R}^2$ from the desired formation position of agent i, $\xi_i \in \mathbb{R}^2$. Using now the transformed state $x_i(t) - \xi_i$ instead of $x_i(t)$ in (2.1) for $i = 1, \ldots, n$, one can write the dynamics $\dot{x}(t) = -\big(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_2\big) \, x(t) + \big(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_2\big) \, \xi$, $x(0) = x_0$, [5, 77], where $\xi = [\xi_1, \cdots, \xi_n]^{\mathrm{T}}$. Note that the above expression addressing the formation problem with $m = 2$ can equivalently be written as $\dot{x}_i(t) = -\sum_{j \in \mathcal{N}_i^f} \big(x_i(t) - x_j(t)\big) + \sum_{j \in \mathcal{N}_i^f} \big(\xi_i - \xi_j\big)$, $x_i(0) = x_{i0}$. In the rest of this paper, we consider a generalized version of this benchmark formation problem that not only allows to create a desired formation for the multiagent system in hand (i.e., formation assignment; see Section 2.3) but also allows formation control while tracking a dynamic, non-stationary target (i.e., formation tracking problem; see Section 2.4). In our proposed algorithm, the original resulting formation as well as the desired position of agent i represented by ξ_i are oriented through the rotation matrix $R(\theta_i(t))$ and the size is controlled by the term $\gamma_i(t)$; thus, the resulting formation is now represented by $\gamma_i(t) R(\theta_i(t)) \xi_i$ for $i = 1, \ldots, n$. In addition, the desired scaling design parameter $\gamma(t)$ and the desired rotation angle design parameter $\theta(t)$ are both locally spread out in the network via two separate layers, and $\gamma_i(t)$ and $\theta_i(t)$ asymptotically converge to these design parameter values. This then allows for spatial control of both the size and orientation of a given original multiagent formation. Although we consider this particular formation problem in this paper,

the presented multiplex information networks-based approach can be extended to many other approaches to formation control. Finally, for the purpose of directly focusing on our main contribution stated in Section 2.1.1, we assume in this paper that the interactions between agents are not subject to time-delays. For practical applications when interaction time-delays are not negligible and significant, one can consider the results in, for example, [95–100] for analytically extending the results of this paper to the time-delay case.

2.3 Spatial Control of Multiagent Systems in Formation Assignment

This section focuses on the formation assignment problem, where we introduce and analyze a multiplex information networks-based distributed control architecture for spatially controlling both size and orientation of multiagent formations (Section 2.3.1). Then, we illustrate the result by a numerical example (Section 2.3.2).

2.3.1 Formation Density and Orientation Control

Consider a system of n agents exchanging information among each other using their local measurements, according to an undirected and connected graph $\mathcal{G}$. Based on the benchmark formation problem outlined in Section 2.2.2, we also consider that ξ_i and ξ_j are locally available to each agent, where this captures an original, desired planar (i.e., $m=2$) formation. In addition, we consider that there is a subset of agents (or at least one agent), i.e., capable agents, that has the knowledge of the desired scaling parameter $\gamma(t)$ and the desired rotation angle $\theta(t)$. To this end, we focus on the problem of developing local information exchange rules for enabling spatial control of size and orientation of the original planar formation through parameters $\gamma(t)$ and $\theta(t)$ available only to a subset of agents (i.e., capable agents). Motivated from this standpoint, we propose the distributed controller having three layers[4]

$$\dot{x}_i(t) = -\sum_{j\in\mathcal{N}_i^f}\big(x_i(t)-x_j(t)\big)+\sum_{j\in\mathcal{N}_i^f}\Big(\gamma_i(t)R(\theta_i(t))\xi_i-\gamma_j(t)R(\theta_j(t))\xi_j\Big)+\dot{\gamma}_i(t)R(\theta_i(t))\xi_i$$
$$+\gamma_i(t)\dot{R}(\theta_i(t))\xi_i,\quad x_i(0)=x_{i0},\qquad (2.2)$$

$$\dot{\gamma}_i(t) = -\sum_{j\in\mathcal{N}_i^f}\big(\gamma_i(t)-\gamma_j(t)\big)-k_i\big(\gamma_i(t)-\gamma(t)\big)-\tau_\gamma\mathrm{sgn}\Big(\sum_{j\in\mathcal{N}_i^f}\big(\gamma_i(t)-\gamma_j(t)\big)+k_i\big(\gamma_i(t)-\gamma(t)\big)\Big),$$
$$\gamma_i(0)=\gamma_{i0},\qquad (2.3)$$

[4]The right hand side of (2.2) coupled with (2.3) and (2.4) represents the local controller $u_i(t)$ for agent dynamics of the form $\dot{x}_i(t)=u_i(t)$.

$$\dot{\theta}_i(t) = -\sum_{j\in\mathcal{N}_i^f}\big(\theta_i(t)-\theta_j(t)\big) - k_i\big(\theta_i(t)-\theta(t)\big) - \tau_\theta \mathrm{sgn}\Big(\sum_{j\in\mathcal{N}_i^f}\big(\theta_i(t)-\theta_j(t)\big) + k_i\big(\theta_i(t)-\theta(t)\big)\Big),$$
$$\theta_i(0) = \theta_{i0}, \qquad (2.4)$$

where $x_i(t) \in \mathbb{R}^2$ denotes the state of the first layer of agent i that corresponds to the actual state of agent i, $\xi_i \in \mathbb{R}^2$ denotes the original formation shape of agent i, $\gamma_i(t) \in \mathbb{R}$ denotes the state of the second layer of agent i that is introduced to distribute the formation scaling parameter or size factor $\gamma(t) \in \mathbb{R}$ through local information exchange, $\theta_i(t) \in \mathbb{R}$ denotes the state of the third layer of agent i that is introduced to distribute the formation orientation parameter or rotation angle $\theta(t) \in \mathbb{R}$ through local information exchange, and $k_i = 1$ for capable agents (a subset or at least one of the n agents in the multiagent system) and otherwise $k_i = 0$. In (2.2), $R(\theta_i(t))$ denotes the rotation matrix of agent i

$$R(\theta_i(t)) \triangleq \begin{bmatrix} \cos\theta_i(t) & -\sin\theta_i(t) \\ \sin\theta_i(t) & \cos\theta_i(t) \end{bmatrix} \in \mathbb{R}^{2\times 2}. \qquad (2.5)$$

Note that the desired formation scaling factor $\gamma(t)$ and rotation angle $\theta(t)$ are considered to be bounded and continuously differentiable, and only available to capable agents as such they have the capability to alter the size and orientation of the resulting formation (i.e., scale and rotate the formation). Similar to [101], we also assume $\dot{\gamma}(t)$ and $\dot{\theta}(t)$ are bounded such that $|\dot{\gamma}(t)| \leq \omega_\gamma$ and $|\dot{\theta}(t)| \leq \omega_\theta$, and τ_γ and τ_θ are chosen such that $\tau_\gamma > \omega_\gamma$ and $\tau_\theta > \omega_\theta$.

It should be also further emphasized that the first layer represented by (2.2) helps in forming the desired formation while the second and third layers represented by (2.3) and (2.4) respectively allow the scaling factor $\gamma(t)$ and the rotation angle $\theta(t)$ to be spread out in the network and be updated in the first layer; hence, the formation size and orientation can be controlled. The next theorem presents our first result.

Theorem 2.3.1. *Consider the networked multiagent system given by (2.2), (2.3), and (2.4), where agents exchange information using local measurements and with $\mathcal{G}$ defining an undirected and connected graph topology. Then,*

$$\lim_{t\to\infty}\Big(\big(x_i(t)-x_j(t)\big) - \gamma(t)R\big(\theta(t)\big)\big(\xi_i-\xi_j\big)\Big) \;=\; 0, \qquad (2.6)$$

holds for all $i = 1,\ldots,n$ and $j \in N_i^f$.

Proof. First, we prove that under the two underlying layers (2.3) and (2.4), $\gamma_i(t)$ and $\theta_i(t)$ converge to desired parameters $\gamma(t)$ and $\theta(t)$ for all $i = 1,\dots,n$. Let us consider the state transformations given by $\tilde{\gamma}_i(t) \triangleq \gamma_i(t) - \gamma(t)$, $i = 1,\dots,n$, and $\tilde{\theta}_i(t) \triangleq \theta_i(t) - \theta(t)$, $i = 1,\dots,n$. Using the first state transformation with (2.3) yields

$$\dot{\tilde{\gamma}}_i(t) = -\sum_{j\in\mathcal{N}_i^f}\big(\tilde{\gamma}_i(t)-\tilde{\gamma}_j(t)\big) - k_i\tilde{\gamma}_i(t) - \tau_\gamma \mathrm{sgn}\Big(\sum_{j\in\mathcal{N}_i^f}\big(\tilde{\gamma}_i(t)-\tilde{\gamma}_j(t)\big) + k_i\tilde{\gamma}_i(t)\Big) - \dot{\gamma}(t), \tag{2.7}$$

and using the second state transformation with (2.4) yields

$$\dot{\tilde{\theta}}_i(t) = -\sum_{j\in\mathcal{N}_i^f}\big(\tilde{\theta}_i(t)-\tilde{\theta}_j(t)\big) - k_i\tilde{\theta}_i(t) - \tau_\theta \mathrm{sgn}\Big(\sum_{j\in\mathcal{N}_i^f}\big(\tilde{\theta}_i(t)-\tilde{\theta}_j(t)\big) + k_i\tilde{\theta}_i(t)\Big) - \dot{\theta}(t). \tag{2.8}$$

By letting $\tilde{\gamma}(t) \triangleq \big[\tilde{\gamma}_1(t),\dots,\tilde{\gamma}_n(t)\big]^{\mathrm{T}}$, $\tilde{\theta}(t) \triangleq \big[\tilde{\theta}_1(t),\dots,\tilde{\theta}_n(t)\big]^{\mathrm{T}}$, the expressions (2.7) and (2.8) can be equivalently written in a compact form as

$$\begin{aligned}
\dot{\tilde{\gamma}}(t) &= -\big(\mathcal{L}(\mathcal{G})+K\big)\tilde{\gamma}(t) - \tau_\gamma \mathrm{sgn}\Big(\big(\mathcal{L}(\mathcal{G})+K\big)\tilde{\gamma}(t)\Big) - \mathbf{1}_n\dot{\gamma}(t), &(2.9)\\
\dot{\tilde{\theta}}(t) &= -\big(\mathcal{L}(\mathcal{G})+K\big)\tilde{\theta}(t) - \tau_\theta \mathrm{sgn}\Big(\big(\mathcal{L}(\mathcal{G})+K\big)\tilde{\theta}(t)\Big) - \mathbf{1}_n\dot{\theta}(t), &(2.10)
\end{aligned}$$

where $K \triangleq \mathrm{diag}\big([k_1,\dots,k_n]^{\mathrm{T}}\big)$. Since $\dot{\tilde{\gamma}}$ and $\dot{\tilde{\theta}}$ have the same structure, we only show the analysis for $\tilde{\gamma}(t)$ here, but the analysis for $\tilde{\theta}(t)$ is similar. Now, consider the Lyapunov function candidate $V(\tilde{\gamma}) = \frac{1}{2}\tilde{\gamma}^T\big(\mathcal{L}(\mathcal{G})+K\big)\tilde{\gamma}$, and so its time derivative along the trajectory of (2.9) is given by

$$\begin{aligned}
\dot{V}(\tilde{\gamma}(t)) &= \tilde{\gamma}^T\big(\mathcal{L}(\mathcal{G})+K\big)\Big(-\big(\mathcal{L}(\mathcal{G})+K\big)\tilde{\gamma}(t) - \tau_\gamma \mathrm{sgn}\Big[\big(\mathcal{L}(\mathcal{G})+K\big)\tilde{\gamma}(t)\Big] - \mathbf{1}_n\dot{\gamma}(t)\Big)\\
&\le -\tilde{\gamma}^T\big(\mathcal{L}(\mathcal{G})+K\big)^2\tilde{\gamma}(t) - \tau_\gamma\|\big(\mathcal{L}(\mathcal{G})+K\big)\tilde{\gamma}(t)\|_1 + |\dot{\gamma}(t)|\|\big(\mathcal{L}(\mathcal{G})+K\big)\tilde{\gamma}(t)\|_1\\
&\le -\tilde{\gamma}^T\big(\mathcal{L}(\mathcal{G})+K\big)^2\tilde{\gamma}(t) - (\tau_\gamma-\omega_\gamma)\|\big(\mathcal{L}(\mathcal{G})+K\big)\tilde{\gamma}(t)\|_1. &(2.11)
\end{aligned}$$

Since $\mathcal{L}(\mathcal{G})+K \in \mathbb{R}_+^{n\times n}$ (Lemma 2 of [102]) and $(\tau_\gamma-\omega_\gamma) > 0$, $\dot{V}(\tilde{\gamma}(t))$ is negative definite. As a result, from Theorem 3.1 of [103], $\lim_{t\to\infty}\tilde{\gamma}_i(t) = 0$, and with the same analysis, $\lim_{t\to\infty}\tilde{\theta}_i(t) = 0$ hold for all $i = 1,\dots,n$. This implies that $\gamma_i(t) \to \gamma(t)$ and $\theta_i(t) \to \theta(t)$ as $t \to \infty$ for all $i = 1,\dots,n$. Hence, it now readily follows from the limit properties along with the squeeze theorem [104] that $R(\theta_i(t)) \to R(\theta(t))$ as $t \to \infty$, where this further implies that $\gamma_i(t)R(\theta_i(t)) \to \gamma(t)R(\theta(t))$ as $t \to \infty$ for all $i = 1,\dots,n$.

Next, we prove that under the main layer (2.2), agents reach consensus, and (2.6) will be achieved. Consider the state transformation given by $\tilde{x}_i(t) \triangleq x_i(t) - \gamma_i(t)R(\theta_i(t))\xi_i$, $i = 1,\ldots,n$. Using this state transformation with (2.2) yields

$$\dot{\tilde{x}}_i(t) \;=\; -\sum_{j\in\mathcal{N}_i^f}\big(\tilde{x}_i(t)-\tilde{x}_j(t)\big). \tag{2.12}$$

By letting $\tilde{x}(t) \triangleq \big[\tilde{x}_1(t),\ldots,\tilde{x}_n(t)\big]^{\mathrm{T}}$, (2.12) can be written in a compact form as

$$\dot{\tilde{x}}(t) = -\big(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_2\big)\tilde{x}(t), \tag{2.13}$$

and therefore, $\lim_{t\to\infty}[\tilde{x}_i(t)]_k = \big([\tilde{x}_1(0)]_k + \cdots [\tilde{x}_n(0)]_k\big)/n$ holds for $i = 1,\ldots,n$ and $k = 1,2$. As a consequence, $x_i(t) - x_j(t) \to \gamma_i(t)R(\theta_i(t))\xi_i - \gamma_j(t)R(\theta_j(t))\xi_j$ as $t\to\infty$ for all $i = 1,\ldots,n$ and $j \in N_i^f$. Finally, from $x_i(t) - x_j(t) \to \gamma_i(t)R(\theta_i(t))\xi_i - \gamma_j(t)R(\theta_j(t))\xi_j$ as $t\to\infty$ and $\gamma_i(t)R(\theta_i(t)) \to \gamma(t)R(\theta(t))$ as $t\to\infty$, one can conclude that $x_i(t) - x_j(t) \to \gamma(t)R\big(\theta(t)\big)\big(\xi_i - \xi_j\big)$ as $t\to\infty$ using the limit properties, where the result is now immediate. ■

Remark 2.3.1. *Theorem 2.3.1 shows that the proposed algorithm given by (2.2), (2.3) and (2.4) allows size and orientation of the multiagent formation to be controlled by formation size parameter $\gamma(t)$ and orientation parameter $\theta(t)$, which are only available to capable agents (not globally).*

Remark 2.3.2. *The dynamical structure of the two underlying layers (2.3) and (2.4) uses the signum functions in order to achieve asymptotic stability in the presence of time-varying signals $\gamma(t)$ and $\theta(t)$, where such functions are also adopted in the networked multiagent systems literature (see, for example, [101, 105]). Note that if $\gamma(t)$ and $\theta(t)$ are constants, then the results of Theorem 2.3.1 still hold without the need for the signum function in (2.3) and (2.4); that is,*

$$\dot{\gamma}_i(t) \;=\; -\sum_{j\in\mathcal{N}_i^f}\big(\gamma_i(t)-\gamma_j(t)\big) - k_i\big(\gamma_i(t)-\gamma(t)\big), \quad \gamma_i(0)=\gamma_{i0}, \tag{2.14}$$

$$\dot{\theta}_i(t) \;=\; -\sum_{j\in\mathcal{N}_i^f}\big(\theta_i(t)-\theta_j(t)\big) - k_i\big(\theta_i(t)-\theta(t)\big), \quad \theta_i(0)=\theta_{i0}. \tag{2.15}$$

Remark 2.3.3. *For improving the rate of convergence of the networked multiagent system, without loss of generality, we can introduce a positive parameter α to the main layer (2.2) as*

$$\dot{x}_i(t) = \alpha\Big[-\sum_{j\in\mathcal{N}_i^f}\big(x_i(t)-x_j(t)\big)+\sum_{j\in\mathcal{N}_i^f}\Big(\gamma_i(t)R(\theta_i(t))\xi_i-\gamma_j(t)R(\theta_j(t))\xi_j\Big)\Big]+\dot{\gamma}_i(t)R(\theta_i(t))\xi_i$$
$$+\gamma_i(t)\dot{R}(\theta_i(t))\xi_i,\quad x_i(0)=x_{i0},\quad (2.16)$$

Remark 2.3.4. *The proposed algorithm in Theorem 2.3.1 can be extended to a three dimensional case with $x_i(t)\in\mathbb{R}^3$. In this case, the rotation matrix becomes*

$$R\big(\theta_i^x(t),\theta_i^y(t),\theta_i^z(t)\big)=R_x\big(\theta_i^x(t)\big)R_y\big(\theta_i^y(t)\big)R_z\big(\theta_i^z(t)\big),\quad (2.17)$$

where $\theta_i^x(t)\in\mathbb{R}$, $\theta_i^y(t)\in\mathbb{R}$, and $\theta_i^z(t)\in\mathbb{R}$ are the rotation angles corresponding to yaw, pitch, and roll, respectively. Also, instead of using only one layer for $\theta_i(t)$ like in 2D case, we now need three layers for $\theta_i^x(t)$, $\theta_i^y(t)$, and $\theta_i^z(t)$.

Remark 2.3.5. *The proposed multiplex networks-based spatial formation control algorithm given by (2.2), (2.3) and (2.4) can be also extended to the case where the graph $\mathcal{G}$ is directed under the assumption that there exists at least one capable agent at the root of the spanning tree [77].*

Remark 2.3.6. *The communication graph topologies for (2.2), (2.3), and (2.4) can be different as long as they are undirected and connected graphs.*

2.3.2 Illustrative Numerical Example

We now consider a group of 9 agents with agent 1 being a capable agent and assign random initial conditions to agents. For the invariant formation problem described earlier, we choose ξ_i of each agent to obtain a formation depicted in Figure 2.1 (in this figure, solid lines represent an undirected information exchange between agents). To control both the size and the orientation of the multiagent formation depicted in Figure 2.1, we use the algorithm given by (2.2), (2.3) and (2.4) with $(\gamma,\theta)=(0.8,-\pi/2)$, $(\gamma,\theta)=(1.0,\pi/3)$, $(\gamma,\theta)=(2.0,\pi/6)$ in Figures 2.2(a), 2.2(b), and 2.2(c), respectively. As expected from the results discussed in Section 2.3.1, the resulting formation in these figures have different sizes and orientations controlled locally through the formation size and orientation parameters (γ,θ) available to the capable agent.

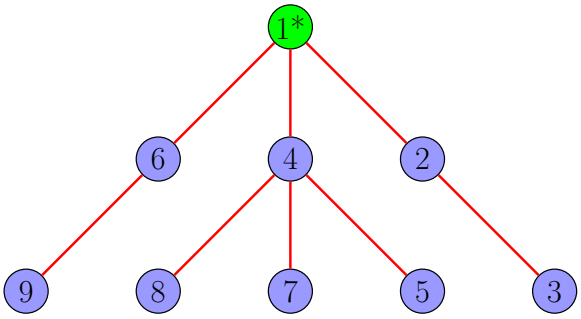
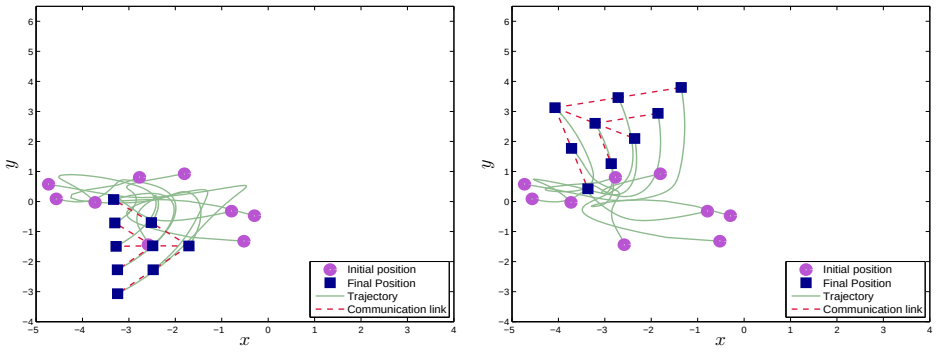
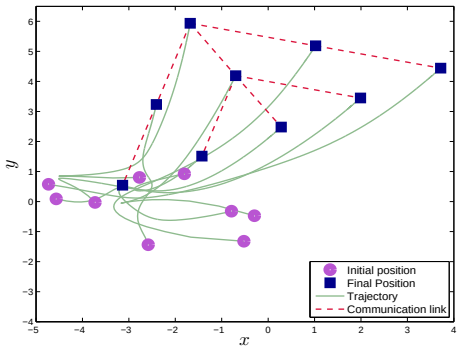


Figure 2.1: A given desired formation for the example in Section 2.3.2



(a) $\gamma = 0.8$, $\theta = -\pi/2$ (b) $\gamma = 1.0$, $\theta = \pi/3$



(c) $\gamma = 2.0$, $\theta = \pi/6$

Figure 2.2: Formation size and orientation control using the results in Theorem 2.3.1 for different (γ, θ) pairs.

2.4 Spatially Evolving Multiagent Formation Tracking

In this section, we generalize the results from the previous section to formation tracking problem in order to control the size and orientation of the formation while tracking a dynamic target (Section 2.4.1). The algorithms are then further extended to allow connectivity maintenance and collision avoidance (Section 2.4.2). A numerical example is presented to illustrate the efficacy of the methods (Section 2.4.3).

2.4.1 Multiagent Formation Tracking through Multiplex Information Networks

Consider a system of n agents exchanging information among each other using their local measurements according to a connected, undirected graph $\mathcal{G}$. Specifically, we propose a distributed control architecture using networks having multiple layers with the main (physical) network layer given by

$$\dot{x}_i(t) = -\sum_{j\in\mathcal{N}_i^f}\Big(\big(x_i(t)-p_i(t)-c_i(t)\big)-\big(x_j(t)-p_j(t)-c_j(t)\big)\Big)-k_i\big(x_i(t)-p_i(t)-c_i(t)\big) +\dot{p}_i(t)+\dot{c}_i(t), \quad x_i(0)=x_{i0}, \tag{2.18}$$

where $x_i(t)\in\mathbb{R}^2$ denotes the state (i.e., physical position) of agent i, and $c_i(t)\triangleq[c_i^x(t),c_i^y(t)]^{\mathrm{T}}\in\mathbb{R}^2$ and

$$p_i(t)\triangleq R\big(\theta_i(t)\big)S\big(\gamma_i^x(t),\gamma_i^y(t)\big)\xi_i\in\mathbb{R}^2, \tag{2.19}$$

correspond to the signals locally obtained through other network layers described in the next paragraph. In (2.18), $k_i=1$ only for capable agents and it is zero otherwise. Note that we implicitly assume that there exists at least one capable agent in the multiagent system. In (2.19), $\xi_i\in\mathbb{R}^2$ denotes the original formation shape of agent i in the sense discussed in Section 2.2.2, $\theta_i(t)\in\mathbb{R}$ is the rotation angle of agent i that is used in its local rotation matrix given by (2.5), and $\gamma_i^x(t)\in\mathbb{R}$ and $\gamma_i^y(t)\in\mathbb{R}$ are scaling factors of agent i in x and y dimensions of the planar space, respectively, that are used in its local scaling matrix given by

$$S\big(\gamma_i^x(t),\gamma_i^y(t)\big)\triangleq\mathrm{diag}\big([\gamma_i^x(t),\gamma_i^y(t)]^T\big)\in\mathbb{R}^{2\times2}. \tag{2.20}$$

To define the dynamical structure of other network layers, let $\phi_i(t)$ denotes either $c_i^x(t)\in\mathbb{R}$, $c_i^y(t)\in\mathbb{R}$, $\theta_i(t)\in\mathbb{R}$, $\gamma_i^x(t)\in\mathbb{R}$, or $\gamma_i^y(t)\in\mathbb{R}$ for conciseness of the following discussion that satisfy

$$\dot{\phi}_i(t) = -q_i(t)-\tau\mathrm{sgn}\big(q_i(t)\big),\ \phi_i(0)=\phi_{i0}, \tag{2.21}$$

$$q_i(t) \triangleq \sum_{j\in\mathcal{N}_i^f}\big(\phi_i(t)-\phi_j(t)\big)+k_i\big(\phi_i(t)-\phi_0(t)\big), \tag{2.22}$$

where $\tau\in\mathbb{R}$ is a positive design parameter and it is assumed that $\phi_0(t)$ and $\dot{\phi}_0(t)$ are bounded. Note that in (2.21) and (2.22), $\phi_0(t)$ denotes either $c^x(t)\in\mathbb{R}$, $c^y(t)\in\mathbb{R}$, $\theta_0(t)\in\mathbb{R}$, $\gamma_0^x(t)\in\mathbb{R}$, or $\gamma_0^y(t)\in\mathbb{R}$, where $c(t)\triangleq[c^x(t),c^y(t)]^{\mathrm{T}}$ is the position of the dynamic target on a planar space, $\theta_0(t)$ is the desired rotation

angle, and $\gamma_0^x(t)$ and $\gamma_0^y(t)$ are desired scaling factors, respectively. Since $k_i = 1$ only for capable agents, $c(t)$, $\theta_0(t)$, $\gamma_0^x(t)$, and $\gamma_0^y(t)$ are only available to these capable agents.

Since $\dot{\phi}_0(t)$ is bounded, this implies that $|\dot{c}^x(t)| \leq \omega_{c^x}$, $|\dot{c}^y(t)| \leq \omega_{c^y}$, $|\dot{\theta}_0(t)| \leq \omega_{\theta_0}$, $|\dot{\gamma}_0^x(t)| \leq \omega_{\gamma_0^x}$, and $|\dot{\gamma}_0^y(t)| \leq \omega_{\gamma_0^y}$. In what follows, we let ω to be the largest constant among ω_{c^x}, ω_{c^y}, ω_{θ_0}, $\omega_{\gamma_0^x}$, and $\omega_{\gamma_0^y}$ without loss of generality (i.e., $|\dot{\phi}_0(t)| \leq \omega$), and set $\tau > \omega$. The next theorem presenting the second result of this paper shows that the multiplex information networks-based distributed controller architecture given by (2.18) and (2.21) not only allows agents to track a dynamic target but also allows them to alter size and orientation of the resulting formation.

Theorem 2.4.1. *Consider the networked multiagent system given by (2.18) and (2.21), where agents exchange their local measurements using an undirected and connected graph $\mathcal{G}$. Then, the expression given by*

$$\lim_{t\to\infty} \Big(x_i(t) - \rho_i(t) \Big) \;=\; 0, \tag{2.23}$$

holds for all $i = 1,\ldots,n$, where $\rho_i(t) \triangleq c(t) + R\big(\theta_0(t)\big) S\big(\gamma_0^x(t), \gamma_0^y(t)\big) \xi_i$.

Proof. We first show that $\phi_i(t)$ converges to $\phi_0(t)$ for all cases when $\phi_i(t)$ denotes either $c_i(t) \in \mathbb{R}^2$, $\theta_i(t) \in \mathbb{R}$, $\gamma_i^x(t) \in \mathbb{R}$, or $\gamma_i^y(t) \in \mathbb{R}$. For this purpose, consider the state transformation given by $\tilde{\phi}_i(t) \triangleq \phi_i(t) - \phi_0(t)$, $i = 1,\ldots,n$. Using this state transformation with (2.21) and (2.22) yields

$$\dot{\tilde{\phi}}_i(t) \;=\; -q_i(t) - \tau \mathrm{sgn}\big(q_i(t)\big) - \dot{\phi}_0(t), \tag{2.24}$$

$$q_i(t) \;=\; \sum_{j\in\mathcal{N}_i^f} \big(\tilde{\phi}_i(t) - \tilde{\phi}_j(t)\big) + k_i \tilde{\phi}_i(t). \tag{2.25}$$

By letting $\tilde{\phi}(t) \triangleq [\tilde{\phi}_1(t),\ldots,\tilde{\phi}_n(t)]^T$, (2.24) and (2.25) can be written in the compact form as

$$\dot{\tilde{\phi}}(t) \;=\; -q(t) - \tau \mathrm{sgn}\big(q(t)\big) - \mathbf{1}_n \dot{\phi}_0(t), \tag{2.26}$$

$$q(t) \;=\; (\mathcal{L}(\mathcal{G}) + K)\tilde{\phi}(t), \tag{2.27}$$

where $K \triangleq \mathrm{diag}([k_1,\ldots,k_n]^T)$. Now, consider the Lyapunov function candidate $V(\tilde{\phi}) = \frac{1}{2}\tilde{\phi}^T(\mathcal{L}(\mathcal{G}) + K)\tilde{\phi}$,

where its time derivative along the trajectory of (2.26) is given by

$$\begin{aligned}\dot{V}(\tilde{\phi}(t)) &= \tilde{\phi}^T(\mathcal{L}(\mathcal{G})+K)\Big(-(\mathcal{L}(\mathcal{G})+K)\tilde{\phi}(t)-\tau\mathrm{sgn}\Big[(\mathcal{L}(\mathcal{G})+K)\tilde{\phi}(t)\Big]-\mathbf{1}_n\dot{\phi}_0(t)\Big)\\ &\leq -\tilde{\phi}^T(\mathcal{L}(\mathcal{G})+K)^2\tilde{\phi}(t)-\tau\|(\mathcal{L}(\mathcal{G})+K)\tilde{\phi}(t)\|_1+|\dot{\phi}_0(t)|\|(\mathcal{L}(\mathcal{G})+K)\tilde{\phi}(t)\|_1\\ &\leq -\tilde{\phi}^T(\mathcal{L}(\mathcal{G})+K)^2\tilde{\phi}(t)-(\tau-\omega)\|(\mathcal{L}(\mathcal{G})+K)\tilde{\phi}(t)\|_1. \end{aligned} \tag{2.28}$$

Since $\mathcal{L}(\mathcal{G})+K\in\mathbb{R}_+^{n\times n}$ (Lemma 2 of [102]) and $(\tau-\omega)>0$ by definition, $\dot{V}(\tilde{\phi}(t))$ is negative definite. Therefore, from Theorem 3.1 of [103], $\tilde{\phi}(t)\to 0$ as $t\to\infty$; or equivalently, $\phi_i(t)\to\phi_0(t)$ as $t\to\infty$. It now readily follows from the limit properties along with the squeeze theorem [104] that $p_i(t)\to R(\theta_0(t))S(\gamma_0^x(t),\gamma_0^y(t))\xi_i$, and $c_i(t)\to c(t)$ as $t\to\infty$; hence, $c_i(t)+p_i(t)\to\rho_i(t)$ as $t\to\infty$.

Next, for the main network layer (2.18), let's consider the state transformation

$$z_i(t)\triangleq x_i(t)-p_i(t)-c_i(t),\quad i=1,\ldots,n. \tag{2.29}$$

Using (2.29), (2.18) can be rewritten as $\dot{z}_i(t)=-\sum_{j\in\mathcal{N}_i^f}\big(z_i(t)-z_j(t)\big)-k_iz_i(t)$. Define $z(t)\triangleq[z_1(t),\ldots,z_n(t)]^T$, then the last expression can be written in the compact form as

$$\dot{z}(t) \;=\; -\big((\mathcal{L}(\mathcal{G})+K)\otimes\mathrm{I}_2\big)z(t), \tag{2.30}$$

Since it is assumed that there exists at least one capable agent in the network (i.e, at least one of the diagonal elements of K is equal to 1), it follows that $\mathcal{L}(\mathcal{G})+K\in\mathbb{R}_+^{n\times n}$, and hence, $-\big(\mathcal{L}(\mathcal{G})+K\big)$ is a Hurwitz matrix. As a direct consequence, $z(t)\to 0$ as $t\to\infty$; thus, $x_i(t)\to p_i(t)+c_i(t)$. Hence, using the limit properties, (2.23) holds and proof is now complete. ■

Remark 2.4.1. *Theorem 2.4.1 shows that under the proposed algorithm given by (2.18) and (2.21),* $\lim_{t\to\infty}\big(\big(x_i(t)-x_j(t)\big)-\big(p_i(t)-p_j(t)\big)\big)=\lim_{t\to\infty}\big(\big(x_i(t)-x_j(t)\big)-R(\theta_0(t))S(\gamma_0^x(t),\gamma_0^y(t))(\xi_i-\xi_j)\big)=0$ *holds; that is, agents has formed the desired formation. Note that (2.23) also implies that each agent is translating a distance $c(t)$ or the formation is tracking the target.*

Remark 2.4.2. *Similar to Remark 2.3.2, if $c^x(t)$, $c^y(t)$, $\theta_0(t)$, $\gamma_0^x(t)$, and $\gamma_0^y(t)$ are constants, then Theorem 2.4.1's results still hold without the need for signum function in (2.21) and (2.22); i.e.,*

$$\dot{\phi}_i(t) \;=\; -\sum_{j\in\mathcal{N}_i^f}\big(\phi_i(t)-\phi_j(t)\big)-k_i\big(\phi_i(t)-\phi_0\big). \tag{2.31}$$

We can also reach a similar conclusion for the case when some of these signals are constant and the respective signum functions for those are removed from (2.21) and (2.22).

Remark 2.4.3. *Similar to Remark 2.3.3, a positive design parameter α can be used in the main network layer given by (2.18) as*

$$\dot{x}_i(t)=\alpha\Big[-\sum_{j\in\mathcal{N}_i^f}\Big(\big(x_i(t)-p_i(t)-c_i(t)\big)-\big(x_j(t)-p_j(t)-c_j(t)\big)\Big)-k_i\big(x_i(t)-p_i(t)-c_i(t)\big)\Big] \\ +\dot{p}_i(t)+\dot{c}_i(t),\;\; x_i(0)=x_{i0}, \tag{2.32}$$

in order to improve convergence rate of the networked multiagent system. In this case, the proof of Theorem 2.4.1 remains identical with the term $(\mathcal{L}(\mathcal{G})+K)$ replaced with $\alpha(\mathcal{L}(\mathcal{G})+K)$ in (2.30). We can also reach a similar conclusion when another positive design parameter is introduced to the other network layers given by (2.21) and (2.22).

Remark 2.4.4. *Similar to Remark 2.3.4, the proposed algorithm of this section can be also extended to a three dimensional case with $x_i(t)\in\mathbb{R}^3$. In this case, $p_i(t)\in\mathbb{R}^3$ can be redefined as*

$$p_i(t)\triangleq R\big(\theta_i^x(t),\theta_i^y(t),\theta_i^z(t)\big)S\big(\gamma_i^x(t),\gamma_i^y(t),\gamma_i^z(t)\big)\xi_i, \tag{2.33}$$

where $\theta_i^x(t)\in\mathbb{R}$, $\theta_i^y(t)\in\mathbb{R}$, and $\theta_i^z(t)\in\mathbb{R}$ are the rotation angles corresponding to yaw, pitch, and roll, respectively, $R\big(\theta_i^x(t),\theta_i^y(t),\theta_i^z(t)\big)$ is the rotation matrix, $\gamma_i^x(t)\in\mathbb{R}$, $\gamma_i^y(t)\in\mathbb{R}$, and $\gamma_i^z(t)\in\mathbb{R}$ are the scaling factors for each dimension, and $S\big(\gamma_i^x(t),\gamma_i^y(t),\gamma_i^z(t)\big)\triangleq diag([\gamma_i^x(t),\gamma_i^y(t),\gamma_i^z(t)]^T)$ is the scaling matrix. In this case, $\phi_i(t)$ represents either $c_i^x(t)$, $c_i^y(t)$, $\theta_i^x(t)$, $\theta_i^y(t)$, $\theta_i^z(t)$, $\gamma_i^x(t)$, $\gamma_i^y(t)$, or $\gamma_i^z(t)$ that satisfies (2.21) and (2.22).

Remark 2.4.5. *The proposed multiplex networks-based spatial formation control algorithm given by (2.18) and (2.21) can be also readily extended to the case where the graph $\mathcal{G}$ is directed under the assumption that there exists at least one capable agent at the root of the spanning tree [77]. A discussion similar to Remark 2.3.6 also holds for the results of this section.*

2.4.2 Multiagent Formation Tracking with Connectivity Maintenance and Collision Avoidance through Multiplex Information Networks

In this subsection, we use tools and methods from differential potential fields (see, for example, [5, 93, 94, 106, 107] and references therein) and generalize the results of Section 2.4.1 to allow connectivity maintenance and collision avoidance that are needed in real-world applications. For this purpose, we let each agent have a communication range as given in Figure 2.3. Specifically, we assume that two arbitrary agents can only exchange information if their relative distance is less than R, i.e., $\|x_{ij}\|_2 < R$. Furthermore, a collision region is defined as a small disk area with radius $r < d < R$ centered at agent i as depicted in this figure. In the same way, we define an escape region as a ring with radius $\Delta < r < R$ also centered at agent i. The region within the collision region and escape region $(d < r < \Delta)$ is called free region.

In what follows, the gradient of a scalar function $f(x)$ is defined by $\nabla_x f = \frac{\partial f}{\partial x}$ with $\frac{\partial f}{\partial x}$ being a column vector as in, for example, [106] and Section 2.4.3 of [108].

We now define a (repulsive) differential potential function for the purpose of collision avoidance as

$$V_{Rij}(x_{ij}) \triangleq \begin{cases} \left(\frac{1}{\|x_{ij}\|_2^2} - \frac{1}{d^2}\right)^2 & \text{if } \|x_{ij}\|_2 \leq d,\ j \in \mathcal{N}_i, \\ 0 & \text{otherwise,} \end{cases} \tag{2.34}$$

where

$$\frac{\partial V_{Rij}(x_{ij})}{\partial x_i} = \begin{cases} -4\left(\frac{1}{\|x_{ij}\|_2^2} - \frac{1}{d^2}\right)\frac{x_{ij}}{\|x_{ij}\|_2^4} & \\ \qquad \text{if } \|x_{ij}\|_2 \leq d,\ j \in \mathcal{N}_i, & \\ 0 \qquad \text{otherwise.} & \end{cases} \tag{2.35}$$

Next, we define a (attractive) differential potential function for the purpose of connectivity maintenance as

$$V_{Cij}(x_{ij}) \triangleq \begin{cases} \frac{(\|x_{ij}\|_2 - \Delta)^2}{R - \|x_{ij}\|_2} & \text{if } \|x_{ij}\|_2 \geq \Delta,\ j \in \mathcal{N}_i^f, \\ 0 & \text{otherwise,} \end{cases} \tag{2.36}$$

where

$$\frac{\partial V_{Cij}(x_{ij})}{\partial x_i} = \begin{cases} \frac{(\|x_{ij}\|_2-\Delta)(2R-\Delta-\|x_{ij}\|_2)}{(R-\|x_{ij}\|_2)^2\|x_{ij}\|_2} x_{ij} \\ \qquad\qquad \text{if } \|x_{ij}\|_2 \geq \Delta,\ j \in \mathcal{N}_i^f, \\ \\ 0 \qquad\quad \text{otherwise.} \end{cases} \tag{2.37}$$

Note that $V_{Rij} = V_{Rji}$ and $V_{Cij} = V_{Cji}$ as well as $V_{Rij} = V_{Cij} = 0$ for $i = j$. Note also that $\partial V_{Rij}(x_{ij})/\partial x_i$ and $\partial V_{Cij}(x_{ij})/\partial x_i$ defined in (2.35) and (2.37) are continuous. The repulsive differential potential function V_{Rij} is smoothly activated when $\|x_{ij}\|_2 \leq d$ and grows to infinity as $\|x_{ij}\|_2$ approaches 0. In addition, the attractive differential potential function V_{Cij} is smoothly activated when $\|x_{ij}\|_2 \geq \Delta$ and grows to infinity as $\|x_{ij}\|_2$ approaches R. Notice that V_{Rij} applies to agent i and any agent j who are neighbor of i (i.e., $j \in \mathcal{N}_i$), while V_{Cij} only affects agent i and its formation neighbors (i.e., $j \in \mathcal{N}_i^f$). In addition, we assume that the desired distance between any two arbitrary agents lies in the free region, where this implies that the scaling factors need to be lower and upper bounded such that this assumption is not violated.

Based on the above definitions, we generalize the results of the previous section by considering the distributed spatial formation control algorithm given by

$$\begin{aligned} \dot{x}_i(t) \;=\; & -\sum_{j\in\mathcal{N}_i^f}\Big(\big(x_i(t)-p_i(t)-c_i(t)\big)-\big(x_j(t)-p_j(t)-c_j(t)\big)\Big)-k_i\big(x_i(t)-p_i(t)-c_i(t)\big) \\ & +\dot{p}_i(t)+\dot{c}_i(t)-\sum_{j\in\mathcal{N}_i}\frac{\partial V_{Rij}(x_{ij})}{\partial x_i}-\sum_{j\in\mathcal{N}_i^f}\frac{\partial V_{Cij}(x_{ij})}{\partial x_i}, \quad x_i(0)=x_{i0}, \end{aligned} \tag{2.38}$$

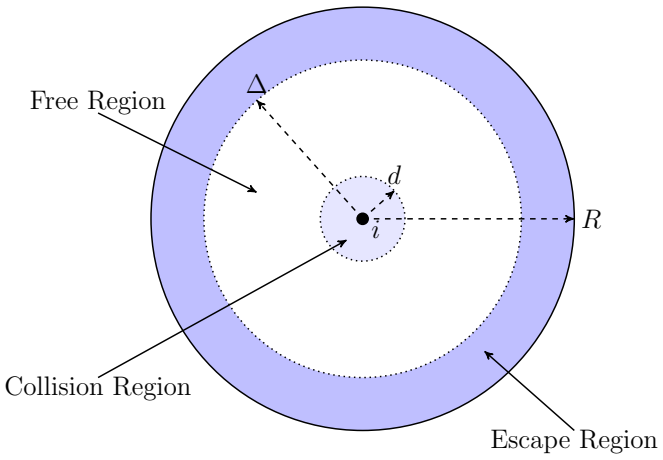


Figure 2.3: Communication range of agent i.

Since we can achieve connectivity maintenance and collision avoidance by only modifying the main network layer in (2.18) as (2.38), all other network layers given by (2.21) remain unchanged in this setting. The following standard assumption is necessary for the next result.

Assumption 2.4.1. *$\partial V_{Rij}(x_{ij})/\partial x_i$ and $\partial V_{Cij}(x_{ij})/\partial x_i$ vanish over time.*

The above assumption implies that the potential field $\partial V_{Rij}(x_{ij})/\partial x_i$ (resp., $\partial V_{Cij}(x_{ij})/\partial x_i$) is able to create a repulsive force (resp., an attractive force) to push two agents out of the collision region (resp., to pull two neighboring agents back to the free region with relative to each other) without causing agents to stuck in locked configurations (i.e., not being stuck in local minima). This assumption is standard in the networked multiagent systems literature that adopts tools and methods from differential potential fields (see Remark 2.4.6 below for further discussion). We further note that once two neighboring agents are in the free region with relative to each other, $\partial V_{Rij}(x_{ij})/\partial x_i$ and $\partial V_{Cij}(x_{ij})/\partial x_i$ equal to zero (i.e., they vanish by definition).

Theorem 2.4.2. *Consider the networked multiagent system given by (2.38) and (2.21), where agents exchange their local measurements using an undirected and connected graph $\mathcal{G}$. If initially agents are connected with their formation neighbors and there is no collision, and Assumption 2.4.1 holds, then (2.23) holds for all $i = 1,\ldots,n$ with connectivity maintenance and collision avoidance for all $t \geq 0$.*

Proof. Following the discussion given after (2.38), we first note that the other network layers represented by (2.21) remain unchanged in the setting of this theorem. Hence, from the first paragraph of the proof of Theorem 2.4.1, $c_i(t) + p_i(t) \to \rho_i(t)$ as $t \to \infty$ holds.

Using the state transformation given by (2.29), we next note that (2.38) can be rewritten as

$$\dot{z}_i(t) \;=\; -\sum_{j\in\mathcal{N}_i^f}\Big(z_i(t)-z_j(t)\Big) - k_i z_i(t) - \sum_{j\in\mathcal{N}_i}\frac{\partial V_{Rij}(x_{ij})}{\partial z_i} - \sum_{j\in\mathcal{N}_i^f}\frac{\partial V_{Cij}(x_{ij})}{\partial z_i}. \tag{2.39}$$

Note also that $\frac{\partial V_{Rij}(x_{ij})}{\partial x_i(t)} = \frac{\partial V_{Rij}(x_{ij})}{\partial z_i(t)}$ and $\frac{\partial V_{Cij}(x_{ij})}{\partial x_i(t)} = \frac{\partial V_{Cij}(x_{ij})}{\partial z_i(t)}$. We define

$$V_{Ai}(z(t)) \triangleq \frac{1}{2}\sum_{j\in\mathcal{N}_i^f}\|z_i(t)-z_j(t)\|_2^2 + \frac{1}{2}k_i\|z_i(t)\|_2^2, \tag{2.40}$$

where the partial derivative of (2.40) with respect to $z_i(t)$ is given by $\frac{\partial V_{Ai}(z(t))}{\partial z_i(t)} = \sum_{j\in\mathcal{N}_i^f}\Big(z_i(t)-z_j(t)\Big) + k_i z_i(t)$. Now, we can write

$$\dot{z}_i(t) = -\frac{\partial V_{Ai}(z(t))}{\partial z_i(t)} - \sum_{j\in\mathcal{N}_i} \frac{\partial V_{Rij}(x_{ij})}{\partial z_i(t)} - \sum_{j\in\mathcal{N}_i^f} \frac{\partial V_{Cij}(x_{ij})}{\partial z_i(t)}$$
$$= -\frac{\partial V_{Ai}(z(t))}{\partial z_i(t)} - \sum_{j=1}^{n} \left(\frac{\partial V_{Rij}(x_{ij})}{\partial z_i(t)} + \frac{\partial V_{Cij}(x_{ij})}{\partial z_i(t)} \right). \tag{2.41}$$

Next, consider the continuously differentiable function $V : D_V \times \mathbb{R}^{2n} \to \overline{\mathbb{R}}_+$ given by

$$V(\cdot) = \left(\frac{1}{2} \sum_{i=1}^{n} V_{Ai}(z(t)) + \frac{1}{4} \sum_{i=1}^{n} k_i \|z_i(t)\|_2^2 \right) + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(V_{Rij}(x_{ij}) + V_{Cij}(x_{ij}) \right), \tag{2.42}$$

where $D_V = \{x \in \mathbb{R}^{2n} : \|x_{ij}\|_2 \in (0,R) \ \forall \ j \in \mathcal{N}_i^f$ and $\|x_{ij}\|_2 \in (0,\infty) \ \forall \ j \in \mathcal{N}_i \setminus \mathcal{N}_i^f\}$. For any $c > 0$, let $\Omega = \{(x,z) \in D_V \times \mathbb{R}^{2n} : \ V(\cdot) \leq c\}$ denote the level sets of $V(\cdot)$ and note that

$$\dot{V}(\cdot) = \begin{bmatrix} \left(\frac{\partial V}{\partial z_1}\right)^{\mathrm{T}} & \left(\frac{\partial V}{\partial z_2}\right)^{\mathrm{T}} & \dots & \left(\frac{\partial V}{\partial z_n}\right)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \\ \vdots \\ \dot{z}_n(t) \end{bmatrix}$$
$$= \sum_{i=1}^{n} \left(\frac{\partial V}{\partial z_i} \right)^{\mathrm{T}} \dot{z}_i(t). \tag{2.43}$$

In what follows, we show that $\frac{\partial V}{\partial z_i} = -\dot{z}_i(t)$. To this end, we first write

$$\frac{\partial V}{\partial z_i} = \frac{\partial}{\partial z_i} \left(\frac{1}{2} \sum_{i=1}^{n} V_{Ai}(z(t)) + \frac{1}{4} \sum_{i=1}^{n} k_i \|z_i(t)\|_2^2 + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(V_{Rij}(x_{ij}) + V_{Cij}(x_{ij}) \right) \right)$$
$$= \underbrace{\frac{1}{2} \frac{\partial}{\partial z_i} \left(\sum_{i=1}^{n} V_{Ai}(z(t)) \right)}_{A} + \underbrace{\frac{1}{4} \frac{\partial}{\partial z_i} \left(\sum_{i=1}^{n} k_i \|z_i(t)\|_2^2 \right)}_{B} + \underbrace{\frac{1}{2} \frac{\partial}{\partial z_i} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \left(V_{Rij}(x_{ij}) + V_{Cij}(x_{ij}) \right) \right)}_{C}, \tag{2.44}$$

where

$$A = \frac{1}{2} \frac{\partial}{\partial z_i} \left(V_{A1}(z(t)) + V_{A2}(z(t)) + \ldots + V_{An}(z(t)) \right)$$
$$= \frac{1}{2} \frac{\partial V_{A1}(z(t))}{\partial z_i} + \frac{1}{2} \frac{\partial V_{A2}(z(t))}{\partial z_i} + \ldots + \frac{1}{2} \frac{\partial V_{An}(z(t))}{\partial z_i}$$
$$= \frac{1}{2} \frac{\partial V_{Ai}(z(t))}{\partial z_i} + \frac{1}{2} \sum_{j=1, \ j\neq i}^{n} \left(\frac{\partial V_{Aj}(z(t))}{\partial z_i} \right). \tag{2.45}$$

Note that if $i \notin \mathcal{N}_j^f$ then $\frac{\partial V_{Aj}(z(t))}{\partial z_i} = 0$ and if $i \in \mathcal{N}_j^f$ then

$$\begin{aligned}\frac{\partial V_{Aj}(z(t))}{\partial z_i} &= \frac{\partial}{\partial z_i}\left(\tfrac{1}{2}\textstyle\sum_{k\in\mathcal{N}_j^f}\|z_j(t)-z_k(t)\|_2^2 + \tfrac{1}{2}k_j\|z_j(t)\|_2^2\right)\\ &= -\big(z_j(t)-z_i(t)\big) = \big(z_i(t)-z_j(t)\big),\end{aligned} \tag{2.46}$$

with respect to agents i and j only. This implies that we graph-wise have

$$A = \frac{1}{2}\frac{\partial V_{Ai}(z(t))}{\partial z_i} + \frac{1}{2}\sum_{j\in\mathcal{N}_i^f}\big(z_i(t)-z_j(t)\big). \tag{2.47}$$

Furthermore, we have

$$B = \frac{k_i}{4}\frac{\partial}{\partial z_i}\left(\|z_i(t)\|_2^2\right) = \frac{k_i z_i(t)}{2}. \tag{2.48}$$

Finally, by symmetry of the function V_{Rij} and V_{Cij}, we have

$$\begin{aligned}C &= \frac{1}{2}\left(2\sum_{j=1}^{n}\left(\frac{\partial V_{Rij}(x_{ij})}{\partial z_i(t)} + \frac{\partial V_{Cij}(x_{ij})}{\partial z_i(t)}\right)\right)\\ &= \sum_{j=1}^{n}\left(\frac{\partial V_{Rij}(x_{ij})}{\partial z_i(t)} + \frac{\partial V_{Cij}(x_{ij})}{\partial z_i(t)}\right).\end{aligned} \tag{2.49}$$

Substituting (2.47), (2.48), and (2.49) back into (2.44) yields

$$\begin{aligned}\frac{\partial V}{\partial z_i} &= \frac{1}{2}\frac{\partial V_{Ai}(z(t))}{\partial z_i} + \frac{1}{2}\sum_{j\in\mathcal{N}_i^f}\big(z_i(t)-z_j(t)\big) + \frac{k_i z_i(t)}{2} + \sum_{j=1}^{n}\left(\frac{\partial V_{Rij}(x_{ij})}{\partial z_i(t)} + \frac{\partial V_{Cij}(x_{ij})}{\partial z_i(t)}\right)\\ &= \frac{1}{2}\frac{\partial V_{Ai}(z(t))}{\partial z_i} + \frac{1}{2}\frac{\partial V_{Ai}(z(t))}{\partial z_i} + \sum_{j=1}^{n}\left(\frac{\partial V_{Rij}(x_{ij})}{\partial z_i(t)} + \frac{\partial V_{Cij}(x_{ij})}{\partial z_i(t)}\right)\\ &= \frac{\partial V_{Ai}(z(t))}{\partial z_i} + \sum_{j=1}^{n}\left(\frac{\partial V_{Rij}(x_{ij})}{\partial z_i(t)} + \frac{\partial V_{Cij}(x_{ij})}{\partial z_i(t)}\right)\\ &= -\dot{z}_i(t),\end{aligned} \tag{2.50}$$

where the second equality comes from the expression $\frac{\partial V_{Ai}(z(t))}{\partial z_i(t)} = \sum_{j\in\mathcal{N}_i^f}\Big(z_i(t)-z_j(t)\Big) + k_i z_i(t)$ given in the paragraph after (2.40). Thus, (2.43) now becomes

$$\dot{V}(\cdot) = \sum_{i=1}^{n} -\dot{z}_i^{\mathrm{T}}(t)\dot{z}_i(t) = \sum_{i=1}^{n} -\|\dot{z}_i(t)\|_2^2 \leq 0. \tag{2.51}$$

Since $\dot{V}(\cdot) \leq 0$, the level sets Ω are positively invariant, and hence, $V_{Ai}(z(t))$, $V_{Rij}(x_{ij})$ and $V_{Cij}(x_{ij})$ are bounded [106]. If for some $j \in \mathcal{N}_i$ such that $\|x_{ij}\|_2 \to 0$, then $V_{Rij} \to \infty$. Therefore, by the continuity of V in D_V, it follows that $\|x_{ij}\|_2 > 0$ for all $j \in \mathcal{N}_i(t)$. Likewise, if for some $j \in \mathcal{N}_i^f$ such that $\|x_{ij}\|_2 \to R$, then $V_{Cij} \to \infty$. Once again, by the continuity of V in D_V, it follows that $\|x_{ij}\|_2 < R$ for all $j \in \mathcal{N}_i^f$. Thus, if the agents are initially connected with their formation neighbors and there is no collision, then collision avoidance between agent i and its neighbors (i.e., $j \in \mathcal{N}_i$) and connectivity maintenance between agent i and its formation neighbors (i.e., $j \in \mathcal{N}_i^f$) are guaranteed for all $t \geq 0$.

The level sets Ω are closed by the continuity of V in D_V and they are bounded since $\dot{V}(\cdot) \leq 0$, and hence, they are compact. By LaSalle's invariance principle, all trajectories starting in Ω converge to the largest invariant set in $E \triangleq \{(x,z) \in D_V \times \mathbb{R}^{2n} : \dot{V}(\cdot) = 0\} = \{(x,z) \in D_V \times \mathbb{R}^{2n} : \dot{z}(t) = 0\}$. From (2.41), this implies that $\frac{\partial V_{Ai}(z(t))}{\partial z_i(t)} = -\sum_{j=1}^{n} \left(\frac{\partial V_{Ri}(x_{ij})}{\partial z_i(t)} + \frac{\partial V_{Ci}(x_{ij})}{\partial z_i(t)} \right)$ holds. From Assumption 2.4.1 (i.e., the agents are not stuck in local minima), the term $-\sum_{j=1}^{n} \left(\frac{\partial V_{Ri}(x_{ij})}{\partial z_i(t)} + \frac{\partial V_{Ci}(x_{ij})}{\partial z_i(t)} \right)$ vanishes over time. Thus, trajectories starting in Ω converge to $M \subset E$ defined by $M \triangleq \{(x,z) \in D_V \times \mathbb{R}^{2n} : \frac{\partial V_{Ai}(z(t))}{\partial z_i(t)} = -\sum_{j=1}^{n} \left(\frac{\partial V_{Ri}(x_{ij})}{\partial z_i(t)} + \frac{\partial V_{Ci}(x_{ij})}{\partial z_i(t)} \right) = 0 \ \forall \ i = 1, \ldots, n\}$. Finally, analyzing $\frac{\partial V_{Ai}(z(t))}{\partial z_i(t)} = 0 \ \forall \ i = 1, \ldots, n$ now follows from the second paragraph of the proof of Theorem 2.4.1 owing to the fact that $\frac{\partial V_{Ai}(z(t))}{\partial z_i(t)} = \sum_{j \in \mathcal{N}_i^f} \Big(z_i(t) - z_j(t) \Big) + k_i z_i(t)$, where the right hand side of this expression was used there. In other words, the largest invariant set of M is trivial in this case and equals to $z_i(t) = 0$ for all $i = 1, \ldots, n$ from the proof of Theorem 2.4.1. Thus, from the discussion given in the last part of Theorem 2.4.1's proof, $x_i(t) \to p_i(t) + c_i(t)$ as $t \to \infty$. Recalling the fact that the other network layers represented by (2.21) remain unchanged once again, $\phi_i(t) \to \phi_0(t)$ as $t \to \infty$ or $c_i(t) + p_i(t) \to \rho_i(t)$ as $t \to \infty$ from the first paragraph of the proof of Theorem 2.4.1. Hence, (2.23) holds by the limit properties. ■

Remark 2.4.6. *Without the assumption that agents are not stuck in local minima (i.e., Assumption 2.4.1), one of the following two cases occurs based on the discussion given in the last paragraph of Theorem 2.4.2's proof:*

- ***i)*** *Agents can converge to the free region and (2.23) holds.*
- ***ii)*** *It follows from LaSalle's invariance principle and (2.41) that* $\frac{\partial V_{Ai}(z(t))}{\partial z_i(t)} = -\sum_{j=1}^{n} \left(\frac{\partial V_{Ri}(x_{ij})}{\partial z_i(t)} + \frac{\partial V_{Ci}(x_{ij})}{\partial z_i(t)} \right)$ *holds, where both left and right hand sides of this equation are not equal to zero.*

Note that the latter case implies that agents are stuck in local minima. Although there are several methods to avoid local minima (see, for example, [109–111]), it is an open problem in the networked multiagent

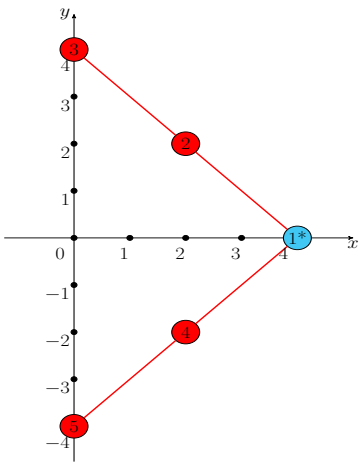


Figure 2.4: A given desired formation for the example in Section 2.4.3

systems literature that adopts tools and methods from differential potential fields. Yet, for example, one can use the idea stated in [109], which assumes that agents that are stuck can be detected (e.g., agents that are not moving for a specific amount of time) and a virtual force

$$F_{vi} \triangleq \begin{cases} F_i & \text{if } \dot{z}_i(t) = 0 \text{ and } \frac{\partial V_{Ai}(z(t))}{\partial z_i(t)} \neq 0, \\ 0 & \text{otherwise}, \end{cases} \tag{2.52}$$

is generated to push such agents out of the local minima with F_i being a random finite value for each agent (to preserve continuity, one can apply filtered version of this force). This force can eventually yield all agents to converge to the free region such that (2.23) follows.

2.4.3 Illustrative Numerical Example

We now present a numerical example to illustrate the results of Sections 2.4.1 and 2.4.2. For this purpose, consider a group of 5 agents with agent 1 being the capable agent and assume that all agents are subject to random initial conditions. We choose ξ_i for each agent to obtain the desired formation depicted in Figure 2.4. Specifically, to illustrate the results of Theorem 2.4.1, we use (2.32) with $\alpha = 5$. In addition, for (2.21), we use $c^x(t) = t$; $c^y(t) = \sin(t)$; $\theta_0 = 0$; and low-pass filtered version of $\psi(t) = 0.5$ for $t \in [0, 10)$, $\psi(t) = 0.25$ for $t \in [10, 20)$, and $\psi(t) = 1$ for $t \in [20, \infty)$ for both $\gamma_0^x(t)$ and $\gamma_0^y(t)$. The time derivatives of $c_i^x(t)$, $c_i^y(t)$, $\theta_i(t)$, $\gamma_i^x(t)$, and $\gamma_i^y(t)$ are all upper bounded by 5 or a smaller constant, and hence, we set $\tau = 5$. Figure 2.5 shows that the considered group of agents perform target tracking while simultaneously forming,

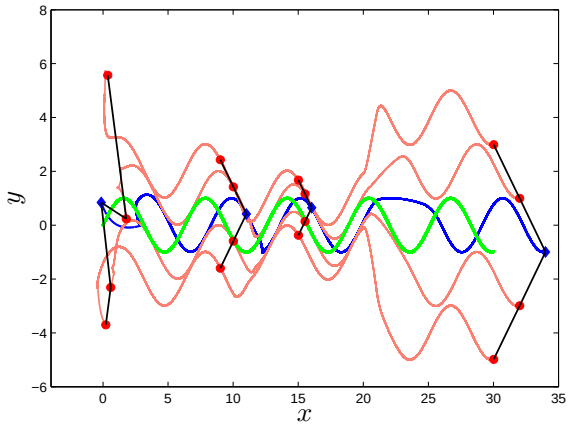
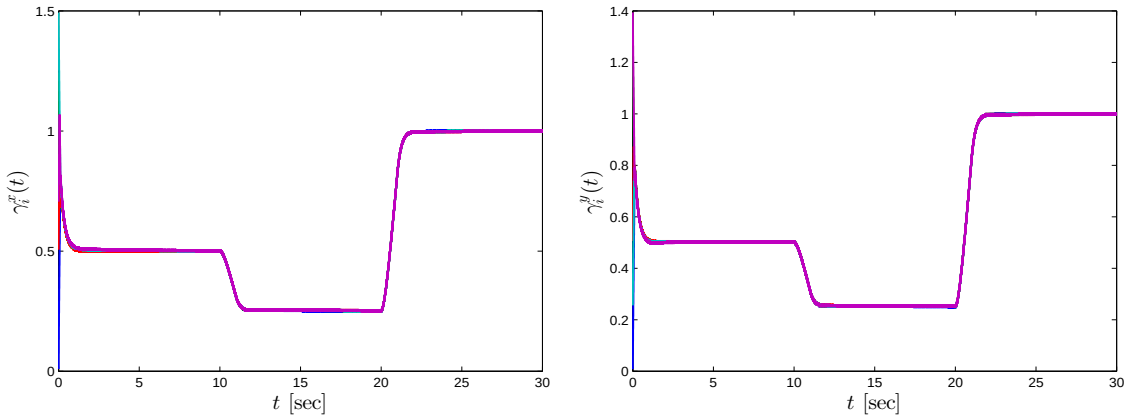


Figure 2.5: Target tracking using the proposed multiplex networks-based spatial formation control algorithm in Theorem 2.4.1.

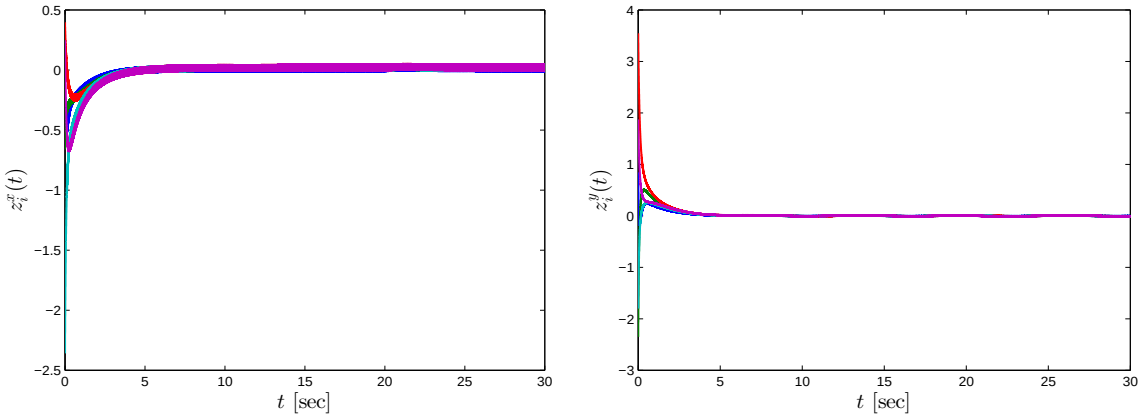
maintaining, and spatially altering their formation in time. Furthermore, Figures 2.6 and 2.7 show that $\gamma_i(t)$ converges to the desired values of the scaling factors and the state transformation variable $z_i(t)$ approaches to zero, respectively.

Next, we illustrate the results of Theorem 2.4.2. In particular, we add the potential field functions to (2.32) as in (2.38) and set $d = 0.5$, $\Delta = 6$, and $R = 8$, where all other design parameters remain the same. Figure 2.8 shows that the considered group of agents achieves the same level of performance as in Figure 2.5 while maintaining connectivity and avoiding collisions. In addition, Figure 2.9 shows the evolution of distances between agents during $t \in [0,5]$ seconds and illustrates collision avoidance properties of the proposed multiplex networks-based spatial formation control algorithm.



(a) Time evolution of $\gamma_i^x(t)$. (b) Time evolution of $\gamma_i^y(t)$.

Figure 2.6: Time evolution of the scaling factors in Figure 2.5.



(a) Time evolution of $z_i^x(t)$. (b) Time evolution of $z_i^y(t)$

Figure 2.7: Time evolution of $z_i(t)$ in Figure 2.5.

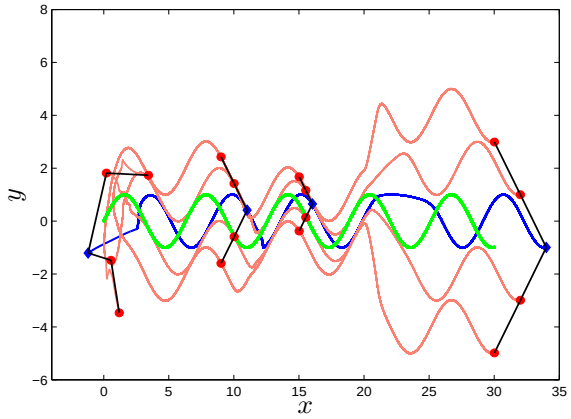


Figure 2.8: Target tracking using the proposed multiplex networks-based spatial formation control algorithm in Theorem 2.4.2.

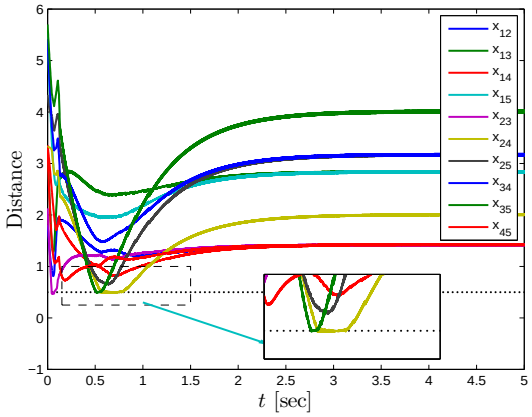
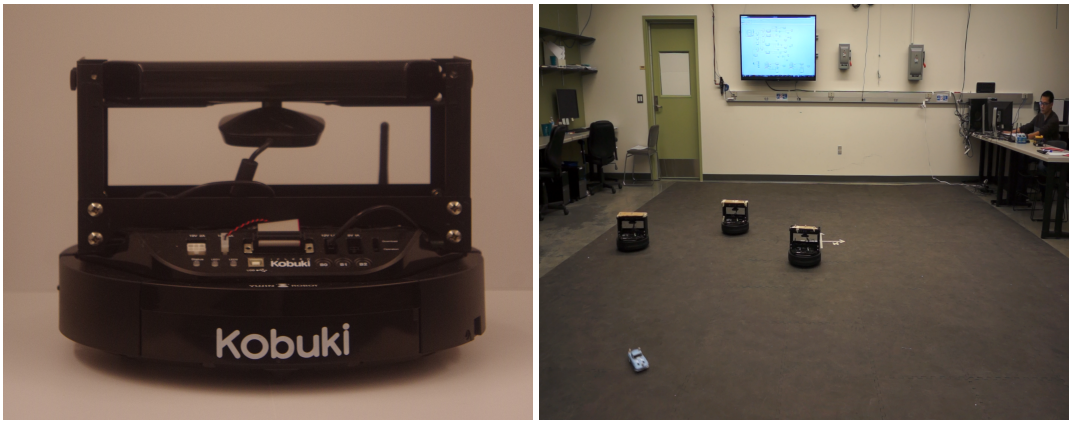


Figure 2.9: Time evolution of distances between agents in Figure 2.8.



(a) Qbot 2 (b) Workspace

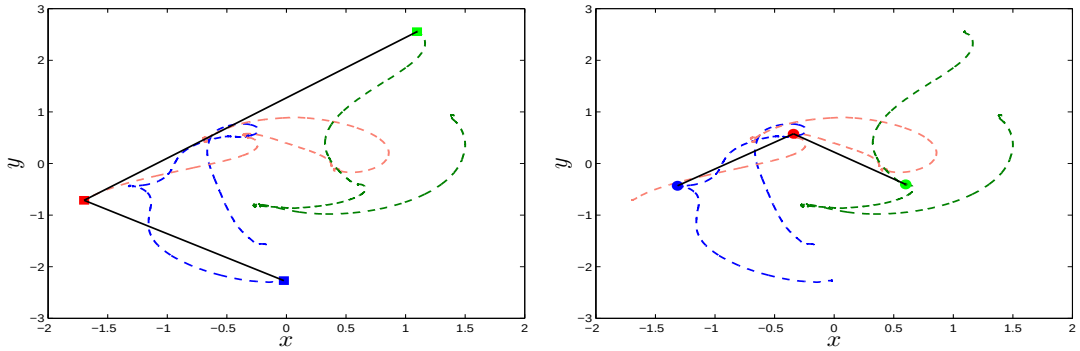
Figure 2.10: Laboratory-level experimental setup.

2.5 Multiagent Formation Experiments

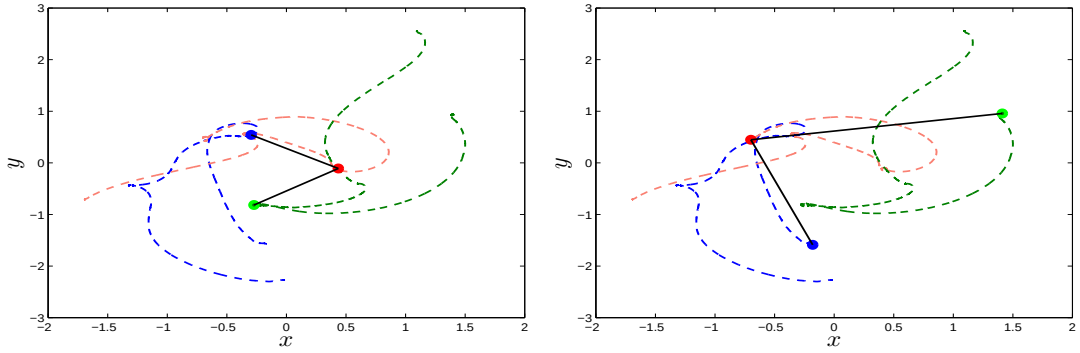
To justify theoretical results, proposed algorithms of this paper are implemented on a group of 3 robots. In the first experiment, the robots form a V-shape formation with the size and orientation changed overtime. For the second experiment, the robots also achieve the same formation while tracking a dynamic target. In the third experiment, the robot formation is controlled to pass through a narrow passage. The mobile robot platform used in our experiments is Qbot 2 (Figure 2.10(a)). In addition, a motion capture system is used to detect the position and orientation of each robot. However, each robot is limited to know only its local measurements and exchanges these data with its neighbors via a wireless network. The motion capture system is able to cover the workspace shown in Figure 2.10(b).

2.5.1 Experiment 1: Formation Density and Orientation Control in Formation Assignment

In this experiment, the robots are implemented with algorithms (2.2), (2.3) and (2.4). The desired scaling factor γ and rotation angle θ are changed by an operator from the computer station. Robot 1 is set as the capable agent, so it is the only one knows these desired values. Initially, the robots are placed randomly in the workspace. The data in Figure 2.11 shows that robots are able to form the formation with $(\gamma,\theta) = (1,0)$ for $t \in [0,36)$, $(\gamma,\theta) = (0.7,-\pi/2)$ for $t \in [36,68)$ and $(\gamma,\theta) = (1.5,\pi/3)$ for $t \in [68,100]$. This experiment has confirmed our Theorem 2.3.1.



(a) $t = 0$ (b) $t = 35$



(c) $t = 67$ (d) $t = 100$

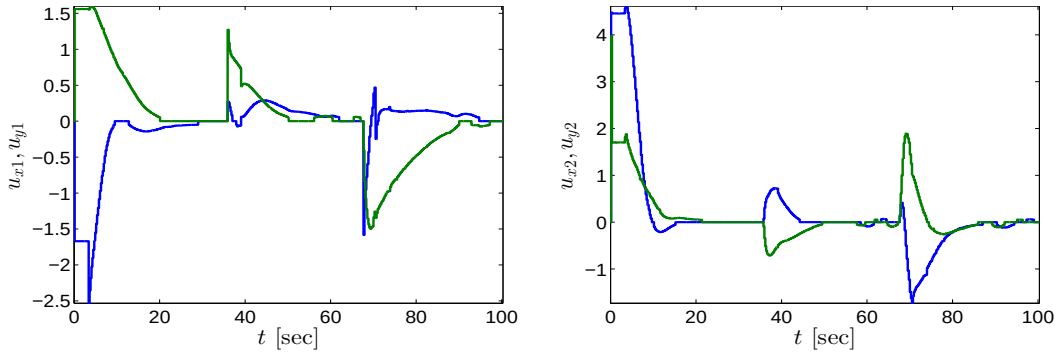
Figure 2.11: Results of multiagent formation experiment 1.

2.5.2 Experiment 2: Spatially Evolving Multiagent Formation Tracking

In this experiment, we implement algorithms (2.38) and (2.21) on the robots. The data in Figure 2.13 illustrates the five configurations of the robots (circles) and the target (square) over time. At t = 0, the robots are far apart from each other while the desired scaling factor is set to $\gamma = 0.7$. At t = 25, the robots are coming closer to form the desired formation and tracking the target. At t = 35, the formation is completed and following the target. As observed from the controller of Robot 1 in Figure 2.14, there is an impulse around t = 40. This is owing to the fact that the operator has just assigned a new scaling factor $\gamma = 1.4$ to the capable agent (i.e., robot 1). At t = 47, the formation with $\gamma = 1.4$ is achieved. At t = 62, the robots are still tracking the target while maintaining the desired formation. This experiment has confirmed our Theorem 2.4.1.

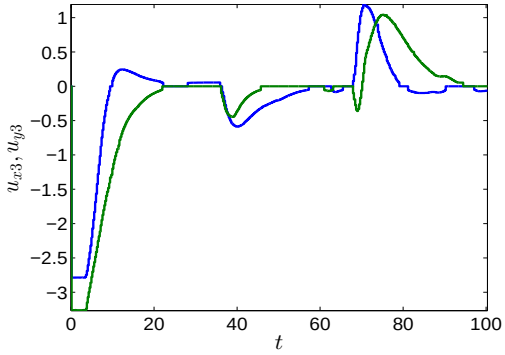
2.5.3 Experiment 3: Formation Passing Through a Narrow Passage

We finally consider the scenario that formation has to track a target and pass through a narrow passage. With the proposed algorithms, we come up with two strategies: For the first strategy, we adjust the



(a) Robot 1 control history

(b) Robot 2 control history



(c) Robot 3 control history

Figure 2.12: Control histories of each robot for multiagent formation experiment 1.

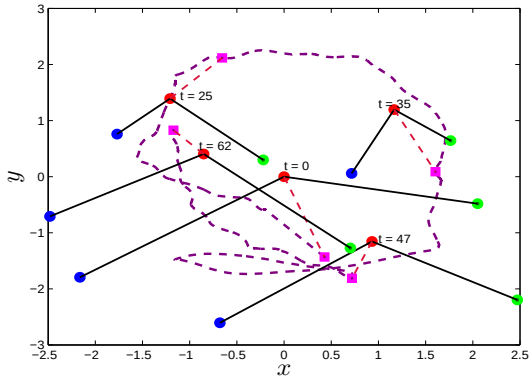
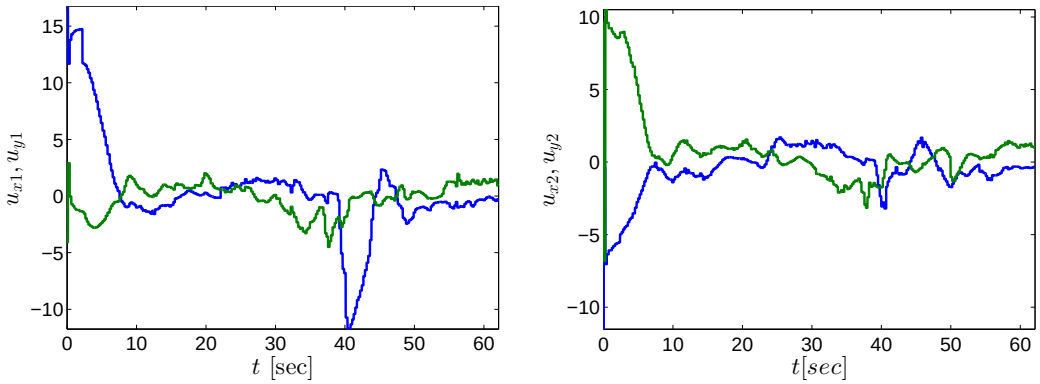


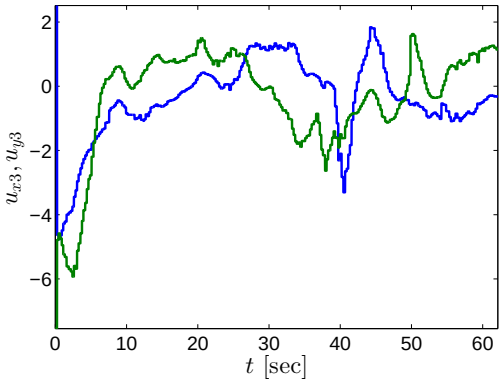
Figure 2.13: Results of multiagent formation experiment 2.

scaling factor γ_x and γ_y to make formation small enough to pass through passage as shown in Figure 2.15. For the second strategy, we observe that the V-shape formation (Figure 2.4) can be compressed to a line formation through setting the scaling factor in x-direction $\gamma_x = 0$. Therefore, in order to pass through the narrow passage, we rotate the V-shape formation by an angle $\theta = \pi/2$ and set $\gamma_y = 0$ (note that, when we rotate the formation by 90 degree, the original x-axis becomes y-axis and vice versa). The results are shown in Figure 2.16.



(a) Robot 1 control history

(b) Robot 2 control history



(c) Robot 3 control history

Figure 2.14: Control histories of each robot for multiagent formation experiment 2.

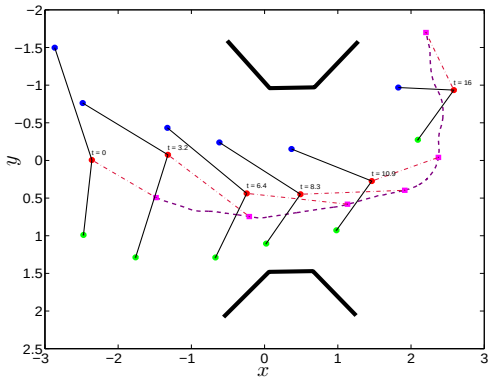


Figure 2.15: Result of multiagent formation experiment 3 with the first strategy.

2.6 Conclusion

In this paper, we investigated how information exchange rules represented by multiplex information networks can be designed to enable spatially evolving multiagent formations. Specifically, we introduced,

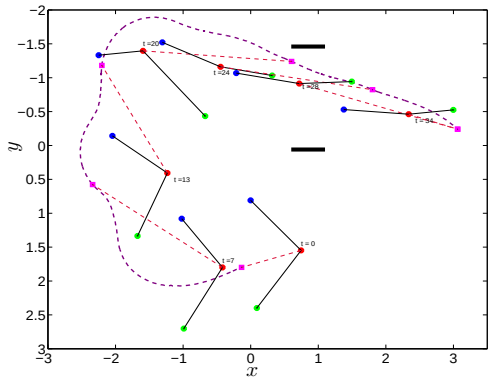


Figure 2.16: Result of multiagent formation experiment 3 with the second strategy.

analyzed, and experimentally validated new distributed control architectures for the formation assignment (i.e., creating a desired formation for the multiagent system in hand) and the formation tracking (i.e., formation control while tracking a dynamic, non-stationary target) problems that allow capable agents to spatially alter size and orientation of the resulting formation without requiring global information exchange ability. Considering multiagent operations with dramatically increasing levels of complexity, the presented multiplex networks-based approach can also be used with many other approaches in multiagent systems to enable advanced distributed information exchange rules to make these systems evolve spatially in adapting to dynamic environments and respond effectively to human interventions. Our future research will include additional theoretical developments and applications for a group of heterogeneous ground and aerial robots with exogenous disturbances, system uncertainties, and communication constraints. We will also consider the cases when the roles of capable agents switch in a given multiagent system.

2.7 Acknowledgments

The authors would like to thank Dr. Zhen Kan from the University of Iowa, Dr. John Singler from the Missouri University of Science and Technology, and the anonymous reviewers for their comments and suggestions.

Chapter 3: On Control of Multiagent Formations through Local Interactions*

We recently showed for multiagent systems with first-order agent dynamics how information exchange rules represented by a network having multiple layers (multiplex information networks) can be designed for enabling spatially evolving multiagent formations. In this paper, we generalize our earlier results for multiagent systems with general linear dynamics. Specifically, we utilize multiplex information networks for formation density control of multiagent systems. The proposed approach allows capable agents to spatially alter density of the resulting formation while tracking a target of interest — without requiring global information exchange ability, and hence, through local interactions. We provide an illustrative numerical example to demonstrate the efficacy of the proposed distributed control architecture.

3.1 Introduction

Multiagent systems require advanced distributed information exchange rules for performing operations with dramatically increasing levels of complexity in order to make these systems evolve spatially for adapting dynamic environments and effectively responding to human interventions. Yet, current distributed control methods lack information exchange infrastructures to enable spatially evolving multiagent formations. This is due to the fact that these methods are designed based on information exchange rules for a network having a single layer (see, for example, [5, 77, 78] and references therein), which leads to multiagent formations with fixed, non-evolving spatial properties. For situations where capable agents have to control the resulting formation through these methods, they can only do so if such vehicles have global information exchange ability — that is not practical for cases involving large numbers of agents and low-bandwidth peer-to-peer communications.

In [113, 114], we showed for multiagent systems with first-order dynamics how information exchange rules represented by a network having multiple layers (multiplex information networks) can be

*This chapter is previously published in [112]. Permission is included in Appendix I.

designed for enabling spatially evolving multiagent formations. In this paper, we generalize our recent results for multiagent systems with general linear dynamics. Specifically, we utilize multiplex information networks for formation density control of multiagent systems. The proposed approach allows capable agents to spatially alter density of the resulting formation while tracking a target of interest — without requiring global information exchange ability, and hence, through local interactions.

In particular, studies in multiplex information networks have recently emerged in the physics and networks science literatures, where they consider system-theoretic characteristics of network dynamics with multiple layers subject to intralayer and interlayer information exchange [10–14, 79, 81]. However, these studies mainly consider cases where all layers perform simple consensus algorithms and analyze the convergence of the overall multiagent systems in the presence of not only intralayer but also interlayer information exchange, and hence, they do not deal with controlling spatial properties of multiagent formations. Note that there are also recent studies on networks of networks by the authors of [83–85]. However, these studies deal with large-scale systems formed from smaller factor networks via graph Cartesian products, and hence, they are also not related with the contribution of this paper.

Spatial multiagent formation control and formation density control in particular is considered by the authors of [86–89] using approaches different from multiplex information networks. Specifically, the authors of [86–88] assume that some of the formation design parameters are known globally by all agents and the authors of [89] assume global knowledge of the complete network at the analysis stage. However, as previously discussed, such assumptions may not be practical in the presence of large numbers of agents and low-bandwidth peer-to-peer communications. From a data security point of view, in addition, it should be noted that one may not desire a multiagent system with all agents sharing some global information about an operation of interest. Throughout this paper, we do not make such assumptions in our multiplex information networks-based formation density control approach.

The organization of this paper is as follows. Section 3.2 introduces the necessary mathematical preliminaries to develop the main results of this paper. Section 3.3 presents the proposed distributed control architecture for density control of multiagent formations with general linear dynamics through local interactions. We provide an illustrative numerical example in Section 3.4 to demonstrate the efficacy of the proposed architecture and concluding remarks are summarized in Section 3.5.

Throughout this paper, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ denotes the set of $n \times 1$ real column vectors, $\mathbb{R}^{n\times m}$ denotes the set of $n \times m$ real matrices, $\mathbb{R}_+$ denotes the set of positive real numbers, $\mathbb{R}_+^{n\times n}$ (resp.,

$\overline{\mathbb{R}}_+^{n\times n}$) denotes the set of $n\times n$ positive-definite (resp., nonnegative-definite) real matrices, $\mathbb{S}_+^{n\times n}$ (resp., $\overline{\mathbb{S}}_+^{n\times n}$) denotes the set of $n\times n$ symmetric positive-definite (resp., symmetric nonnegative-definite) real matrices, $\mathbb{Z}$ denotes the set of integers, $\mathbb{Z}_+$ (resp., $\overline{\mathbb{Z}}_+$) denotes the set of positive (resp., nonnegative) integers, 0_n denotes the $n\times 1$ vector of all zeros, $\mathbf{1}_n$ denotes the $n\times 1$ vector of all ones, $0_{n\times n}$ denotes the $n\times n$ zero matrix, and I_n denotes the $n\times n$ identity matrix. In addition, we write $(\cdot)^{\mathrm{T}}$ for transpose, $(\cdot)^{-1}$ for inverse, $\|\cdot\|_2$ for the Euclidian norm, $\|\cdot\|_F$ for the Frobenius norm, $\lambda_{\min}(A)$ (resp., $\lambda_{\max}(A)$) for the minimum (resp., maximum) eigenvalue of the Hermitian matrix A, $\lambda_i(A)$ for the i-th eigenvalue of A (A is symmetric and the eigenvalues are ordered from least to greatest value), and $\mathrm{diag}(a)$ for the diagonal matrix with the vector a on its diagonal.

3.2 Mathematical Preliminaries

We first recall some basic notions from graph theory, where we refer to [5, 92] for details. In the multiagent literature, graphs are broadly adopted to encode interactions in networked systems. An undirected graph $\mathcal{G}$ is defined by a set $\mathcal{V}_\mathcal{G}=\{1,\ldots,N\}$ of nodes and a set $\mathcal{E}_\mathcal{G}\subset\mathcal{V}_\mathcal{G}\times\mathcal{V}_\mathcal{G}$ of edges. If $(i,j)\in\mathcal{E}_\mathcal{G}$, then the nodes i and j are neighbors and the neighboring relation is indicated with $i\sim j$. The degree of a node is given by the number of its neighbors. Letting d_i be the degree of node i, then the degree matrix of a graph $\mathcal{G}$, $\mathcal{D}(\mathcal{G})\in\mathbb{R}^{N\times N}$, is given by

$$\mathcal{D}(\mathcal{G})\triangleq\mathrm{diag}(d),\quad d=[d_1,\ldots,d_N]^{\mathrm{T}}. \tag{3.1}$$

A path $i_0i_1\ldots i_L$ is a finite sequence of nodes such that $i_{k-1}\sim i_k$, $k=1,\ldots,L$, and a graph $\mathcal{G}$ is connected if there is a path between any pair of distinct nodes. The adjacency matrix of a graph $\mathcal{G}$, $\mathcal{A}(\mathcal{G})\in\mathbb{R}^{N\times N}$, is given by

$$[\mathcal{A}(\mathcal{G})]_{ij}\triangleq\begin{cases}1, & \text{if } (i,j)\in\mathcal{E}_\mathcal{G},\\ 0, & \text{otherwise.}\end{cases}$$

The Laplacian matrix of a graph, $\mathcal{L}(\mathcal{G})\in\overline{\mathbb{S}}_+^{N\times N}$, playing a central role in many graph theoretic treatments of multiagent systems, is given by

$$\mathcal{L}(\mathcal{G})\triangleq\mathcal{D}(\mathcal{G})-\mathcal{A}(\mathcal{G}).$$

Throughout this paper, we model a given multiagent system by a connected, undirected graph $\mathcal{G}$, where nodes and edges represent agents and inter-agent communication links, respectively.

Next we introduce two necessary lemmas for the results of this paper.

Lemma 3.2.1 ([5]). *The spectrum of the Laplacian of a connected, undirected graph can be ordered as*

$$0 = \lambda_1\big(\mathcal{L}(\mathcal{G})\big) < \lambda_2\big(\mathcal{L}(\mathcal{G})\big) \leq \ldots \leq \lambda_N\big(\mathcal{L}(\mathcal{G})\big), \tag{3.2}$$

with $\mathbf{1}_n$ *as the eigenvector corresponding to the zero eigenvalue* $\lambda_1\big(\mathcal{L}(\mathcal{G})\big)$ *and*

$$\mathcal{L}(\mathcal{G})\mathbf{1}_N = 0_N. \tag{3.3}$$

Lemma 3.2.2 ([102]). *Let*

$$G = [\mu_1, \mu_2, \ldots, \mu_N]^{\mathrm{T}}, \quad \mu_i \in \overline{\mathbb{Z}}_+, \quad i = 1, \ldots, N, \tag{3.4}$$

and assume at least one element of G is nonzero. Then, for the Laplacian of a connected, undirected graph,

$$F \triangleq \mathcal{L}(\mathcal{G}) + \mathrm{diag}(G), \tag{3.5}$$

is a positive-definite matrix.

3.3 Control of Multiagent Formations with General Linear Dynamics

In this section, we consider a system with N agents exchanging information among each other using their local measurements according to a connected, undirected graph $\mathcal{G}$. Specifically, let the dynamics of each agent be described by

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t), \quad x_i(0) = x_{i0}, \tag{3.6}$$

$$y_i(t) = Cx_i(t) \tag{3.7}$$

where for agent i, $i = 1, \ldots, N$, $x_i(t) \in \mathbb{R}^n$ denotes the state vector, $u_i(t) \in \mathbb{R}^m$ denotes the control vector, and $y_i \in \mathbb{R}^p$ denotes the output vector with $p \leq n$. In (3.6) and (3.7), in addition, $A \in \mathbb{R}^{n\times n}$ denotes the system

matrix, $B \in \mathbb{R}^{n\times m}$ denotes the control input matrix, and $C \in \mathbb{R}^{p\times n}$ denotes the output matrix such that the triple (A,B,C) is minimal.

Our objective is to design a distributed control signal $u_i(t)$ for each agent i, $i = 1,\ldots,N$, such that the resulting multiagent system not only generates a desired formation but also the density of this formation is spatially altered by capable agents while tracking a target of interest — without requiring global information exchange ability, and hence, through local interactions. For this purpose, we propose the distributed control architecture given by

$$u_i(t) = -K_1 x_i(t) - K_2 z_i(t), \tag{3.8}$$

$$\dot{z}_i(t) = \sum_{i\sim j} \big(y_i(t) - \xi_i(t) - y_j(t) + \xi_j(t)\big) + \mu_i\big(y_i(t) - \xi_i(t) - c(t)\big), \tag{3.9}$$

where $K_1 \in \mathbb{R}^{m\times n}$ and $K_2 \in \mathbb{R}^{m\times p}$ are feedback controller gain matrices, $z_i(t) \in \mathbb{R}^p$ is the integral state vector, and $c(t) \in \mathbb{R}^p$ is the position of a target of interest (i.e., command to be followed by the multiagent system). In (3.9), in addition, $\xi_i(t) \in \mathbb{R}^p$ denotes

$$\xi_i(t) \triangleq \gamma_i(t)\xi_i^*, \tag{3.10}$$

where $\xi_i^* \in \mathbb{R}^p$ captures a desired formation objective, $\gamma_i(t) \in \mathbb{R}$ is an additional network layer satisfying

$$\dot{\gamma}_i(t) = -\alpha \sum_{i\sim j} \big(\gamma_i(t) - \gamma_j(t)\big) - \mu_i \alpha \big(\gamma_i(t) - \gamma^*(t)\big), \tag{3.11}$$

with $\gamma^*(t) \in \mathbb{R}$ being the scaling factor for the density of the resulting formation and $\alpha > 0$, and $\mu_i = 1$ only for capable (i.e., leader) agents and it is zero otherwise. Throughout this paper, we assume that there is at least one capable agent in the multiagent system.

More specifically, our objective is to guarantee

$$y_i(t) \to c + \xi_i^* \gamma^*, \quad i = 1,\ldots,N, \tag{3.12}$$

asymptotically for the case when the position of the target and the scaling factor for the density of the resulting formation are constants (i.e., $c(t) \equiv c$ and $\gamma^*(t) \equiv \gamma^*$, respectively) and approximately otherwise. To this end, we introduce two assumptions on the selection of the feedback controller gain matrices in (3.8).

Assumption 3.3.1. *There exists K_1 and K_2 such that*

$$\mathcal{H} \triangleq \begin{bmatrix} A - BK_1 & -BK_2 \\ \lambda_i C & 0 \end{bmatrix}, \tag{3.13}$$

is Hurwitz for all λ_i, $i = 1, \ldots, n$, where $\lambda_i \in \operatorname{spec}(F)$ and

$$F \triangleq \mathcal{L}(\mathcal{G}) + \operatorname{diag}(G), \quad G \triangleq [\mu_1, \mu_2, \ldots, \mu_N]^{\mathrm{T}}, \tag{3.14}$$

with $\mathcal{L}(\mathcal{G}) \in \overline{\mathbb{S}}_+^{N\times N}$ and $F \in \mathbb{S}_+^{N\times N}$ by Lemmas 3.2.1 and 3.2.2, respectively.

Assumption 3.3.2. *There exists K_1 and K_2 such that*

$$\mathcal{J} \triangleq C\bar{A}^{-1}\bar{B}, \tag{3.15}$$

is invertible, where $\mathcal{J} \in \mathbb{R}^{p\times p}$,

$$\bar{A} \triangleq A - BK_1 \in \mathbb{R}^{n\times n}, \tag{3.16}$$

and

$$\bar{B} \triangleq BK_2 \in \mathbb{R}^{n\times p}. \tag{3.17}$$

Next, let the aggregated vectors be given by

$$x(t) = \left[x_1(t), x_2(t), \ldots, x_N(t)\right]^{\mathrm{T}} \in \mathbb{R}^{Nn}, \tag{3.18}$$

$$\xi(t) = \left[\xi_1(t), \xi_2(t), \ldots, \xi_N(t)\right]^{\mathrm{T}} \in \mathbb{R}^{Np}, \tag{3.19}$$

and

$$\xi(t) = \begin{bmatrix} \xi_i^* & & 0 \\ & \ddots & \\ 0 & & \xi_N^* \end{bmatrix} \begin{bmatrix} \gamma_1(t) \\ \vdots \\ \gamma_N(t) \end{bmatrix} \triangleq \psi\gamma(t). \tag{3.20}$$

Using these vectors, (3.6), (3.7), (3.9), and (3.11) can now be written in a compact form as

$$\dot{x}(t) = (\mathrm{I}_N \otimes A)x(t) + (\mathrm{I}_N \otimes B)u(t), \tag{3.21}$$

$$y(t) = (\mathrm{I}_N \otimes C)x(t), \tag{3.22}$$

$$\dot{z}(t) = (F \otimes C)x(t) - (F \otimes \mathrm{I}_p)\psi\gamma(t) - (G \otimes \mathrm{I}_p)c(t), \tag{3.23}$$

$$\dot{\gamma}(t) = -\alpha F\gamma(t) + \alpha G\gamma^*(t). \tag{3.24}$$

Now, consider the multiagent system given by (3.6), (3.7), where N agents exchange information among each other using their local measurements according to a connected, undirected graph $\mathcal{G}$. In addition, consider the distributed controller architecture given by (3.8), (3.9), and (3.11) subject to Assumptions 3.3.1 and 3.3.2. If the position of the target and the scaling factor for the density of the resulting formation are constants, then it can be shown that

$$\lim_{t\to\infty} y_i(t) = c + \xi_i^*\gamma^*, \quad i = 1,\ldots,N. \tag{3.25}$$

In other words, for the case when $c(t) \equiv c$ and $\gamma^*(t) \equiv \gamma^*$, the proposed distributed control architecture given by (3.8), (3.9), and (3.11) not only generates a desired formation but also spatially alters the density of the resulting formation[5].

Building on the above result, we next consider a more practical case when the position of the target and the scaling factor for the density of the resulting formation are time-varying with bounded time rates of change; that is,

$$\|\dot{c}(t)\|_2 \leq \beta_1, \tag{3.26}$$

$$\|\dot{\gamma}^*(t)\|_2 \leq \beta_2 \tag{3.27}$$

For this purpose, once again, consider the multiagent system given by (3.6), (3.7), where N agents exchange information among each other using their local measurements according to a connected, undirected graph $\mathcal{G}$. In addition, consider the distributed controller architecture given by (3.8), (3.9), and (3.11) subject to Assumptions 3.3.1 and 3.3.2. If the position of the target and the scaling factor for the density of the

[5]The detail proof of 3.25 is provided in Appendix A.

resulting formation are time-varying with bounded time rates of change, then it can be shown that $y_i(t)$ converges to a neighborhood of[6]

$$c(t)+\xi_i^*\gamma^*(t), \quad i=1,\ldots,N. \tag{3.28}$$

Furthermore, an ultimate bound for the distance of

$$\tilde{q}(t)=q(t)+A_q^{-1}B_q p_c(t), \tag{3.29}$$

can be computed as [115]

$$\|\tilde{q}(t)\|_2 \;\le\; 2\sqrt{\frac{\lambda_{\max}(P_q)}{\lambda_{\min}(P_q)}}\,\frac{\|P_qA_q^{-1}B_q\|_F(\beta_1+\beta_2)}{\lambda_{\min}(R_q)}, \qquad t\ge T. \tag{3.30}$$

In (3.30), if its right hand side is small, then the distance of (3.29) is small for $t \ge T$. It now can be shown that a small (3.29) implies $y_i(t)$ to stay close to

$$c(t)+\xi_i^*\gamma^*(t), \quad i=1,\ldots,N, \tag{3.31}$$

for $t \ge T$.

Finally, in addition to controlling the density of the resulting formation, one can also control its orientation by adding an additional network layer. For example, for a two dimensional formation problem, one can use the proposed controller architecture in (3.8) and (3.9) with (3.10) replaced by

$$\xi_i(t) \;\triangleq\; \gamma_i(t)R\big(\theta_i(t)\big)\xi_i^* \;\in\; \mathbb{R}^2, \tag{3.32}$$

where

$$R(\theta_i(t)) \triangleq \begin{bmatrix} \cos\theta_i(t) & -\sin\theta_i(t) \\ \sin\theta_i(t) & \cos\theta_i(t) \end{bmatrix} \;\in\; \mathbb{R}^{2\times 2}. \tag{3.33}$$

[6]The detail proof of 3.28 is provided in Appendix B.

In (3.32), $\gamma_i(t) \in \mathbb{R}$ and $\theta_i(t) \in \mathbb{R}$ are additional network layers respectively satisfying (3.11) and

$$\dot{\theta}_i(t) = -\alpha \sum_{i \sim j} \big(\theta_i(t) - \theta_j(t)\big) - \mu_i \alpha \big(\theta_i(t) - \theta^*(t)\big), \tag{3.34}$$

with $\theta^*(t) \in \mathbb{R}$ controlling the orientation of the resulting formation.

3.4 Illustrative Numerical Example

In this section, we present an illustrative numerical example to demonstrate the efficacy of the proposed distributed control architecture in Section 3.3. Specifically, consider a multiagent system with six agents exchanging information among each other using their local measurements according to a connected, undirected graph depicted in Figure 3.1, where the first agent is a capable agent (i.e., $\mu_1 = 1$ and $\mu_i = 0$, $i = 2, \ldots, 6$). The dynamics of each agent is represented by (3.6) and (3.7) with

$$A = \begin{bmatrix} 0_{3\times 3} & \mathrm{I}_3 \\ A_1 & A_2 \end{bmatrix}, \quad B = \begin{bmatrix} 0_{3\times 3} \\ \mathrm{I}_3 \end{bmatrix}, \quad C = \begin{bmatrix} \mathrm{I}_3 & 0_{3\times 3} \end{bmatrix}, \tag{3.35}$$

where

$$A_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3\omega_0^2 & 0 \\ 0 & 0 & -\omega_0^2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 2\omega_0 & 0 \\ -2\omega_0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.36}$$

and

$$\omega_0 = 0.0015. \tag{3.37}$$

Note that (3.35) and (3.36) represent the linearized equations of the relative translational dynamics, which are described by the Clohessy-Wiltshire equations [116], and

$$x_i(t) = \Big[\bar{\omega}_i^{\mathrm{T}}(t), \dot{\bar{\omega}}_i^{\mathrm{T}}(t)\Big]^{\mathrm{T}}, \tag{3.38}$$

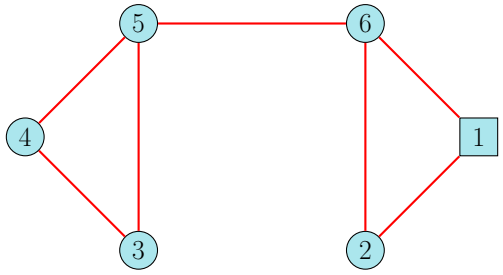


Figure 3.1: A multiagent system with six agents (square denotes the capable agent, circles denote the other agents, and solid lines denote the connected, undirected graph topology).

with $\bar{\omega}_i(t)$ being the position of agent i, $i = 1,\ldots,6$, in a three dimensional space.

In this example, all agents are subject to random initial conditions and we let $z_i(0) = 0$ and $\gamma_i(0) = 1$, $i = 1,\ldots,6$. In addition, we choose

$$K_1 = \begin{bmatrix} 25.46 & -0.005 & 0 & 15.84 & 0 & 0 \\ 0.005 & 25.46 & 0 & 0 & 15.84 & 0 \\ 0 & 0 & 25.46 & 0 & 0 & 15.84 \end{bmatrix}, \tag{3.39}$$

$$K_2 = \begin{bmatrix} 14.14 & -0.003 & 0 \\ 0.003 & 14.14 & 0 \\ 0 & 0 & 14.14 \end{bmatrix}, \tag{3.40}$$

and

$$\alpha = 5, \tag{3.41}$$

where Assumptions 3.3.1 and 3.3.2 are satisfied. For ξ_i^*, $i = 1,\ldots,6$, we choose $\xi_1^* = \begin{bmatrix} 2 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$, $\xi_2^* = \begin{bmatrix} 1 & -1 & 0 \end{bmatrix}^{\mathrm{T}}$, $\xi_3^* = \begin{bmatrix} -1 & -1 & 0 \end{bmatrix}^{\mathrm{T}}$, $\xi_4^* = \begin{bmatrix} -2 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$, $\xi_5^* = \begin{bmatrix} -1 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$, and $\xi_6^* = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$, which yields to an uniform hexagon desired formation on a two dimensional space. Finally, we let

$$c(t) = \begin{bmatrix} 0.1t & 2.5\sin(0.02t) & 0 \end{bmatrix}^{\mathrm{T}}. \tag{3.42}$$

Figure 3.2A presents the results when the scaling factor for the density of the resulting formation is $\gamma^*(t) = 0.5$ for $t \in [0,80)$ seconds, $\gamma^*(t) = 1$ for $t \in [80,160)$ seconds, and $\gamma^*(t) = 1.5$ for $t \geq 160$ seconds.

In addition, Figure 3.2B presents the results when the scaling factor for the density of the resulting formation is $\gamma^*(t) = 1.5$ for $t \in [0, 80)$ seconds, $\gamma^*(t) = 1$ for $t \in [80, 160)$ seconds, and $\gamma^*(t) = 0.5$ for $t \geq 160$ seconds. In both figures, we use a low-pass filter to smoothen the transition between $\gamma^*(t)$ changes (in order to have a bounded time rate of change of $\gamma^*(t)$). It is clear from these figures that the proposed distributed control architecture allows the capable agent to spatially alter density of the resulting formation while tracking a dynamic target of interest. Finally, the norm of the control signals for each agent is depicted in Figures 3.2C and 3.2D for the cases in Figures 3.2A and 3.2B, respectively.

3.5 Conclusion

Current distributed control methods lack information exchange infrastructures to enable spatially evolving multiagent formations without having global information exchange ability. We recently showed for multiagent systems with first-order agent dynamics how information exchange rules represented by a network having multiple layers (multiplex information networks) can be designed for enabling spatially evolving multiagent formations. This paper generalized our recent results for multiagent systems with general linear dynamics. Specifically, multiplex information networks are utilized for formation density control of multiagent systems. The proposed approach allows capable agents to spatially alter density of the resulting formation while tracking a target of interest — without requiring global information exchange ability, and hence, through local interactions. An illustrative numerical result demonstrated the efficacy of the proposed methodology.

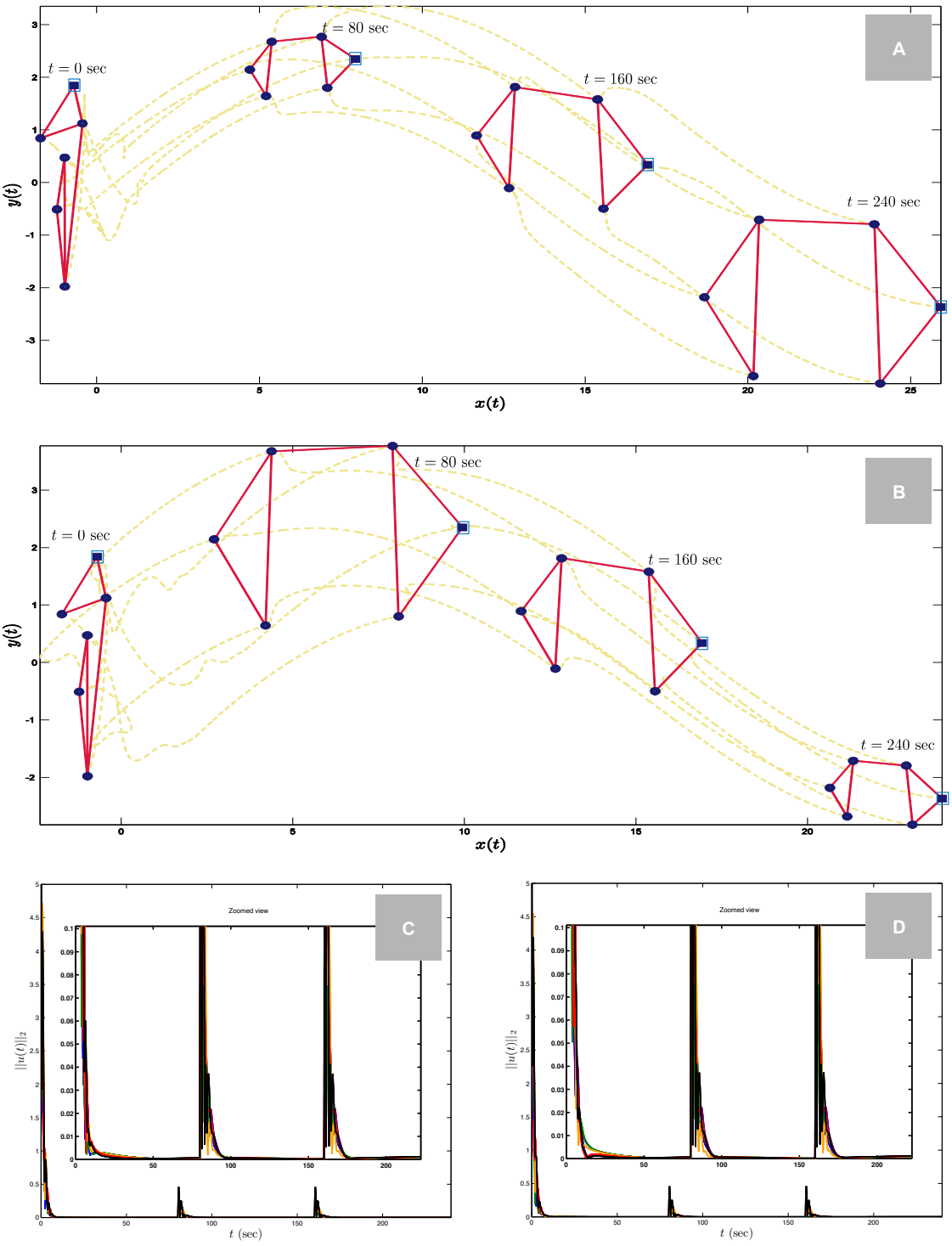


Figure 3.2: Responses of the multiagent system in Figure 3.1 with the proposed distributed control architecture for two different scaling factors for the density of the resulting formation (A and B) and respectively the norm of the control signals of agents for each case (C and D). In A and B, square denotes the capable agent, circles denote the other agents, solid lines denote the connected, undirected graph topology, and dashed lines denote trajectories of agents on a two dimensional space.

Chapter 4: Bandwidth Control of Multiagent Systems*

Increasing complexity of engineering tasks requires the spatial and temporal properties of agent teams to change in real-time. However, single-layer structure of existing distributed control algorithms does not provide the necessary flexibility to control such properties through local interactions. Motivated by this standpoint, the contribution of this paper is to make the first attempt in addressing how to develop a distributed approach for controlling the bandwidth (i.e., the temporal property) of multiagent systems. Specifically, we propose a distributed control architecture predicated on a multiplex information network having two layers for controlling the bandwidth of agent teams though local interactions, where a leader-follower algorithm is considered in the first layer (main layer) and a bandwidth distribution algorithm is considered in the second layer. The key feature of the proposed architecture is that the bandwidth commands on the second layer, which are available to the leader or leaders in the multiagent system, directly drive the bandwidth of the main layer. Both constant and time-varying bandwidth commands are considered and system-theoretic stability properties for both cases are established. Finally, illustrative numerical examples are presented to demonstrate the efficacy of the proposed architecture.

4.1 Introduction

4.1.1 Background

Distributed control of multiagent systems is an active research field with a wide array of applications in both civilian and military domains. Considering the increasing complexity of engineering tasks, the spatial and temporal properties of agent teams are required to change in real-time. To elucidate this point, consider a fleet of ground vehicles as an example that are commanded to form and maintain a formation while simultaneously tracking a dynamic target. The distance between each agent in the formation can be large under ideal conditions to maximize the sensing ability of the overall multiagent system as a whole. Yet,

*This chapter has been submitted to the 2020 American Control Conference for possible publication.

when agents pass through a narrow passage, it is then necessary for the formation to scale down (i.e., the spatial property) in real-time to fit in. In addition, depending on the speed of the dynamic target, agents need to adjust their their bandwidth (i.e., the temporal property) in real-time for maintaining a desired tracking distance with the target.

To address engineering tasks with increasing complexity, multiagent systems need new distributed methods that allow control of their spatial and temporal properties. Because, single-layer structure of existing approaches does not provide the necessary flexibility to control such properties through local interactions (e.g., see [4–9] and references therein). For example, the formation control architectures proposed in [7–9] result in a fixed formation; that is, the size and the orientation of the formation are fixed and they cannot be altered distributively once being formed. As another example, the well-known consensus and consensus-like approaches have a fixed bandwidth, which is depends on the Fiedler eigenvalue of the graph Laplacian, and it cannot be changed distributively on the fly.

Multiplex information networks is an appropriate method for altering these fixed properties in real-time. In particular, multiplex information networks describes networks with multiple information exchange layers comprising both intralayer and interlayer communication links. They have emerged in the physics and social science fields (e.g., see [10–14] and references therein). However, these fields mainly focus on studying the system-theoretic characteristics as well as the convergence of the overall network dynamics without focusing on the control design aspect. Recently, the authors of [76, 91, 112, 113] utilize multiplex information networks for spatial control of multiagent systems through local interactions. Yet, how to system-theoretically use these networks for bandwidth control of multiagent systems is still an open and important scientific question.

4.1.2 Contribution

The contribution of this paper is to make the first attempt in addressing how to develop a distributed approach for controlling the bandwidth (i.e., the temporal property) of multiagent systems. Specifically, we propose a distributed control architecture predicated on a multiplex information network having two layers for controlling the bandwidth of agent teams though local interactions, where a leader-follower algorithm is considered in the first layer (main layer) and a bandwidth distribution algorithm is considered in the second layer. Here, we note that although a leader-follower algorithm is considered in the main layer, the presented results can be readily applied to consensus algorithms, formation algorithms, and containment algorithms,

to name but a few examples, for controlling the bandwidths of these algorithms in the main layer. The key feature of the proposed architecture is that the bandwidth commands on the second layer, which are available to the leader or leaders in the multiagent system, directly drive the bandwidth of the main layer. Both constant and time-varying bandwidth commands are considered and system-theoretic stability properties for both cases are established. Finally, illustrative numerical examples are presented to demonstrate the efficacy of the proposed architecture.

The organization of this paper is as follows. In Section 4.2, we present the necessary mathematical preliminaries. The proposed architecture and its system-theoretic stability properties are given in Section 4.3 for constant bandwidth commands case and in Section 4.4 for time-varying bandwidth commands case. Section 4.5 then presents illustrative numerical examples and Section 4.6 finally summarizes the concluding remarks.

4.2 Preliminaries

In this paper, $\mathbb{R}$ stands for the set of real numbers, $\mathbb{R}^n$ stands for the set of $n \times 1$ real column vectors, $\mathbb{R}_+^{n\times n}$ (resp., $\overline{\mathbb{R}}_+^{n\times n}$) stands for the set of $n \times n$ positive-definite (resp., positive semi-definite) real matrices, $\mathbf{1}_n$ stands for the $n \times 1$ vector of all ones, and I_n stands for the $n \times n$ identity matrix. We also use $(\cdot)^{\mathrm{T}}$ for transpose, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ respectively for the minimum and maximum eigenvalue of a matrix A, $\mathrm{diag}(a)$ for diagonal matrix with vector a on its diagonal, $[x]_i$ for the entry of vector x on the i-th row, and A_{ij} for the entry of matrix A on the i-th row and j-th column.

We now recall several graph-theoretical notions for completeness (e.g., see [5] and [92] for details). Specifically, an undirected graph $\mathcal{G}$ is defined by a set $\mathcal{V}_\mathcal{G} = \{1,\ldots,N\}$ of nodes and a set $\mathcal{E}_\mathcal{G} \subset \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ of edges. If $(i,j) \in \mathcal{E}_\mathcal{G}$, then the nodes i and j are neighbors and the neighboring relation is indicated with $i \sim j$. The number of agent i's neighbors is its degree and denoted as d_i. The degree matrix of a graph $\mathcal{G}$, $\mathcal{D}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is then defined by $\mathcal{D}(\mathcal{G}) \triangleq \mathrm{diag}(d)$, where $d = [d_1,\ldots,d_N]^{\mathrm{T}}$. In addition, a path $i_0 i_1 \ldots i_L$ is a finite sequence of nodes such that $i_{k-1} \sim i_k$, $k = 1,\ldots,L$, and a graph $\mathcal{G}$ is called connected when there exists a path between any pair of distinct nodes. The adjacency matrix of a graph $\mathcal{G}$, $\mathcal{A}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is also defined by $[\mathcal{A}(\mathcal{G})]_{ij} = 1$ when $(i,j) \in \mathcal{E}_\mathcal{G}$ and $[\mathcal{A}(\mathcal{G})]_{ij} = 0$ otherwise. Finally, the graph Laplacian matrix, $\mathcal{L}(\mathcal{G}) \in \overline{\mathbb{R}}_+^{N\times N}$, is defined by $\mathcal{L}(\mathcal{G}) \triangleq \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$. For the results of this paper, we consider a connected, undirected graph $\mathcal{G}$.

Lemma 4.2.1 (Proposition 9.4.9 of [117]). *Let $A \in \mathbb{R}^{n\times m}$. Then, $\|A\|_2 = \sigma_{\max}(A) = \sqrt{\lambda_{\max}(A^{\mathrm{T}}A)}$, where $\sigma_{\max}(A)$ denotes the largest singular vale of A.*

From Lemma 4.2.1, it follows that $\|A\|_2 = |\lambda_{\max}(A)|$ when A is symmetric. In addition, if $A = \mathrm{diag}(a)$ with $a = [a_1, \ldots, a_N]^{\mathrm{T}}$, then $\|A\|_2 = \max_i |a_i|$.

4.3 Constant Bandwidth Control

4.3.1 Proposed Architecture

Consider a multiagent system with N agents exchanging information according to a connected, undirected graph $\mathcal{G}$ and operating under the dynamical structure given by

$$\dot{x}_i(t) = \alpha_i(t)\left(-\sum_{i\sim j}(x_i(t)-x_j(t)) - k_i(x_i(t)-c(t))\right), \quad x_i(0) = x_{i0}, \tag{4.1}$$

$$\dot{\alpha}_i(t) = -\beta\left(\sum_{i\sim j}(\alpha_i(t)-\alpha_j(t)) + k_i(\alpha_i(t)-\alpha_0)\right), \quad \alpha_i(0) = \alpha_{i0}, \tag{4.2}$$

where $x_i(t) \in \mathbb{R}$ and $\alpha_i(t) \in \mathbb{R}$ respectively denote the state and the bandwidth of agent i, $i = 1, \ldots, N$, $c(t) \in \mathbb{R}$ is the time-varying tracking command with bounded time rate of change (i.e., $|\dot{c}(t)| \leq \bar{c}$ with $\bar{c} \in \mathbb{R}_+$), $\alpha_0 \in \mathbb{R}_+$ denote a constant bandwidth command, and $\beta \in \mathbb{R}_+$ denote a constant gain. While it does not change the theoretical results of this paper, α_{i0} for all $i = 1, \ldots, N$ should be selected as positive from a practical standpoint. In addition, we consider that the network has at least one leader, where $k_i = 1$ when agent i is a leader and $k_i = 0$ otherwise. In this setting, the tracking command $c(t)$ and the bandwidth command α_0 is only available to the leader agent(s).

4.3.2 System-Theoretic Analysis

We first define the state and bandwidth errors as

$$\tilde{x}_i(t) \triangleq x_i(t) - c(t), \tag{4.3}$$

$$\tilde{\alpha}_i(t) \triangleq \alpha_i(t) - \alpha(0). \tag{4.4}$$

By taking the time derivative of (4.3), we obtain

$$\dot{\tilde{x}}_i(t) = -(\tilde{\alpha}_i(t)+\alpha_0)\left(\sum_{i\sim j}(\tilde{x}_i(t)-\tilde{x}_j(t)) + k_i\tilde{x}_i(t)\right) - \dot{c}(t), \quad \tilde{x}_i(0) = \tilde{x}_{i0}. \tag{4.5}$$

Similarly, taking the time derivative of (4.4) yields

$$\dot{\tilde{\alpha}}_i(t) = -\beta\left(\sum_{i\sim j}(\tilde{\alpha}_i(t)-\tilde{\alpha}_j(t))+k_i\tilde{\alpha}_i(t)\right), \quad \tilde{\alpha}_i(0)=\tilde{\alpha}_{i0}. \tag{4.6}$$

Next, let $\tilde{x}(t) \triangleq [\tilde{x}_1(t),\ldots,\tilde{x}_N(t)]^{\mathrm{T}}$ and $\tilde{\alpha}(t) \triangleq [\tilde{\alpha}_1(t),\ldots,\tilde{\alpha}_N(t)]^{\mathrm{T}}$ be the aggregated vectors. Then, (4.5) and (4.6) can be written in the compact form as

$$\begin{aligned}
\dot{\tilde{x}}(t) &= -\operatorname{diag}(\tilde{\alpha}(t)+\alpha_0\mathbf{1}_N)\left(\mathcal{L}(\mathcal{G})\tilde{x}(t)+K\tilde{x}(t)\right)-\mathbf{1}_N\dot{c}(t) \\
&= -\operatorname{diag}(\tilde{\alpha}(t)+\alpha_0\mathbf{1}_N)F\tilde{x}(t)-\mathbf{1}_N\dot{c}(t), \quad \tilde{x}(0)=\tilde{x}_0, \qquad (4.7)\\
\dot{\tilde{\alpha}}(t) &= -\beta F\tilde{\alpha}(t), \quad \tilde{\alpha}(0)=\tilde{\alpha}_0, \qquad (4.8)
\end{aligned}$$

where $F \triangleq \mathcal{L}(\mathcal{G})+K \in \mathbb{R}^{N\times N}$ with $K \triangleq \operatorname{diag}([k_1,\ldots,k_N])$. Here, we note that F is a positive-definite matrix (e.g., see Lemma 3.3 of [16]). We are now ready to state the first result of this paper.

Theorem 4.3.1. *Consider the multiagent system given by (4.1) and (4.2) with N agents exchanging information according to a connected, undirected graph $\mathcal{G}$. Then, the closed-loop error dynamics of the overall network system given by (4.7) and (4.8) is input-to-state stable.*

Proof. We start with the explicit solution of (4.8) given by

$$\tilde{\alpha}(t) = e^{-\beta Ft}\tilde{\alpha}(0). \tag{4.9}$$

Note that since F is a symmetric matrix, it can be decomposed as $F \triangleq U\Lambda U^{\mathrm{T}}$ (e.g., see Theorem 2.5.6 of [118]), where $\Lambda = \operatorname{diag}(\lambda_i(F)) \in \mathbb{R}^{N\times N}$ with $\lambda_i(F) > 0 \;\forall\; i = 1,\ldots,N$ being the eigenvalues of F and $U \triangleq [u_1 \ldots u_N] \in \mathbb{R}^{N\times N}$ denotes an orthonormal matrix with $u_i \in \mathbb{R}^N$, $\|u_i\|_2^2 = 1$, being the eigenvector corresponding to $\lambda_i(F)$. Therefore, we have

$$\begin{aligned}
\|e^{-\beta Ft}\|_2 &= \|e^{-\beta U\Lambda U^{\mathrm{T}}t}\|_2 = \|Ue^{-\beta\Lambda t}U^{\mathrm{T}}\|_2 \\
&\leq \|U\|_2\|U^{\mathrm{T}}\|_2\|e^{-\beta\Lambda t}\|_2 \\
&\leq ke^{-\beta\bar{\lambda}t}, \qquad (4.10)
\end{aligned}$$

for some positive constant k and $\bar{\lambda}$. We now substitute (4.9) back to (4.7) and obtain

$$\dot{\tilde{x}}(t) \quad = \quad -\operatorname{diag}\left(e^{-\beta F t}\tilde{\alpha}(0)+\alpha_0 \mathbf{1}_N\right) F\tilde{x}(t) - \mathbf{1}_N \dot{c}(t), \quad \tilde{x}(0)=\tilde{x}_0. \tag{4.11}$$

Let $A(t) \triangleq -\operatorname{diag}\left(e^{-\beta F t}\tilde{\alpha}(0)+\alpha_0 \mathbf{1}_N\right) F$. Then, (4.11) can be rewritten as

$$\dot{\tilde{x}}(t) \quad = \quad A(t)\tilde{x} - \mathbf{1}_N \dot{c}(t), \quad \tilde{x}(0)=\tilde{x}_0. \tag{4.12}$$

Since the term $-\mathbf{1}_N \dot{c}(t)$ in (4.12) is bounded by assumption, one can consider it as the bounded input to the system. Furthermore, by Lemma 4.6 of [119], if the origin of the unforced system (i.e., $\dot{\tilde{x}}(t) = A(t)\tilde{x}(t)$) is globally exponentially stable, then one can conclude that the system (4.12) is input-to-state stable. Therefore, we now prove that the origin of $\dot{\tilde{x}}(t) = A(t)\tilde{x}(t)$ is globally exponentially stable.

To this end, the unforced system can be first expanded into

$$\begin{aligned} \dot{\tilde{x}}(t) &= A(t)\tilde{x}(t) \\ &= -\alpha_0 F\tilde{x}(t) - \operatorname{diag}\left(e^{-\beta F t}\tilde{\alpha}(0)\right) F\tilde{x}(t). \end{aligned} \tag{4.13}$$

Consider next the Lyapunov function candidate given by

$$V(\tilde{x}) = \frac{1}{2}\tilde{x}^{\mathrm{T}}(t)\tilde{x}(t). \tag{4.14}$$

Note that $V(0)=0$ and $V(\tilde{x})>0$ for all $\tilde{x}\neq 0$. The time derivative of (4.14) along the trajectories of (4.13) is given by

$$\begin{aligned} \dot{V}(\tilde{x}) &= -\alpha_0 \tilde{x}^{\mathrm{T}}(t)F\tilde{x}(t) - \tilde{x}^{\mathrm{T}}(t)\operatorname{diag}\left(e^{-\beta F t}\tilde{\alpha}(0)\right) F\tilde{x}(t) \\ &\leq -\alpha_0 \lambda_{\min}(F)\|\tilde{x}(t)\|_2^2 + k e^{-\beta\tilde{\lambda}t}\|\alpha(0)\|_2 \|F\|_2 \|\tilde{x}(t)\|_2^2 \\ &= -\underbrace{2\left(\alpha_0\lambda_{\min}(F) - k e^{-\beta\tilde{\lambda}t}\lambda_{\max}(F)\|\alpha(0)\|_2\right)}_{B(t)} \frac{\|\tilde{x}(t)\|_2^2}{2} \\ &= -B(t)V(\tilde{x}), \end{aligned} \tag{4.15}$$

where the second inequality comes from (4.10) and the third equality comes directly from Lemma 4.2.1. Defining $H(t) \triangleq e^{\int_0^t B(s)ds}V(\tilde{x})$ and taking its time derivative yields

$$\begin{aligned} \dot{H}(t) &= B(t)e^{\int_0^t B(s)ds}V(\tilde{x}) + e^{\int_0^t B(s)ds}\dot{V}(\tilde{x}) \\ &\leq B(t)e^{\int_0^t B(s)ds}V(\tilde{x}) - e^{\int_0^t B(s)ds}B(t)V(\tilde{x}) = 0. \end{aligned} \tag{4.16}$$

As a result, $H(t) \leq H(0)$. Equivalently,

$$V(\tilde{x}(t)) \leq \quad e^{-\int_0^t B(s)ds} V(\tilde{x}(0)). \tag{4.17}$$

Substituting (4.14) into (4.17) and expanding $B(t)$, we obtain

$$\begin{aligned} \|\tilde{x}(t)\|_2^2 \leq \quad & e^{-\int_0^t 2\left(\alpha_0 \lambda_{\min}(F) - k e^{-\beta\bar{\lambda}s}\lambda_{\max}(F)\|\alpha(0)\|_2\right)ds} \|\tilde{x}(0)\|_2^2 \\ = \quad & e^{-2\alpha_0\lambda_{\min}(F)t + 2k\lambda_{\max}(F)\|\alpha(0)\|_2 \int_0^t e^{-\beta\bar{\lambda}s}ds} \|\tilde{x}(0)\|_2^2. \end{aligned} \tag{4.18}$$

Taking square root of both sides of (4.18) yields

$$\|\tilde{x}(t)\|_2 \leq \quad e^{-\alpha_0\lambda_{\min}(F)t + k\lambda_{\max}(F)\|\alpha(0)\|_2 \int_0^t e^{-\beta\bar{\lambda}s}ds} \|\tilde{x}(0)\|_2. \tag{4.19}$$

Note that since

$$\int_0^t e^{-\beta\bar{\lambda}s}ds \quad = \quad \frac{1}{\beta\bar{\lambda}} - \frac{e^{-\beta\bar{\lambda}t}}{\beta\bar{\lambda}} \leq \frac{1}{\beta\bar{\lambda}}, \tag{4.20}$$

(4.19) becomes

$$\begin{aligned} \|\tilde{x}(t)\|_2 \leq \quad & e^{-\alpha_0\lambda_{\min}(F)t + \frac{k\lambda_{\max}(F)\|\alpha(0)\|_2}{\beta\bar{\lambda}}} \|\tilde{x}(0)\|_2 \\ = \quad & e^{-\alpha_0\lambda_{\min}(F)t} \left(e^{\eta}\|\tilde{x}(0)\|_2\right), \end{aligned} \tag{4.21}$$

where $\eta \triangleq \frac{k\lambda_{\max}(F)\|\alpha(0)\|_2}{\beta\bar{\lambda}}$. From (4.21), we can conclude that the origin of the unforced system (4.13) is globally exponentially stable. As a consequence, the system (4.12) is input-to-state stable; hence, the closed-loop error dynamics of the overall network system given by (4.7) is input-to-state stable. ■

Remark 4.3.1. *Theorem 4.3.1 indicates that when the tracking command is a constant $c(t) \triangleq c$ or when $\dot{c}(t)$ approaches 0 as $t \to \infty$, then the closed-loop error $\tilde{x}(t)$ will also converge to 0 as $t \to \infty$. In addition, when $c(t)$ is time-varying, $\tilde{x}(t)$ is ultimately bounded by a class $\mathcal{K}$ function of* $\sup_{0\leq\tau\leq t} \| - \mathbf{1}_N \dot{c}(\tau)\|_2 = \sqrt{N}\bar{c}$.

Remark 4.3.2. *From (4.21), it can be seen that the performance of the overall network system depends on the command bandwidth α_0, the design parameter β, and the structure of network via $\lambda_{\max}(F), \lambda_{\min}(F), \bar{\lambda}$ and k. Therefore, one can use this result to judiciously choose the design parameters.*

4.4 Time-varying Bandwidth Control

In this section, we generalize the result of Section 4.3 to the case where the bandwidth command is time-varying. For this purpose, we consider the same dynamical structure given by (4.1) and (4.2), where $\alpha_0(t) \in \mathbb{R}_+$ is now a time-varying bandwidth command with a bounded time rate of change such that $|\dot{\alpha}_0(t)| \leq \bar{\alpha}_0 \in \mathbb{R}_+$ and satisfies the condition $0 < \underline{a} \leq \alpha_0(t) \leq \bar{a}$ with $\underline{a}, \bar{a} \in \mathbb{R}_+$. Along the lines of the mathematical discussion given in Section 4.3.2, we obtain the compact from of the close-loop error dynamics given by

$$\dot{\tilde{x}}(t) = -\operatorname{diag}(\tilde{\alpha}(t) + \alpha_0(t)\mathbf{1}_N)F\tilde{x}(t) - \mathbf{1}_N\dot{c}(t), \quad \tilde{x}(0) = \tilde{x}_0, \tag{4.22}$$

$$\dot{\tilde{\alpha}}(t) = -\beta F\tilde{\alpha}(t) - \mathbf{1}_N\dot{\alpha}_0(t), \quad \tilde{\alpha}(0) = \tilde{\alpha}_0. \tag{4.23}$$

We are now ready to state the second result of this paper.

Theorem 4.4.1. *Consider the multiagent system given by (4.1) and (4.2) with N agents exchanging information according to a connected, undirected graph $\mathcal{G}$. If the gain β is sufficiently large, then under a time-varying bandwidth command with a bounded time rate of change the closed-loop error dynamics of the overall network system given by (4.22) and (4.23) is uniformly bounded.*

Proof. Once again, we start with the explicit solution of (4.23) given by

$$\tilde{\alpha}(t) = e^{-\beta Ft}\tilde{\alpha}(0) + \int_0^t e^{-\beta F(t-s)}\left(-\mathbf{1}_N\dot{\alpha}_0(t)\right)ds. \tag{4.24}$$

Substituting (4.24) to (4.22) yields

$$\dot{\tilde{x}}(t) = -\operatorname{diag}\left(e^{-\beta Ft}\tilde{\alpha}(0) + \alpha_0(t)\mathbf{1}_N\right)F\tilde{x}(t) - \mathbf{1}_N\dot{c}(t) + \operatorname{diag}\left(\int_0^t e^{-\beta F(t-s)}\left(\mathbf{1}_N\dot{\alpha}_0(t)\right)ds\right)F\tilde{x}(t), \quad \tilde{x}(0) = \tilde{x}_0. \tag{4.25}$$

In order to analyze the system given by (4.25), we first analyze the following system

$$\begin{aligned}\dot{\tilde{x}}(t) &= -\operatorname{diag}\left(e^{-\beta Ft}\tilde{\alpha}(0) + \alpha_0(t)\mathbf{1}_N\right)F\tilde{x}(t) \\ &= -\alpha_0(t)F\tilde{x}(t) - \operatorname{diag}\left(e^{-\beta Ft}\tilde{\alpha}(0)\right)F\tilde{x}(t), \quad \tilde{x}(0) = \tilde{x}_0.\end{aligned} \tag{4.26}$$

To this end, consider the Lyapunov function candidate given by (4.14). By taking its time derivative along the trajectory of (4.26), we obtain

$$\begin{aligned}\dot{V}(\tilde{x}) &= -\alpha_0(t)\tilde{x}^{\mathrm{T}}F\tilde{x}(t) - \tilde{x}^{\mathrm{T}}\operatorname{diag}\left(e^{-\beta F t}\tilde{\alpha}(0)\right)F\tilde{x}(t) \\ &\leq -\underline{a}\tilde{x}^{\mathrm{T}}F\tilde{x}(t) - \tilde{x}^{\mathrm{T}}\operatorname{diag}\left(e^{-\beta F t}\tilde{\alpha}(0)\right)F\tilde{x}(t). \end{aligned} \tag{4.27}$$

From this point, (4.27) can be brought to the form of (4.15) and then with a similar analysis as in proof of Theorem 4.3.1 one can conclude that the origin of the system given by (4.26) is globally exponentially stable.

By the Converse Lyapunov theorem (e.g., see Theorem 4.14 of [119]), next note that there exists a continuously differentiable function $V(\tilde{x})$ that satisfies

$$c_1\|\tilde{x}(t)\|_2^2 \leq V(\tilde{x}) \leq \quad c_2\|\tilde{x}(t)\|_2^2, \tag{4.28}$$

$$\tfrac{\partial V}{\partial x}\left(-\operatorname{diag}\left(e^{-\beta F t}\tilde{\alpha}(0)+\alpha_0(t)\mathbf{1}_N\right)F\tilde{x}(t)\right) \leq \quad -c_3\|\tilde{x}(t)\|_2^2, \tag{4.29}$$

$$\|\frac{\partial V}{\partial x}\|_2 \leq \quad c_4\|\tilde{x}(t)\|_2, \tag{4.30}$$

where c_1, c_2, c_3, and c_4 are positive constants. We now utilize the above function $V(\tilde{x})$ satisfying (4.28), (4.29), and (4.30) as the Lyapunov function candidate and take its time derivative with respect to the trajectory of (4.25). Mathematically speaking, we have

$$\begin{aligned}\dot{V}(\tilde{x}) &= \frac{\partial V}{\partial x}\Big[-\operatorname{diag}\left(e^{-\beta F t}\tilde{\alpha}(0)+\alpha_0(t)\mathbf{1}_N\right)F\tilde{x}(t) - \mathbf{1}_N\dot{c}(t) + \operatorname{diag}\left(\textstyle\int_0^t e^{-\beta F(t-s)}\left(\mathbf{1}_N\dot{\alpha}_0(t)\right)ds\right)F\tilde{x}(t)\Big] \\ &\leq -c_3\|\tilde{x}(t)\|_2^2 + c_4\|\tilde{x}(t)\|_2\|\mathbf{1}_N\bar{c}\|_2 + c_4\|\tilde{x}(t)\|_2\|\operatorname{diag}\left(\textstyle\int_0^t e^{-\beta F(t-s)}\left(\mathbf{1}_N\dot{\alpha}_0(t)\right)ds\right)\|_2\|F\|_2\|\tilde{x}(t)\|_2 \\ &= -c_3\|\tilde{x}(t)\|_2^2 + c_4\|\tilde{x}(t)\|_2\|\mathbf{1}_N\bar{c}\|_2 + c_4\lambda_{\max}(F)\|\operatorname{diag}\left(\textstyle\int_0^t e^{-\beta F(t-s)}\left(\mathbf{1}_N\dot{\alpha}_0(t)\right)ds\right)\|_2\|\tilde{x}(t)\|_2^2, \end{aligned} \tag{4.31}$$

where the second inequality comes from (4.29) and (4.30). Note also that

$$\begin{aligned}\|\operatorname{diag}\left(\int_0^t e^{-\beta F(t-s)}\left(\mathbf{1}_N\dot{\alpha}_0(t)\right)ds\right)\|_2 &\leq \|\int_0^t e^{-\beta F(t-s)}\left(\mathbf{1}_N\dot{\alpha}_0(t)\right)ds\|_2 \\ &\leq \int_0^t \|e^{-\beta F(t-s)}\|_2\|\mathbf{1}_N\bar{\alpha}_0\|_2 ds \\ &\leq \int_0^t k e^{-\beta\bar{\lambda}(t-s)}(\sqrt{N}\bar{\alpha}_0)ds \\ &\leq \frac{k\sqrt{N}\bar{\alpha}_0}{\beta\bar{\lambda}} \end{aligned} \tag{4.32}$$

where the third inequality comes from (4.10). Therefore, (4.31) becomes

$$\dot{V}(\cdot) \leq \quad -(c_3 - \tfrac{c_4\lambda_{\max}(F)k\sqrt{N}\bar{\alpha}_0}{\beta\bar{\lambda}})\|\tilde{x}(t)\|_2^2 + c_4\sqrt{N}\bar{c}\|\tilde{x}(t)\|_2 \tag{4.33}$$

From here we note that if β is designed such that

$$\beta > \frac{c_4\lambda_{\max}(F)k\sqrt{N}\bar{\alpha}_0}{c_3\bar{\lambda}}, \tag{4.34}$$

then $\rho \triangleq (c_3 - \frac{c_4\lambda_{\max}(F)k\sqrt{N}\bar{\alpha}_0}{\beta\bar{\lambda}}) > 0$. In addition, let $\phi \triangleq \frac{c_4\sqrt{N}\bar{c}}{\rho}$. Then, $\dot{V}(\cdot) \leq 0$ outside the compact set given by $\mathcal{S} \triangleq \{\tilde{x}(t) : \tilde{x}(t) < \phi\}$. As a consequence, the closed-loop error dynamics of the overall network system given by (4.22) and (4.23) is uniformly bounded. ■

Remark 4.4.1. *Similar to the discussion in Remark 4.3.2, the inequality given by (4.34) suggests an appropriate choice for the design gain β depending on the network size and structure (via the parameters $\sqrt{N}, \lambda_{\max}(F), \bar{\lambda}$) and the time rate of change of $\alpha_0(t)$ (via $\bar{\alpha}_0$).*

4.5 Illustrative Numerical Examples

In this section, we demonstrate the efficacy of the proposed distributed bandwidth control architecture through two numerical examples, where the first one considers a one-dimensional (1D) application case and the second one considers a two-dimensional (2D) application case. For both cases, we consider a group of 5 agents subject to a connected, undirected ring graph with agent 3 being the leader.

4.5.1 Example 1

This example focuses on the 1D application case. Agents are subject to the initial conditions $x(0) = [7;5;1;-2;-4]$ and $\alpha(0) = [0.5;0.8;1.5;1.25;2]$. The design gain β is set to 2. The simulation time is set to 72 seconds. In addition, for $t \in [0,24)$, the command bandwidth is set to $\alpha_0 = 2$; for $t \in [24,48)$, $\alpha_0 = 20$; and for $t \in [48,72]$, $\alpha_0 = 5$. We consider the same setup for two different tracking commands.

For the first scenario, the tracking command is a square wave such that for $t \in [0,18)$ and $t \in [36,54)$, $c(t) = 5$; for $t \in [18,36)$ and $t \in [54,72)$, $c(t) = 0$. The performance of the overall network system under the proposed control architecture is depicted in Figure 4.1. Specifically, during the first 24 seconds, agents slowly converge to the tracking command. During $t \in [24,48)$, since the bandwidth command α_0 increases,

agents react and converge to the tracking command $c(t)$ faster as depicted in the top plot of Figure 4.1. For the last 24 seconds, the bandwidth command α_0 is reduced to 5 and, as expected, the transient response is faster than when $\alpha_0 = 1$ but slower than when $\alpha_0 = 20$. In addition, the bottom plot of Figure 4.1 shows the convergence of agents' bandwidth parameters $\alpha_i(t)$ to the command bandwidth α_0.

For the second scenario, the tracking command is a sinusoidal wave depicted by $c(t) = 3\sin(0.25t)$. The performance of the overall network system under the proposed control architecture is depicted in Figure 4.2. In particular, during the first 24 seconds, agents' states lag well behind the tracking command. During $t \in [24, 48)$, with the command bandwidth $\alpha_0 = 20$, agents can closely follow the tracking command $c(t)$

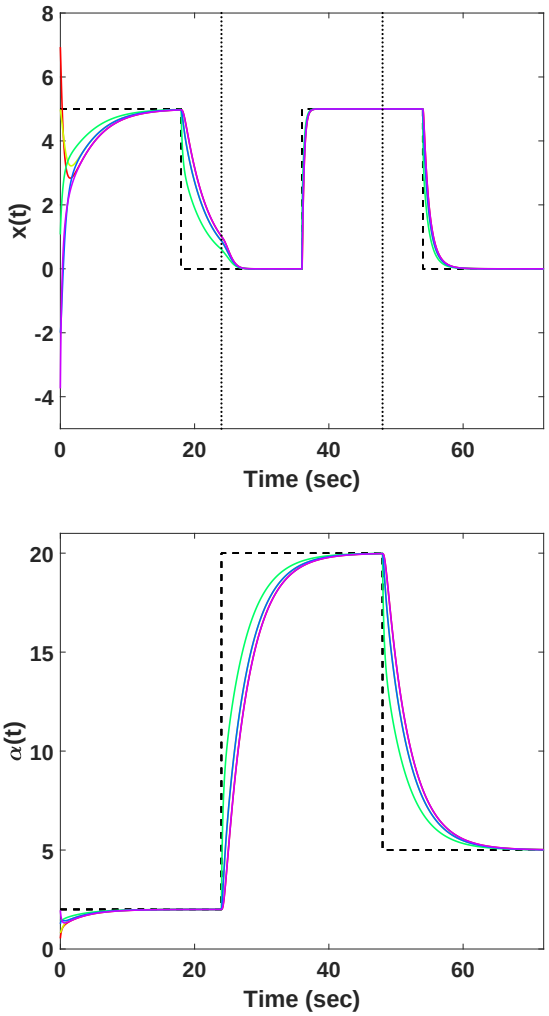


Figure 4.1: The evolution of the agents' state $x(t)$ (top) and the bandwidth parameter $\alpha(t)$ (bottom) for a square wave tracking command $c(t)$. The dotted lines denote the time when the command bandwidth is changed.

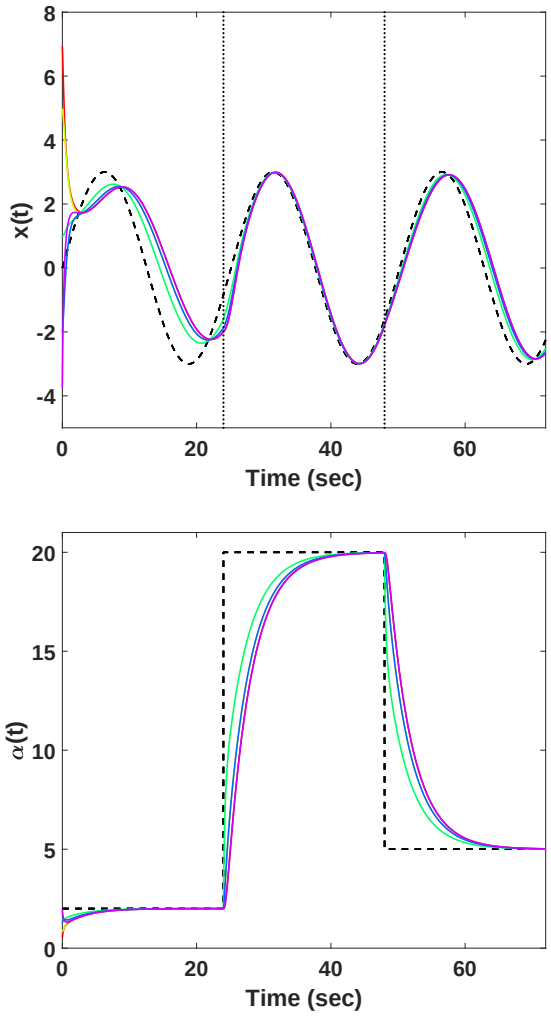


Figure 4.2: The evolution of the agents' state $x(t)$ (top) and the bandwidth parameter $\alpha(t)$ (bottom) for a sinusoidal wave tracking command $c(t) = 3\sin(0.25t)$. The dotted lines denote the time when the command bandwidth is changed.

as depicted in the top plot of Figure 4.2. For the last 24 seconds, when the command bandwidth is set to $\alpha_0 = 5$, the performance degrades as compared to the case when $\alpha_0 = 20$ but is still better than the case when $\alpha_0 = 2$. Once again, the bottom plot of Figure 4.2 shows the convergence of agents' bandwidth parameters $\alpha_i(t)$ to the command bandwidth $\alpha_0(t)$.

4.5.2 Example 2

This example focuses on the 2D application case. Agents are initially located at $(-1.5;1), (-1;1), (-1;-2), (-2;-1), (-2;0)$ and subject to $\alpha(0) = [0.5;0.8;1.5;1.25;2]$. The design gain β is set to 2. The

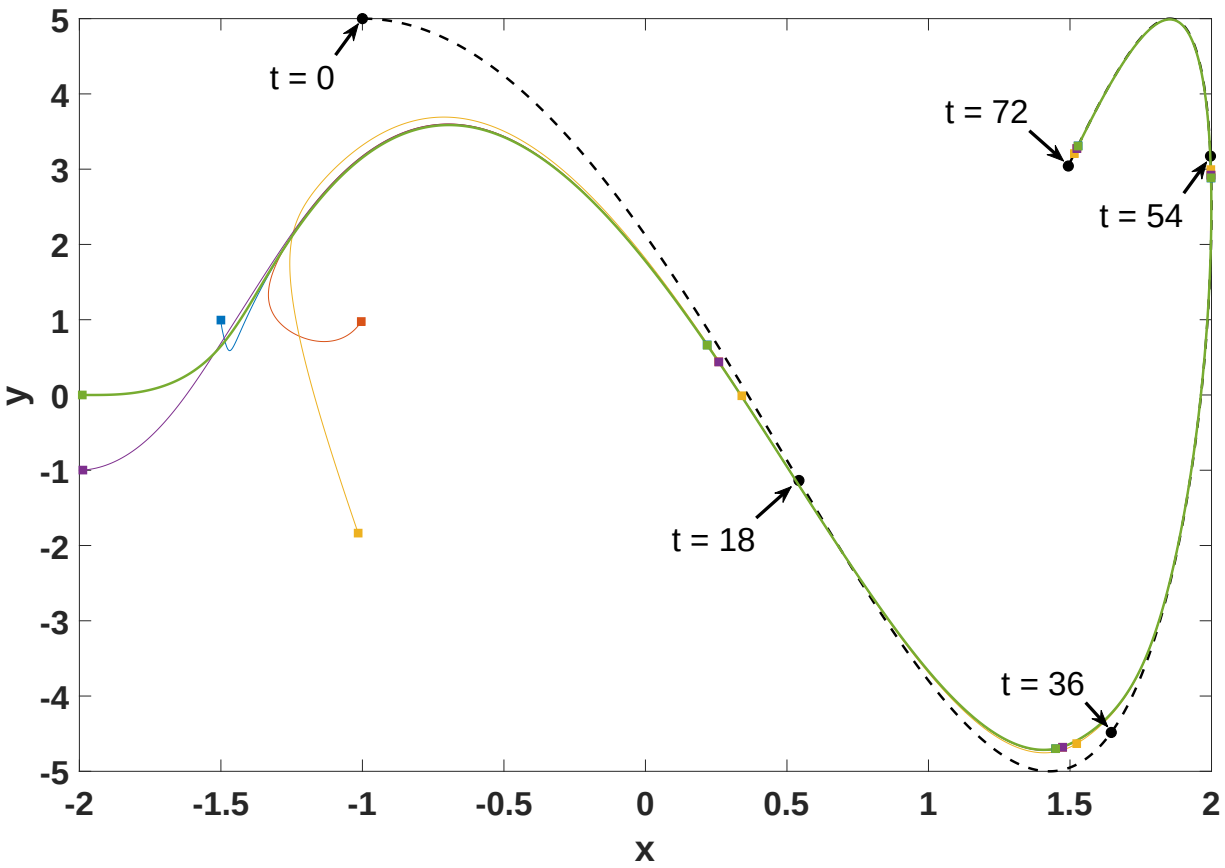


Figure 4.3: Response of the multiagent system with the proposed distributed control architecture for two different bandwidth commands. Circles denote the position of the target, squares denote positions of agents at some specific time instants, dashed line denotes the target's trajectory, and solid lines denote the trajectories of agents.

target's trajectory is depicted by $(c_x, c_y) = (-1+3\sin(0.03t), 5\cos(0.1t))$. The simulation time is set to 72 seconds. In addition, for $t \in [0, 36)$, the command bandwidth is set to $\alpha_0 = 2$; for $t \in [36, 72)$, $\alpha_0 = 10$.

The performance of the overall network system under the proposed control architecture is depicted in Figure 4.3, where circles denote the position of the target and square denotes positions of agents at some specific time instants. In particular, during the first 36 seconds, agents converge and then track the target. It can be seen that during this time period, the agents are lagged behind the target due to low bandwidth. For the last 36 seconds, when the bandwidth is increased, agents can closely track the target, even at a shape turn.

4.6 Conclusion

In this paper, we developed a distributed control architecture predicated on a multiplex information network having two layers for controlling the bandwidth (i.e., the temporal property) of agent teams though local interactions. Specifically, the first layer (main layer) considered a leader-follower algorithm and the second layer considered a bandwidth distribution algorithm, where the bandwidth commands on the second layer that were available to the leader or leaders in the multiagent system directly drove the bandwidth of the main layer. Both constant and time-varying bandwidth commands were considered and system-theoretic stability properties for both cases were established. In addition to our theoretical findings, illustrative numerical examples demonstrated the efficacy of the proposed architecture for controlling the bandwidth of multiagent systems. Future research can consider employing additional network layers to control not only temporal but also spatial properties of multiagent systems and generalizations to agent teams with high-order dynamics.

Chapter 5: On New Laplacian Matrix with a User-Assigned Nullspace in Distributed Control of Multiagent Systems*

The common denominator of most notable distributed control results is that they utilize the benchmark consensus algorithm, which is built on the well-known Laplacian matrix whose nullspace spans the vector of ones. Since this algorithm is the key building block for a wide array of existing distributed control architectures, extensions of this algorithms are also predicated on this Laplacian matrix. Motivated by this standpoint, this paper explores how one can generalize the Laplacian nullspace, which can span any vector with positive elements, to pave the way for composing complex cooperative behaviors in multiagent systems. Specifically, we introduce a new Laplacian matrix for undirected and connected graphs that generalizes the well-known, standard Laplacian matrix, where it is based on a desired, user-assigned nullspace. We first give the mathematical definition of this Laplacian matrix and show that it inherits some fundamental properties of the standard Laplacian matrix. We then present distributed control architectures for convergence to the desired nullspace and for convergence to a specific vector within that nullspace. Finally, an application of the proposed Laplacian matrix to formation tracking and scaling problem is given. To complement our theoretical results, we also present several numerical examples.

5.1 Introduction

Predicated on the recent advances in technology, multiagent systems have become an active research field during the last two decades. These systems have an important potential to impact a wide array of applications in civilian and military domains such as surveillance, reconnaissance, ground and air traffic management, payload and passenger transportation, task assignment, rapid internet delivery, and emergency response; to name but a few examples (e.g., see [120–124]). In such applications, agents (e.g., aerial, ground, water, and underwater vehicles) with information exchange ability are required to collaborate and coordinate with each other for accomplishing given group tasks as a team. Since global information exchange between

*This chapter has been submitted to the 2020 American Control Conference for possible publication.

agents is not preferred due to energy and security constraints, most state-of-the-art results now focus on the development of distributed control architectures that allow agents to exchange only local information with each other toward given group tasks. The common denominator of most notable distributed control results is that they utilize the benchmark consensus algorithm, which is built on the well-known Laplacian matrix whose nullspace spans the vector of ones (e.g., see [5, 8, 15–25]).

To elucidate the last sentence of the above paragraph, consider the benchmark consensus algorithm over undirected and connected graphs with scalar integrator dynamics given by $\dot{x}_i(t) = -\sum_{i\sim j}\big(x_i(t) - x_j(t)\big)$, where $x_i(t)$ denotes the state of agent i, $i = 1,\ldots,N$, and $i \sim j$ indicates that agents i and j are neighbors. Defining $x(t) \triangleq [x_1(t),\ldots,x_n(t)]^{\mathrm{T}}$, one can compactly write the overall dynamics of this multiagent system as $\dot{x}(t) = -\mathcal{L}x(t)$, where $\mathcal{L} \triangleq \mathcal{D} - \mathcal{A}$ is the Laplacian matrix with $\mathcal{D} \in \mathbb{R}^{n\times n}$ denoting its degree matrix and $\mathcal{A} \in \mathbb{R}^{n\times n}$ denoting its adjacency matrix (we also refer to the first paragraph of Section 5.2 for details on notation). In particular, the spectrum of the corresponding Laplacian matrix can now be ordered as $0 = \lambda_1(\mathcal{L}) < \lambda_2(\mathcal{L}) \leq \ldots \leq \lambda_n(\mathcal{L})$ ($\lambda_2(\mathcal{L})$ is called as the Fiedler eigenvalue that determines the convergence rate), the null-space of this Laplacian matrix spans $\mathbf{1}_n = [1,\ldots,1]^{\mathrm{T}}$ ($\mathbf{1}_n$ is the eigenvector corresponding the zero eigenvalue $\lambda_1(\mathcal{L})$), and $\lim_{t\to\infty} x(t) = c\mathbf{1}_n$ with c being a scalar (consensus). Note that the above consensus algorithm is the key building block for a wide array of existing distributed control architectures including but not limited to formation architectures, pinning architectures, containment architectures, and dynamic information fusion architectures. Hence, these extensions are also predicated on this Laplacian matrix with a nullspace spanning the vector of ones. The following question is now immediate: *To pave the way for composing complex cooperative behaviors in multiagent systems, can we generalize the Laplacian nullspace such that it can span any vector with positive elements?*

The contribution of this paper is to address the above question, where we introduce a new Laplacian matrix for undirected and connected graphs that generalizes the well-known Laplacian matrix (hereinafter referred to as the standard Laplacian matrix) whose nullspace spans the vector of ones. Specifically, the proposed Laplacian matrix is based on a desired, user-assigned nullspace. We first give the mathematical definition of this Laplacian matrix and show that it inherits some fundamental properties of the standard Laplacian matrix. We then present distributed control architectures for convergence to the desired nullspace and for convergence to a specific vector within that nullspace. Finally, an application of the proposed Laplacian matrix to formation tracking and scaling problem is given. To complement our theoretical results, we also present several illustrative numerical examples.

Note that the authors of [125, 126] also investigate how to drive a given multiagent system to different Laplacian nullspace for undirected and connected graphs. They utilize a similarity transformation onto the standard Laplacian matrix to change its resulting nullspace, where this process leads to the same, standard degree matrix but to a new adjacency matrix. In contrast, our approach is simply predicated on keeping the same, standard adjacency matrix and altering the degree matrix instead. That is, considering a distributed control architecture developed based on the standard Laplacian matrix, one can simply add self-loops to that architecture to achieve convergence to a given user-assigned nullspace based on the results of this paper; however, the results in [125, 126] require the exact knowledge of each neighboring agent states for the same purpose.

5.2 New Laplacian Matrix and the Nullspace Convergence Protocol

We first recall several graph-theoretical notions (e.g., see [5] and [92] for details). Specifically, an undirected graph $\mathcal{G}$ is defined by a set $\mathcal{V}_\mathcal{G} = \{1,\dots,N\}$ of nodes and a set $\mathcal{E}_\mathcal{G} \subset \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ of edges. If $(i,j) \in \mathcal{E}_\mathcal{G}$, then the nodes i and j are neighbors and the neighboring relation is indicated with $i \sim j$. The number of agent i's neighbors is its degree and denoted as d_i. The degree matrix of a graph $\mathcal{G}$, $\mathcal{D}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is then defined by $\mathcal{D}(\mathcal{G}) \triangleq \text{diag}(d)$ with $d = [d_1,\dots,d_N]^{\mathrm{T}}$. In addition, a path $i_0 i_1 \dots i_L$ is a finite sequence of nodes such that $i_{k-1} \sim i_k$, $k = 1,\dots,L$, and a graph $\mathcal{G}$ is called connected when there exists a path between any pair of distinct nodes. The adjacency matrix of a graph $\mathcal{G}$, $\mathcal{A}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is also defined by $[\mathcal{A}(\mathcal{G})]_{ij} = 1$ when $(i,j) \in \mathcal{E}_\mathcal{G}$ and $[\mathcal{A}(\mathcal{G})]_{ij} = 0$ otherwise. Finally, in this paper, the standard Laplacian matrix, $\mathcal{L}(\mathcal{G}) \in \overline{\mathbb{R}}_+^{N\times N}$, is defined by $\mathcal{L}(\mathcal{G}) \triangleq \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$ with $\mathbf{span}\{\mathbf{1}_N\}$ being its nullspace.

5.2.1 The New Laplacian Matrix

Consider a multiagent system with N nodes communicating under a connected and undirected graph $\mathcal{G}$ with the standard adjacency matrix $\mathcal{A}(\mathcal{G})$. Let $w = [w_1,\dots,w_N] \in \mathbb{R}^N$ be a vector with positive elements (i.e., $w_i \in \mathbb{R}_+$ for all $i = 1,\dots,N$), which is the representative vector for the desired nullspace $\mathbf{span}\{w\}$. We define the new, altered degree matrix $\bar{\mathcal{D}}(\mathcal{G},w)$ as a diagonal matrix such that

$$[\bar{\mathcal{D}}(\mathcal{G},w)]_{ii} = \sum_{j=1}^{N} \frac{[\mathcal{A}(\mathcal{G})]_{ij} w_j}{w_i} = \sum_{i\sim j} \frac{w_j}{w_i}, \tag{5.1}$$

or equivalently,

$$\bar{\mathcal{D}}(\mathcal{G},w) \triangleq \mathrm{diag}(\mathcal{A}(\mathcal{G})w)(\mathrm{diag}(w))^{-1} \in \mathbb{R}^{N\times N}. \tag{5.2}$$

For simplicity, we now write $\bar{\mathcal{D}}$ for the new degree matrix defined in (5.2) and $\mathcal{A}$ for the standard adjacency matrix in this section, unless stated otherwise. Next, we define the new Laplacian matrix with the desired, user-assigned nullspace $\mathbf{span}\{w\}$ as

$$\bar{\mathcal{L}}(\mathcal{G},w) \triangleq \bar{\mathcal{D}} - \mathcal{A} = \mathrm{diag}(\mathcal{A}w)(\mathrm{diag}(w))^{-1} - \mathcal{A}. \tag{5.3}$$

Note that when $w = \mathbf{1}_N$, $\bar{\mathcal{L}}(\mathcal{G},w) \equiv \mathcal{L}(\mathcal{G})$, where $\mathcal{L}(\mathcal{G})$ is the standard Laplacian matrix. For the standard Laplacian matrix, $w_i = w_j = \frac{w_j}{w_i} = 1$ for all $i,j = 1,\ldots,N$; thus, the degree of agent i is simply the number of its neighbors. For the case where $w \neq \mathbf{span}\{\mathbf{1}_N\}$, agent i requires w_j value either by default (i.e., preprogrammed) or through information exchange. In what follows, we investigate the properties of the new Laplacian matrix $\bar{\mathcal{L}}(\mathcal{G},w)$.

We first show that $\bar{\mathcal{L}}(\mathcal{G},w)$ is a positive semidefinite matrix. For this purpose, consider the quadratic form of the new Laplacian matrix

$$\begin{aligned}
x^{\mathrm{T}}\bar{\mathcal{L}}(\mathcal{G},w)x &= x^{\mathrm{T}}(\bar{\mathcal{D}} - \mathcal{A})x \\
&= x^{\mathrm{T}}\big(\mathrm{diag}(\mathcal{A}w)(\mathrm{diag}(w))^{-1} - \mathcal{A}\big)x \\
&= x^{\mathrm{T}}\big(\mathrm{diag}(\mathcal{A}w)(\mathrm{diag}(w))^{-1}\big)x - x^{\mathrm{T}}\mathcal{A}x \\
&= \sum_{i=1}^{N} \frac{[\mathcal{A}w]_i}{w_i}x_i^2 - \sum_{(i,j)\in\mathcal{E}_{\mathcal{G}}} 2x_ix_j \\
&= \sum_{i=1}^{N}\left(\sum_{i\sim j}\frac{w_j}{w_i}x_i^2\right) - \sum_{(i,j)\in\mathcal{E}_{\mathcal{G}}} 2x_ix_j \\
&= \sum_{(i,j)\in\mathcal{E}_{\mathcal{G}}}\left(\frac{w_j}{w_i}x_i^2 + \frac{w_i}{w_j}x_j^2\right) - \sum_{(i,j)\in\mathcal{E}_{\mathcal{G}}} 2x_ix_j \\
&= \sum_{(i,j)\in\mathcal{E}_{\mathcal{G}}}\left(\frac{w_j}{w_i}x_i^2 + \frac{w_i}{w_j}x_j^2 - 2x_ix_j\right) \\
&= \sum_{(i,j)\in\mathcal{E}_{\mathcal{G}}}\left(\sqrt{\frac{w_j}{w_i}}x_i - \sqrt{\frac{w_i}{w_j}}x_j\right)^2 \geq 0.
\end{aligned} \tag{5.4}$$

Therefore, $\bar{\mathcal{L}}(\mathcal{G},w)$ is a positive semidefinite matrix. This implies that $0=\lambda_1\leq\lambda_2\leq\ldots\leq\lambda_N$, where λ_i, $i=1,\ldots,N$ are the eigenvalues of the new Laplacian matrix $\bar{\mathcal{L}}(\mathcal{G},w)$. In this paper, the indices of eigenvalues of the new Laplacian matrix follow the above order, unless stated otherwise.

Remark 5.2.1. *We note that the second to last equality in (5.4) requires w_j/w_i and w_i/w_j to be positive. In addition, without loss of generality, the last equality in (5.4) can be rewritten as*

$$x^{\mathrm{T}}\bar{\mathcal{L}}(\mathcal{G},w)x \;=\; \sum_{(i,j)\in\mathcal{E}_{\mathcal{G}}}\left(\sqrt{\frac{|w_j|}{|w_i|}}x_i-\sqrt{\frac{|w_i|}{|w_j|}}x_j\right)^2\geq 0. \tag{5.5}$$

Therefore, (5.4) holds as long as all elements in w are nonzero and shares the same sign. In other words, in general, the new Laplacian (5.3) is defined under any vector w with nonzero elements and share the same sign. The result for w with nonzero elements and arbitrary signs will be presented in a future research.

Remark 5.2.2. *We note that* ***span****$\{w\}$ is in the nullspace of $\bar{\mathcal{L}}(\mathcal{G},w)$ as*

$$\begin{aligned}\bar{\mathcal{L}}(\mathcal{G},w)w &= \mathrm{diag}(\mathcal{A}w)(\mathrm{diag}(w))^{-1}w-\mathcal{A}w\\ &= \mathrm{diag}(\mathcal{A}w)\mathbf{1}_N-\mathcal{A}w=\mathcal{A}w-\mathcal{A}w=0_N.\end{aligned} \tag{5.6}$$

Theorem 1 below further shows that ***span****$\{w\}$ is indeed the nullspace of $\bar{\mathcal{L}}(\mathcal{G},w)$.*

Remark 5.2.3. *The new Laplacian defined in (5.3) can be extended to a weighted graph. Specifically, define a function m: $\mathcal{E}_{\mathcal{G}}\to\mathbb{R}_+$ and let the adjacency matrix of the weighted graph $\mathcal{G}$ denoted by $\mathcal{A}_w$ and is given by*

$$[\mathcal{A}_w]_{ij} \;\triangleq\; \begin{cases} m(i,j), & \text{if } (i,j)\in\mathcal{E}_{\mathcal{G}},\\ 0, & \text{otherwise.}\end{cases}$$

Let $w\in\mathbb{R}^N$ be a vector with positive element. Then, the degree matrix of the weighted graph $\mathcal{G}$ denoted by $\bar{\mathcal{D}}_w$ is a diagonal matrix such that

$$[\bar{\mathcal{D}}_w]_{ii}=\sum_{j=1}^{N}\frac{[A_w]_{ij}w_j}{w_i}=\sum_{i\sim j}\frac{m(i,j)w_j}{w_i}, \tag{5.7}$$

or equivalently,

$$\bar{D}_w \triangleq \text{diag}(A_w w)(\text{diag}(w))^{-1} \in \mathbb{R}^{N\times N}. \tag{5.8}$$

Now, the new weighted Laplacian matrix is defined by

$$\tilde{\mathcal{L}}_w(\mathcal{G}) \triangleq \bar{\mathcal{D}}_w - \mathcal{A}_w. \tag{5.9}$$

Similar to Remark 5.2.2, note that it can be shown that $\boldsymbol{span}\{w\}$ *belongs to the nullspace of* $\tilde{\mathcal{L}}_w(\mathcal{G})$*. In addition, the properties shown in this section can be also generalized to the new weighted Laplacian matrix.*

Next, we assume orientation of each edge is assigned arbitrarily and define the new incidence matrix $\bar{E}(\mathcal{G}) \in \mathbb{R}^{N\times m}$ with m being the number of edges in the graph $\mathcal{G}$ as

$$[\bar{E}(\mathcal{G})]_{ik} = \begin{cases} -\sqrt{\frac{w_j}{w_i}} & \text{if } v_i \text{ is the tail of the edge } e_k = (i,j) \\ \sqrt{\frac{w_j}{w_i}} & \text{if } v_i \text{ is the head of the edge } e_k = (i,j) \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

where $i = 1,\ldots,N$ and $k = 1,\ldots,m$. Therefore, the last equality in (5.4) can be rewritten as

$$\begin{aligned} x^{\mathrm{T}}\bar{\mathcal{L}}(\mathcal{G},w)x &= \sum_{(i,j)\in\mathcal{E}_\mathcal{G}} \left(\sqrt{\frac{w_j}{w_i}}x_i - \sqrt{\frac{w_i}{w_j}}x_j\right)^2 \\ &= \|\bar{E}(\mathcal{G})^{\mathrm{T}}x\|_2^2 \\ &= x^{\mathrm{T}}\bar{E}(\mathcal{G})\bar{E}(\mathcal{G})^{\mathrm{T}}x \geq 0. \end{aligned} \tag{5.11}$$

As a result, the new Laplacian matrix can now be rigorously defined in a similar way as the standard Laplacian matrix

$$\bar{\mathcal{L}}(\mathcal{G},w) = \bar{\mathcal{D}} - \mathcal{A} = \bar{E}(\mathcal{G})\bar{E}(\mathcal{G})^{\mathrm{T}}. \tag{5.12}$$

We are now ready to state the following theorem.

Theorem 5.2.1. *The graph* $\mathcal{G}$ *is connected if and only if* $\lambda_2 > 0$.

Proof. Theorem 2.4.3 of [117] states that let $A \in \mathbb{R}^{n\times m}$, then $A = A^{\mathrm{T}}A$). By letting $\bar{E}(\mathcal{G})^{\mathrm{T}} \equiv A$ in this case, we obtain

$$\begin{aligned} \mathcal{N}(\bar{E}(\mathcal{G})^{\mathrm{T}}) &= \mathcal{N}((\bar{E}(\mathcal{G})^{\mathrm{T}})^{\mathrm{T}}\bar{E}(\mathcal{G})^{\mathrm{T}}) \\ &= \mathcal{N}(\bar{E}(\mathcal{G})\bar{E}(\mathcal{G})^{\mathrm{T}}) \\ &= \mathcal{N}(\bar{\mathcal{L}}(\mathcal{G},w)). \end{aligned} \tag{5.13}$$

That is, the nullspace of $\bar{\mathcal{L}}(\mathcal{G},w)$ and $\bar{E}(\mathcal{G})^{\mathrm{T}}$ are the same. From here, the proof follows in the same spirit of Theorem 2.8 of [5].

Suppose there exists a vector $z \notin \mathbf{span}\{w\}$ such that $z^{\mathrm{T}}\bar{E}(\mathcal{G}) = 0$ (i.e., $z \in \mathcal{N}(\bar{E}(\mathcal{G})^{\mathrm{T}})$), then for each edge $(i,j) \in \mathcal{E}_{\mathcal{G}}$ we have

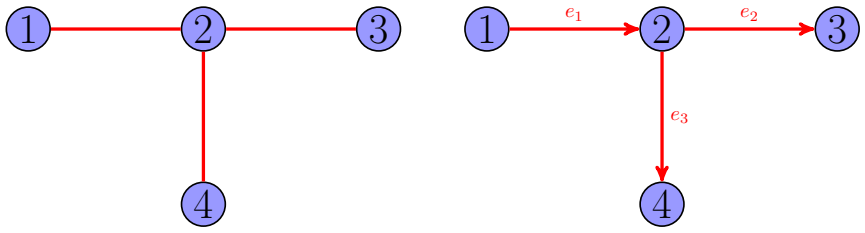
$$-\sqrt{\tfrac{w_j}{w_i}}z_i + \sqrt{\tfrac{w_i}{w_j}}z_j = 0. \tag{5.14}$$

Since w_i and w_j are positive, when $z_i = 0$, (5.14) indicates that $z_j = 0$. Thus, since the graph $\mathcal{G}$ is connected, $z_i = 0$ for all $i = 1,2,\ldots,N$ or $z = 0_N \in \mathbf{span}\{w\}$, which is a contradiction to the initial assumption. In other words, any vector $z \notin \mathbf{span}\{w\}$ and containing zero element does not satisfies (5.14). In addition, when z_i and z_j are nonzero and the graph $\mathcal{G}$ is connected, then from (5.14) the ratio

$$\frac{z_i}{z_j} = \frac{w_i}{w_j}. \tag{5.15}$$

is correct for all $i,j \in \mathcal{V}$. Yet, this means $z \in \mathbf{span}\{w\}$, which is a contradiction to the initial assumption. Consequently, $\mathcal{N}(\bar{E}(\mathcal{G})^{\mathrm{T}})$ has dimension one, or equivalently, the geometric and algebraic multiplicity of the zero eigenvalue of the new Laplacian matrix is one. That is, $\lambda_1 = 0$ and $\lambda_2 > 0$.

Conversely, consider $\lambda_2 > 0$. Then, the nullspace of $\bar{\mathcal{L}}(\mathcal{G},w)$ and $\bar{E}(\mathcal{G})^{\mathrm{T}}$ has dimension one. Thus, if $z \in \mathcal{N}(\bar{\mathcal{L}}(\mathcal{G},w))$, then by (5.6) $z \in \mathbf{span}\{w\}$ and $z^{\mathrm{T}}\bar{E}(\mathcal{G}) = 0$. If the graph $\mathcal{G}$ is not connected, then there exists at least one vector z satisfies (5.14) with some $z_i = z_j = 0$ while other elements is different from 0; that is, $z \notin \mathbf{span}\{w\}$ but $z^{\mathrm{T}}\bar{E}(\mathcal{G}) = 0$. This also indicates that the dimension of $\dim\left(\mathcal{N}(\bar{E}(\mathcal{G})^{\mathrm{T}})\right) \geq 2$, which is a contradiction. Therefore, the graph $\mathcal{G}$ is connected. ■



(a) (b)

Figure 5.1: An undirected and connected graph $\mathcal{G}$ with 4 nodes (a) and its oriented graph (b).

We now illustrate the result above with an example. Consider an undirected and connected graph $\mathcal{G}$ with 4 nodes as shown in Figure 5.1(a). The adjacency matrix $\mathcal{A}$ of the corresponding graph $\mathcal{G}$ is

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5.16}$$

Let $w = [1,2,3,4]^{\mathrm{T}}$. Then, the modified degree matrix is

$$\bar{\mathcal{D}} \triangleq \mathrm{diag}(\mathcal{A}w)(\mathrm{diag}(w))^{-1} = \mathrm{diag}([2,4,\tfrac{2}{3},\tfrac{1}{2}]). \tag{5.17}$$

As a result, we obtain the new Laplacian matrix in the form

$$\bar{\mathcal{L}}(\mathcal{G},w) = \bar{\mathcal{D}} - \mathcal{A} = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 4 & -1 & -1 \\ 0 & -1 & \frac{2}{3} & 0 \\ 0 & -1 & 0 & \frac{1}{2} \end{bmatrix}. \tag{5.18}$$

In addition, we can assign the orientation of the edge as in Figure 5.1(b) and construct the new incidence matrix $\bar{E}(\mathcal{G})$ based on (5.10) as

$$\bar{E}(\mathcal{G}) = \begin{bmatrix} -\sqrt{2} & 0 & 0 \\ \sqrt{\frac{1}{2}} & -\sqrt{\frac{3}{2}} & -\sqrt{2} \\ 0 & \sqrt{\frac{2}{3}} & 0 \\ 0 & 0 & \sqrt{\frac{1}{2}} \end{bmatrix}. \tag{5.19}$$

It can be readily verified that $\bar{\mathcal{L}}(\mathcal{G},w) = \bar{E}(\mathcal{G})\bar{E}(\mathcal{G})^{\mathrm{T}}$.

5.2.2 The Nullspace Convergence Algorithm

In this section, we introduce a new distributed algorithm for multiagent networked system to converge to a desired nullspace and prove the stability of the algorithm. For this purpose, we consider a multiagent system with N agents exchanging information according to a connected and undirected graph $\mathcal{G}$ and operating under the following distributed algorithm

$$\dot{x}_i(t) = -\sum_{i\sim j}\left(\frac{w_j}{w_i}x_i(t) - x_j(t)\right), \quad x_i(0) = x_{i0}, \tag{5.20}$$

or equivalently,

$$\dot{x}_i(t) = -\sum_{i\sim j}(x_i(t) - x_j(t)) + \sum_{i\sim j}\left(1 - \frac{w_j}{w_i}\right)x_i(t), \quad x_i(0) = x_{i0}, \tag{5.21}$$

where $x_i(t) \in \mathbb{R}$ is the state of agent i, $i = 1,\ldots,N$, and $w_i \in \mathbb{R}_+$ denotes the element i-th of the desired nullspace $w \triangleq [w_1,\ldots,w_N] \in \mathbb{R}^N$.

For stability analysis, let $x(t) \triangleq [x_1,\ldots,x_N]^{\mathrm{T}} \in \mathbb{R}^N$ be the aggregated vector, then (5.21) can be written in the compact form given by

$$\dot{x}(t) = -\bar{\mathcal{L}}(\mathcal{G},w)x(t), \quad x(0) = x_0, \tag{5.22}$$

where $\bar{\mathcal{L}}(\mathcal{G},w)$ is the new Laplacian matrix defined in (5.3).

Theorem 5.2.2. *Consider an undirected and connected graph $\mathcal{G}$ with N nodes and the new Laplacian matrix $\bar{\mathcal{L}}(\mathcal{G},w)$ defined by (5.3), where w is a vector with positive entries. Under the distributed protocol given by (5.21) or the compact form given by (5.22), $x(t)$ exponentially converges to $\frac{(w^{\mathrm{T}}x_0)}{\|w\|_2^2}w$.*

Proof. We first note that since $\bar{\mathcal{L}}(\mathcal{G},w)$ is a symmetric matrix, it is diagonalizable by an orthogonal matrix $U \in \mathbb{R}^{N\times N}$ (that is, $UU^{\mathrm{T}} = \mathrm{I}$). In other words, it can be written in the form

$$\bar{\mathcal{L}}(\mathcal{G},w) = U\Lambda U^{\mathrm{T}}, \tag{5.23}$$

where $\Lambda \in \mathbb{R}^{N\times N}$ is a diagonal matrix with eigenvalues of the new Laplacian matrix $\bar{\mathcal{L}}(\mathcal{G},w)$ on the diagonal and U consists of corresponding normalized eigenvectors.

Next, by the variation of constants formula, the explicit solution of (5.22) satisfies

$$\begin{aligned} x(t) &= e^{-\bar{\mathcal{L}}(\mathcal{G},w)t}x_0 \\ &= e^{-U\Lambda U^{\mathrm{T}}t}x_0 \\ &= Ue^{-\Lambda t}U^{\mathrm{T}}x_0 \\ &= u_1e^{-\lambda_1 t}u_1^{\mathrm{T}}x_0+\ldots+u_Ne^{-\lambda_N t}u_N^{\mathrm{T}}x_0. \end{aligned} \tag{5.24}$$

By Theorem 5.2.1, $\lambda_i > 0$ for all $i = 2,\ldots,N$, and $\lambda_1 = 0$ with $u_1 = \frac{w}{\|w\|_2}$. Therefore, $x(t)$ exponentially converges to $(u_1^{\mathrm{T}}x_0)u_1 = \frac{(w^{\mathrm{T}}x_0)}{\|w\|_2^2}w$. ■

Remark 5.2.4. *Under the protocol (5.22), the above theorem shows that $x(t)$ exponentially converges to $\frac{(w^{\mathrm{T}}x_0)}{\|w\|_2^2}w \equiv \frac{w^{\mathrm{T}}x_0}{\|w\|_2}\frac{w}{\|w\|_2}$, which is a vector projection. In other words, the protocol (5.22) projects the initial vector x_0 onto w, or equivalently, the nullspace of $\bar{\mathcal{L}}(\mathcal{G},w)$. Therefore, when $w = \mathbf{1}_N$, or equivalently, $\bar{\mathcal{L}}(\mathcal{G},w) \equiv \mathcal{L}(\mathcal{G})$, the protocol (5.22) shows that the average consensus is actually the projection of the initial vectors onto vector $\mathbf{1}_N$. Figure 5.2 simply illustrates this view point for a network with two agents. Furthermore, the ratio between two agents can be viewed in term of angles. Specifically, let θ_{12} be the angle between x_1-axis and the desired nullspace, then $\tan(\theta_{12}) = \frac{w_2}{w_1}$. Similarly, let θ_{21} be the angle between x_2-axis and the desired nullspace, then $\tan(\theta_{21}) = \frac{w_1}{w_2}$. Therefore, $\theta_{12} = \pi/2 - \theta_{21}$ and $\tan(\theta_{12}) = 1/\tan(\theta_{21})$. This indicates that the original Laplacian matrix is a special case, where $\theta_{ij} = \theta_{ji} = \pi/4$ that leads to the ratio $\frac{w_i}{w_j} = \frac{w_j}{w_i} = 1$. Hence, the desired nullspace for a multiagent system can be obtained through manipulating w_i and w_j (or the ratio $\frac{w_i}{w_j}$). In addition, define the error $e(t) \triangleq x(t) - \frac{(w^{\mathrm{T}}x_0)}{\|w\|_2^2}w$. Then, $e(t)$ is orthogonal to the nullspace represented by w. To elucidate this point, we first obtain*

$$\begin{aligned} w^{\mathrm{T}}x(t) &= w^{\mathrm{T}}e^{-\bar{\mathcal{L}}(\mathcal{G},w)t}x_0 \\ &= w^{\mathrm{T}}\big(\mathrm{I}+\sum_{k=1}^{\infty}\frac{(-\bar{\mathcal{L}}(\mathcal{G},w)t)^k}{k!})\big)x_0 \\ &= w^{\mathrm{T}}x_0, \end{aligned} \tag{5.25}$$

where the last equality come the fact that $w^{\mathrm{T}}\bar{\mathcal{L}}(\mathcal{G},w) = 0_N^{\mathrm{T}}$. We also have

$$w^{\mathrm{T}}\frac{(w^{\mathrm{T}}x_0)}{\|w\|_2^2}w = w^{\mathrm{T}}w\frac{(w^{\mathrm{T}}x_0)}{\|w\|_2^2} = w^{\mathrm{T}}x_0. \tag{5.26}$$

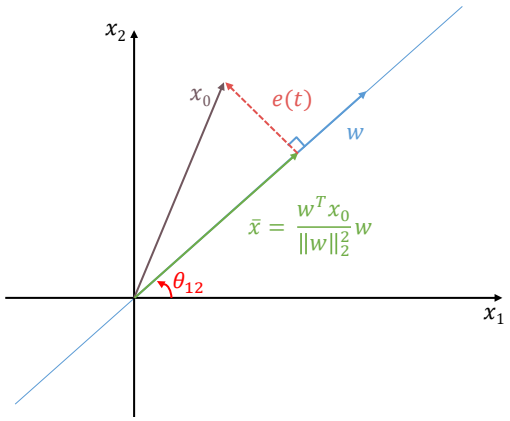


Figure 5.2: The initial vector x_0 is projected onto the nullspace represented by vector w.

.

Combining (5.25) and (5.26) yields

$$w^{\mathrm{T}} e(t) = w^{\mathrm{T}} \left(x(t) - \frac{(w^{\mathrm{T}} x_0)}{\|w\|_2^2} w \right) = 0. \tag{5.27}$$

Hence, the result is immediate. This result is also consistent with other results on the standard Laplacian matrix (see, for example, [19, 95]) that the disagreement vector (i.e, $e(t)$) is orthogonal to $\mathbf{1}_N$.

Remark 5.2.5. *The proposed protocol (5.21) generalizes a wide range of nullspace convergence including the so-called average consensus protocol. Specifically, the nullspace of the average consensus protocol is $w = \boldsymbol{span}\{\mathbf{1}_N\}$. As a result, $w_i = w_j$ for all $i,\ j = 1,\ldots,N$ and the second term of (5.21) is eliminated yielding the structure of the average consensus protocol. In addition, the second term of (5.21) clearly shows that only the degree of agent i (self-loops) is modified compared to the standard average consensus protocol. While the approach in [125] and [126] requires the exact knowledge of each neighboring agent state, the protocol (5.21) only requires each agent to know the "distance" to its neighbor(s) (i.e., $(x_i(t) - x_j(t))$. Therefore, our approach has the potential to require less information exchange compare to the method in [125] and [126].*

5.3 Nullspace Control with the Leader-Follower Algorithm

5.3.1 Convergence to a Specific Vector in the Nullspace

In this section, we extend the result of Section 5.2 to its leader-follower version and show that this new algorithm can be directly applied to drive the multiagent system toward a specific vector in the

nullspace of the new Laplacian matrix. In particular, this new leader-follower algorithm is different from the standard leader-follower algorithm in the sense that while the leader tracks the command, the followers arrange themselves relative to their neighbors such that the ratios $\frac{x_i(t)}{x_j(t)} = \frac{w_i}{w_j}$ for all $(i,j) \in \mathcal{E}_\mathcal{G}$ are satisfied. Mathematically speaking, we consider a multiagent system with N agents exchanging information according to a connected and undirected graph $\mathcal{G}$ and operating under the following distributed algorithm

$$\dot{x}_i(t) = -\sum_{i\sim j}(x_i(t)-x_j(t)) + \sum_{i\sim j}\left(1-\frac{w_j}{w_i}\right)x_i(t) - k_i\left(x_i(t)-c(t)\right), \quad x_i(0)=x_{i0}, \tag{5.28}$$

where $x_i(t) \in \mathbb{R}$ is the state of agent i, $i = 1,\ldots,N$, and $w_i \in \mathbb{R}_+$ denotes the element i-th of the desired nullspace $w \triangleq [w_1,\ldots,w_N] \in \mathbb{R}^N$, $c(t) \in \mathbb{R}$ is the time-varying tracking command with bounded time rate of change (i.e., $|\dot{c}(t)| \leq \bar{c}$ with $\bar{c} \in \mathbb{R}_+$). In addition, we consider the network has at least one leader, where $k_i = 1$ when agent i is a leader and $k_i = 0$ otherwise. Thus, the tracking command $c(t)$ is only available to the leader(s).

Let $x(t) \triangleq [x_1,\ldots,x_N]^{\mathrm{T}} \in \mathbb{R}^N$ be the aggregated vector, then (5.28) can be written in the compact form

$$\dot{x}(t) = -Fx(t) + K\mathbf{1}_N c(t), \quad x(0)=x_0, \tag{5.29}$$

where and $F \triangleq \bar{\mathcal{L}}(\mathcal{G},w) + K \in \mathbb{R}^{N\times N}$ with $K = \mathrm{diag}([k_1,\ldots,k_N]) \in \mathbb{R}^{N\times N}$ and $\bar{\mathcal{L}}(\mathcal{G},w)$ is the new Laplacian matrix defined in (5.3). We are now ready to state the following theorem.

Theorem 5.3.1. *Consider a connected and undirected graph $\mathcal{G}$ with N nodes and the new Laplacian matrix $\bar{\mathcal{L}}(\mathcal{G},w)$ defined by (5.3), where w is a vector with positive elements. Under the distributed protocol given by (5.28) or its compact form given by (5.29), $x(t)$ approaches to the neighborhood of $F^{-1}K\mathbf{1}_N c(t)$ as $t \to \infty$.*

Proof. We first show that F is positive-definite by considering

$$\begin{aligned} x^{\mathrm{T}}(t)Fx(t) &= x^{\mathrm{T}}(t)\big(\bar{\mathcal{L}}(\mathcal{G},w)+K\big)x(t) \\ &= x^{\mathrm{T}}\bar{\mathcal{L}}(\mathcal{G},w)x(t) + x^{\mathrm{T}}(t)Kx(t). \end{aligned} \tag{5.30}$$

Since $x^{\mathrm{T}}(t)\bar{\mathcal{L}}(\mathcal{G},w)x(t) \geq 0$ and $x(t)^{\mathrm{T}}Kx(t) \geq 0$, $x(t)^{\mathrm{T}}Fx(t) \geq 0$. Let $\Omega_1 = \{x|\ x^{\mathrm{T}}\bar{\mathcal{L}}(\mathcal{G},w)x = 0\}$, $\Omega_2 = \{x|\ x^{\mathrm{T}}Kx = 0\}$ and $\Omega = \{x|\ x^{\mathrm{T}}Fx = 0\}$. Note that $x^{\mathrm{T}}(t)Fx(t) = 0$ only when $x^{\mathrm{T}}(t)\bar{\mathcal{L}}(\mathcal{G},w)x(t) = 0$ and

$x^{\mathrm{T}}(t)Kx(t) = 0$ simultaneously. Hence, $\Omega = \Omega_1 \cap \Omega_2$. In addition, $x^{\mathrm{T}}(t)\bar{\mathcal{L}}(\mathcal{G},w)x(t) = x^{\mathrm{T}}(t)\bar{E}(\mathcal{G})\bar{E}(\mathcal{G})^{\mathrm{T}}x(t) = \|\bar{E}(\mathcal{G})^{\mathrm{T}}x(t)\|_2^2 = 0$ only if $x \in \mathcal{N}(\bar{E}(\mathcal{G})^{\mathrm{T}})$. By (5.13), we have $\Omega_1 = \mathcal{N}(\bar{\mathcal{L}}(\mathcal{G},w)) = \mathbf{span}\{w\} = \gamma w$ for any arbitrary $\gamma \in \mathbb{R}$. As a result, $\Omega = \{x|\ x^{\mathrm{T}}Kx = \gamma^2 w^{\mathrm{T}}Kw = \gamma\sum_{i=1}^{N} k_i w_i^2 = 0\}$. Since there is at least one leader or $k_i = 1$ in the network and w_i is nonzero for all $i = 1,\ldots,N$, $\sum_{i=1}^{N} k_i w_i^2 > 0$. Therefore, $x^{\mathrm{T}}Kx = 0$ only when $\gamma = 0$. In other words, $\Omega = \{0_N\}$, or $x^{\mathrm{T}}Fx > 0$ for all $x \neq 0_N$. Thus, F is a positive-definite matrix.

Next, we define the error $e(t) \triangleq x(t) - F^{-1}K\mathbf{1}_N c(t)$, and take its time derivative to obtain

$$\begin{aligned}\dot{e}(t) &= \dot{x}(t) - F^{-1}K\mathbf{1}_N\dot{c}(t)\\ &= -F\big(e(t) + F^{-1}K\mathbf{1}_N c(t)\big) + K\mathbf{1}_N c(t) - F^{-1}K\mathbf{1}_N\dot{c}(t)\\ &= -Fe(t) - F^{-1}K\mathbf{1}_N\dot{c}(t). \end{aligned} \tag{5.31}$$

If $c(t)$ is constant, it is obvious from (5.31) that $x(t) \to F^{-1}K\mathbf{1}_N c(t)$ as $t \to \infty$. We next consider the case when $c(t)$ is not constant. Specifically, consider the Lyapunov function candidate given by

$$V(e) = \frac{1}{2}e^{\mathrm{T}}e. \tag{5.32}$$

Note that $V(0) = 0$ and $V(e) > 0$ for all $e \neq 0$. Taking its time derivative along (5.31) yields

$$\begin{aligned}\dot{V}(\cdot) &= e^{\mathrm{T}}(t)\big(-Fe(t) - F^{-1}K\mathbf{1}_N\dot{c}(t)\big)\\ &= -e^{\mathrm{T}}(t)Fe(t) - e^{\mathrm{T}}(t)F^{-1}K\mathbf{1}_N\dot{c}(t)\\ &= -\lambda_{\min}(F)\|e(t)\|_2^2 + \|e(t)\|_2\|F^{-1}K\|_2\sqrt{N}\bar{c}\\ &= -\lambda_{\min}(F)\|e(t)\|_2\big(\|e(t)\|_2 - \phi\big), \end{aligned} \tag{5.33}$$

where $\phi \triangleq \frac{\|F^{-1}K\|_2\sqrt{N}\bar{c}}{\lambda_{\min}(F)}$. Therefore, $\dot{V}(\cdot) < 0$ outside the compact set $\Psi \triangleq \{e(t)|\ \|e(t)\|_2 \leq \phi\}$, which shows that the error $e(t)$ is uniformly bounded. Hence, the result is now immediate. ■

Remark 5.3.1. *As discussed in the proof of above theorem, when $c(t)$ is constant (that is, $\dot{c}(t) = 0$ for all $t \in [0,\infty)$), then $\phi = 0$. As a result, $\dot{V}(\cdot) < 0$ and the close-loop error $e(t)$ exponentially goes to 0. Furthermore, if $c(t)$ is a time-varying command such that $\dot{c}(t)$ approaches 0 as $t \to \infty$, the close-loop error also converges to 0 as $t \to \infty$.*

Remark 5.3.2. *Observe that* $Fw = \big(\bar{\mathcal{L}}(\mathcal{G},w)+K\big)w = \bar{\mathcal{L}}(\mathcal{G},w)w + Kw = Kw$. *Hence,* $w = F^{-1}Kw$. *In addition, without loss of generality, the command can be written in the form of* $c(t) = \gamma(t)w_i$, *where* w_i *is the leader's corresponding component in the desired nullspace vector* w, *then by definition of* K *matrix, we have* $K\mathbf{1}_N c(t) = \gamma(t)Kw$. *As a result,* $F^{-1}K\mathbf{1}_N c(t) = \gamma(t)F^{-1}Kw = \gamma(t)w$. *This means under the protocol (5.29),* $x(t)$ *converges to the neighborhood of* $\gamma(t)w$. *From Remark 5.3.1, we note that when* $\gamma(t) = 1$, *the system asymptotically converges to* w. *In other words, when the leader converges to* w_i, *the whole network approaches to vector* w. *Furthermore, by adjusting* $\gamma(t)$, *the final convergence value of each agent changes, yet the ratios* $\frac{x_i(t)}{x_j(t)} = \frac{w_i}{w_j}$ *are still maintained for all* $(i,j) \in \mathcal{E}_{\mathcal{G}}$[7].

Remark 5.3.3. *The result shown in the first paragraph of the proof of Theorem 5.3.1 indicates an interesting property of the new Laplacian matrix: Any increment in the degree of any agent in the networked system depicted by a new Laplacian matrix makes the matrix become positive-definite.*

5.3.2 An Application to Formation Control

In this section, we utilize the result of Section 5.3.1 and the multiplex information network architecture proposed in [76] to allow formation scaling and tracking in a distributed manner. First, we note that multiplex information network architecture describes a multiagent system with multiple layers of information exchange including intralayer and interlayer communication links. For the 2D formation tracking problem, we use the standard formation translation algorithm as the main layer and the algorithm (5.28) as the second layer to update the desired relative position of each agent in the formation. Mathematically speaking, consider a group of N vehicles communicating with each other under a connected and undirected graph and operating under the following algorithm

$$\dot{x}_i(t) = -\sum_{i\sim j}\big((x_i(t)-\xi_i(t))-(x_j(t)-\xi_j(t))\big) - k_i(x_i(t)-\xi_i(t)-c_x(t)), \quad x_i(0)=x_{i0}, \tag{5.34}$$

$$\dot{\xi}_i(t) = -\sum_{i\sim j}(\xi_i(t)-\xi_j(t)) + \sum_{i\sim j}\left(1-\frac{w_{xj}}{w_{xi}}\right)x_i(t) - k_i\left(\xi_i(t)-\gamma w_i\right), \quad \xi_i(0)=\xi_{i0}, \tag{5.35}$$

where $x_i(t)$ and $\xi_i(t) \in \mathbb{R}$, $i = 1,\ldots,N$ are the current position and the desired relative position of agent i in x-axis, respectively[8]; $w_x \triangleq [w_{x1},\ldots,w_{xN}] \in \mathbb{R}^N$ represents the desired baseline formation in x-axis of the agent

[7] If there are more than one leader in the network, then the result still holds for $c_i(t) = \gamma(t)w_i$, where i is in the leader set and the term $\gamma(t)$ is known among the leaders.

[8] The same structure is utilized for y-axis, and hence, omitted.

teams with $w_{xi} \in \mathbb{R}_+$ denoting the desired relative position of individual agent i; and $c_x(t) \in \mathbb{R}$ is a time-varying tracking command for the formation in x-axis with bounded time rate of change (i.e., $|\dot{c}_x(t)| \leq \bar{c}_x$ where $\bar{c}_x \in \mathbb{R}_+$). Along the lines of the discussion in Remark 5.3.1, note that the tracking command for the second layer (5.35) is now explicitly in the form of γw_{xi}, where $\gamma \in \mathbb{R}_+$ plays the role as the command scaling factor for the formation. In addition, we consider the network has at least one leader, where $k_i = 1$ if agent i is the leader and $k_i = 0$ otherwise. Thus, the tracking command $c_x(t)$ and the scaling factor γ are only available to the leader(s).

For stability analysis, we define the position error as

$$\tilde{x}_i(t) \triangleq x_i(t) - \xi_i(t) - c_x(t), \tag{5.36}$$

and taking its time derivative to obtain

$$\dot{\tilde{x}}_i(t) \quad = \quad -\sum_{i\sim j}(\tilde{x}_i(t) - \tilde{x}_i(t)) - k_i\tilde{x}_i(t) - \dot{\xi}_i(t) - \dot{c}_x(t), \quad \tilde{x}_i(0) = \tilde{x}_{i0}. \tag{5.37}$$

Let $\tilde{x}(t) \triangleq [\tilde{x}_1, \ldots, \tilde{x}_N]^{\mathrm{T}} \in \mathbb{R}^N$ and $\xi \triangleq [\xi_1, \ldots, \xi_N]^{\mathrm{T}} \in \mathbb{R}^N$ be the aggregated vectors. Then, (5.37) can be written in the compact form given by

$$\dot{\tilde{x}}(t) = -G\tilde{x}(t) - \dot{\xi}(t) - \dot{c}_x(t)\mathbf{1}_N, \quad \tilde{x}(0) = \tilde{x}_0 \tag{5.38}$$

where $G \triangleq \mathcal{L}(\mathcal{G}) + K$ with $K = \mathrm{diag}([k_1, \ldots, k_N]) \in \mathbb{R}^{N\times N}$ and $\mathcal{L}(\mathcal{G})$ is the standard Laplacian matrix. We note that G is a positive definite matrix (e.g., see Lemma 3.3 of [16]).

We are now ready state the following theorem.

Theorem 5.3.2. *Consider the networked multiagent system given by (5.34) and (5.35), where agents exchange their local measurements under a connected and undirected graph $\mathcal{G}$. Then, the closed-loop error dynamics given by (5.38) is input-to-state stable.*

Proof. Utilizing the result of Theorem 5.3.1 and Remark 5.3.1, we can conclude that $\dot{\xi}(t)$ is a bounded signal. In addition, $\dot{c}_x(t)\mathbf{1}_N$ is bounded by assumption. Therefore, we can consider the term $-\dot{\xi}(t) - \dot{c}_x(t)\mathbf{1}_N$ as the bounded input of the closed-loop error dynamics (5.38). Furthermore, since G is

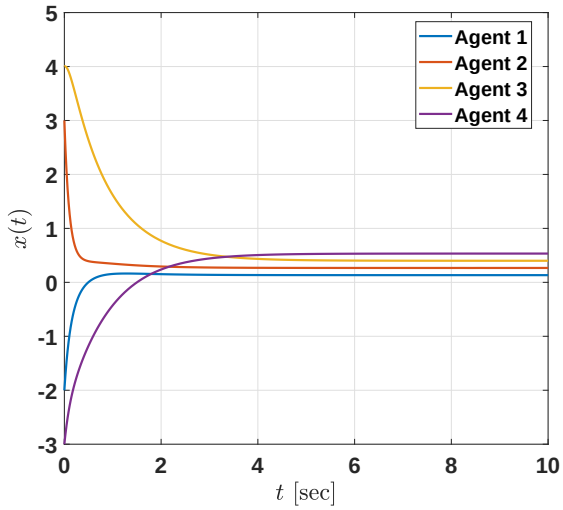


Figure 5.3: The evolution of $x(t)$ under the protocol (5.21) in Example 1.

positive definite, the origin of the unforced system $\dot{\tilde{x}}(t) = -G\tilde{x}(t)$ is exponentially stable. Hence, by Lemma 4.6 of [119], the system (5.38) is input-to-state stable. ■

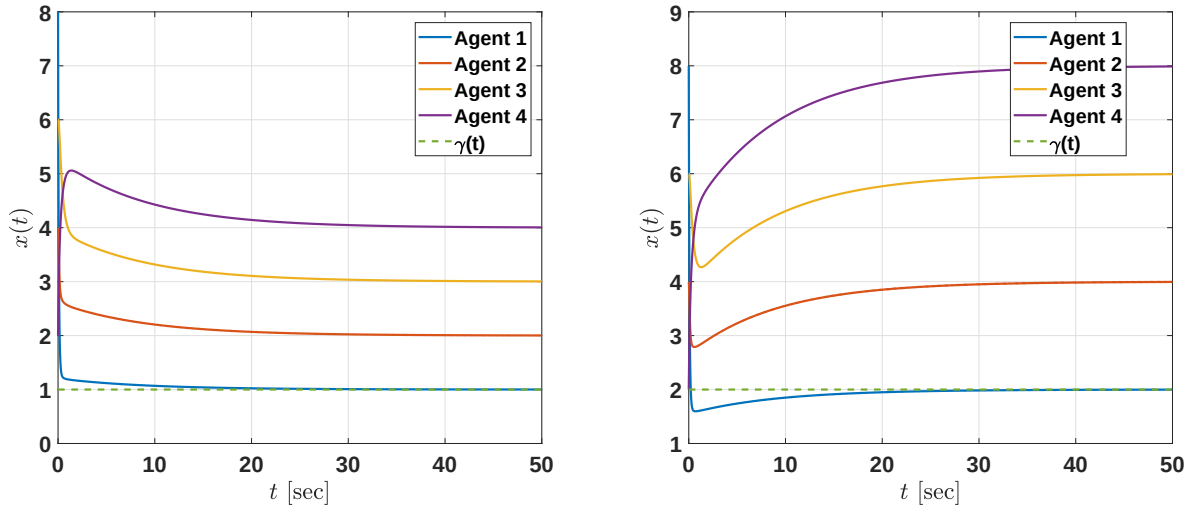
Remark 5.3.4. *By the result of Theorem 5.3.1 and Remark 5.3.1, note that $\xi(t)$ exponentially converges to γw_x; hence, $lim_{t\to\infty}\dot{\xi}(t) = 0_N$. Therefore, Theorem 5.3.2 indicates that when $c_x(t) \equiv c_x$ is a constant command or when $\dot{c}_x(t)$ approaches to 0, then the closed-loop error $\tilde{x}(t)$ approaches to 0. In addition, when $c(t)$ is time-varying, $\tilde{x}(t)$ is ultimately bounded. Furthermore, when $\tilde{x}(t)$ approaches 0, $\tilde{x}_i(t) = \tilde{x}_j(t) = 0$. As a result, $x_i(t) - \xi_i(t) - c_x(t) = x_j(t) - \xi_j(t) - c_x(t)$, or equivalently, $x_i(t) - \gamma w_{xi} - c_x(t) = x_j(t) - \gamma w_{xj} - c_x(t)$ for all $i, j = 1, \ldots, N$, which indicates that agents achieve the formation and are tracking the command.*

5.4 Illustrative Numerical Examples

In this section, we consider several examples to illustrate four contribution. Specifically, we consider a group of 4 agents communicating under a connected and undirected graph $\mathcal{G}$ as depicted in Figure 5.1(a) for all following examples.

5.4.1 Example 1

This example aims to illustrate the protocol given by(5.21). Initially, the state of agents are set to $x(0) = [-2; 3; 4; -3]$. The desired nullspace is chosen to be the span of the representative vector $w = [1; 2; 3; 4]$ and the protocol given by (5.21) is utilized. To accelerate the convergence, a gain $a = 2$ is used to



(a) (b)

Figure 5.4: The evolution of $x(t)$ under the protocol (5.29) in Example 2 with (a) $\gamma(t) = 1$ and (b) $\gamma(t) = 2$.

multiply the protocol. Figure 5.3 shows that agents converge to vector $\frac{(w^{\mathrm{T}}x_0)}{\|w\|_2^2}w = [0.1333; 0.2667; 0.4000; 0.5333]$ as expected from Theorem 5.2.2.

5.4.2 Example 2

This example aims to illustrate the result of Theorem 5.3.1. For this example, agent 1 is chosen as the leader. The initial condition is set to $x(0) = [8,4,6,2]^{\mathrm{T}}$ and the desired nullspace is chosen as $w = [1,2,3,4]^{\mathrm{T}}$. In this case, the protocol (5.28) is implemented and as discussed in Remark 5.3.2, we should choose the command $c(t) = \gamma(t)w_1 \equiv \gamma(t)$. Once again, the protocol is multiplied by a gain $a = 5$ to accelerate the convergence. Figure 5.4(a) shows that when $\gamma(t) = 1$, all agents converge to the desired vector w. In addition, Figure 5.4(b) shows that all agents converges to the $2w$ when $\gamma(t) = 2$.

5.4.3 Example 3

In this example, we illustrate the result of Theorem 5.3.2. Specifically, we choose agent 3 to be the leader in this case. Initially, agents are located at $(x_i, y_i) = (-6,2), (-5,4), (-5,7), (-2,-1)$ for $i = 1,\ldots,4$. The initial desired relative positions of agents in the formation (ξ_{xi}, ξ_{yi}) are set to $(6,6), (2,2), (0,3), (1,1)$ for $i = 1,\ldots,4$, while the actual desired formation (diamond shape) is encoded within $(w_{xi}, w_{yi}) = (2,3), (3,2), (2,1), (1,2)$ for $i = 1,\ldots,4$. The tracking command is set to $(c_x(t), c_y(t)) = (0.1t, 2.5\sin(0.05t))$. As discussed in Section 5.3.2, γ plays the role as the scaling factor. Therefore, to see its effect, for the first

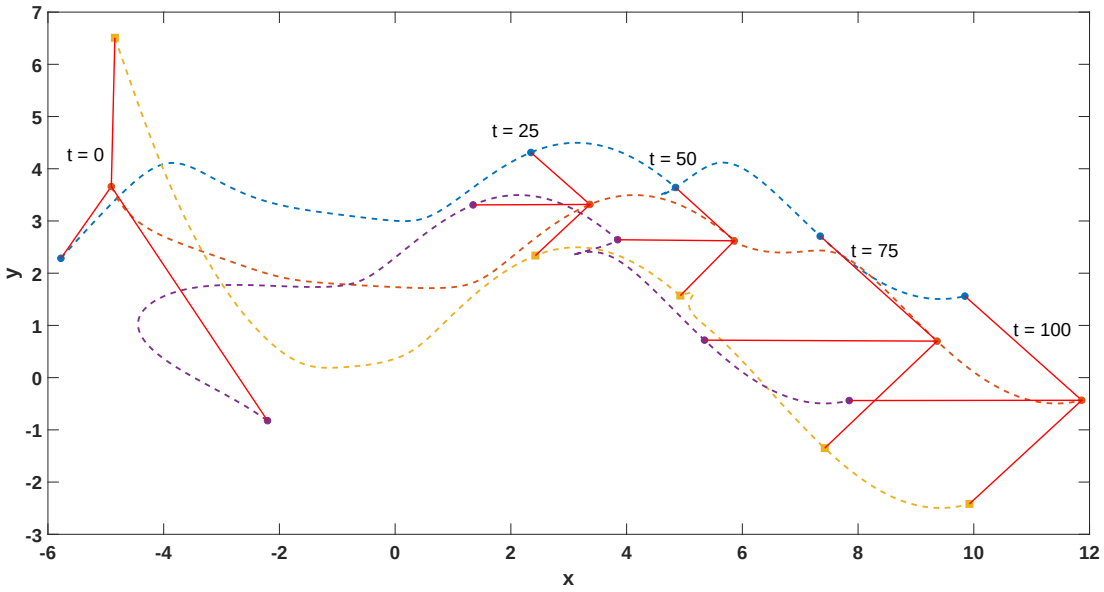


Figure 5.5: Response of the multiagent system under the proposed control architecture given by (5.34) and (5.35) in Example 3. Circles and square denote agents' position at some specific time instants where the square denotes the leader, dashed lines denote agents' trajectories, and solid lines denote the communication links between agents.

50 seconds, we set $\gamma = 1$ and for the last 50 seconds, we set $\gamma = 2$. Under the proposed protocol given by (5.34) and (5.35), agents achieved the desired the formation while tracking the command as illustrated in Figure 5.5. In addition, at t = 50 seconds, the formation size is doubled as expected.

5.5 Conclusion

This paper's contribution was to generalize the standard Laplacian matrix, which has a nullspace spanning the vector of ones, through introducing a new Laplacian matrix, which has a user-assigned nullspace spanning any vector with positive elements. Focusing on undirected and connected graphs, the mathematical definition of this Laplacian matrix was given and its fundamental properties were shown. Distributed control architecture were then presented for convergence to the desired nullspace and for convergence to a specific vector within that nullspace. An application of the proposed Laplacian matrix to formation tracking and scaling problem was also given and several illustrative numerical examples were shown to complement our theoretical results. We believe that the contribution of this paper will open up many research directions to investigate from here toward composing complex cooperative behaviors in multiagent systems through nullspace assignment and control.

Chapter 6: Finite-Time Control of Multiagent Networks as Systems with Time Transformation and Separation Principle*

In this paper, we study finite-time control of *multiagent networks as systems*, where they involve *floating agents* that exchange local information and *driver agents* that not only exchange local information but also take input and output roles. For this class of multiagent networks, control algorithms are applied to the actuators of the driver agents based on the measurements collected from their sensors for the purpose of influencing the overall behavior of the resulting system. Specifically, we consider time-critical applications in the control of multiagent networks as systems and propose a finite-time control approach predicated on a recent *time transformation* method. The proposed method guarantees execution of control algorithms over a prescribed time interval $[0, T)$ with T being a user-defined convergence time based on analysis performed over a stretched, infinite-time interval $[0, \infty)$. We analytically show that the resulting system achieves user-defined finite-time convergence regardless of the initial conditions of agents, where we also discuss the separation principle obtained with the proposed method. The presented theoretical contributions are not only illustrated by numerical examples but also experimentally validated using ground mobile robots.

6.1 Introduction

The last decades have witnessed a considerable attention and growth in theory and application of multiagent networks (see, e.g., [5, 16, 77] and references therein). In the near future, these systems will play a key role for enabling network-centric operations that range from collaborative surveillance and reconnaissance to guidance and control of underwater, ground, aerial, and space vehicle teams. Motivated from this standpoint, this paper contributes to the studies in control of *multiagent networks as systems* (see, e.g., [5, Chapter 10]). This class of multiagent networks consists of *floating agents* and *driver agents*, where the former agents exchange local information through consensus or consensus-like algorithms and the latter

*This chapter will be submitted to the *Control Engineering Practice* journal for possible publication.

agents not only exchange local information but also take input and output roles in the system. Here, control algorithms of interest are applied to the actuators of the driver agents based on the measurements collected from their sensors for the purpose of influencing the overall behavior of the resulting system. An example multiagent network as a system is depicted in Figure 6.1.

6.1.1 Contribution

In this paper, we consider time-critical applications in the control of multiagent networks as systems. In particular, a finite-time control approach is proposed based on a recent *time transformation* method [40, 41]. The key feature of this method is that it guarantees execution of control algorithms over a prescribed time interval $[0, T)$, where T is a user-defined convergence time, based on analysis performed over a stretched, infinite-time interval $[0, \infty)$. Utilizing this method for finite-time control of multiagent networks as systems, we analytically show i) user-defined finite-time convergence of the resulting system regardless of the initial conditions of agents, and ii) separation principle of the proposed time-critical algorithms. Note that the preliminary conference version of this paper has appeared in [127]. This present paper considerably expands on [127] by providing the detail proofs in [127]; additional information discussion, remarks, examples and practical considerations; and experimental results with ground mobile robot platforms.

6.1.2 Related Literature

Finite-time control offers an appealing framework for time-critical applications of dynamical systems. We start with the seminal papers [26, 27], where the authors define finite-time stability for non-

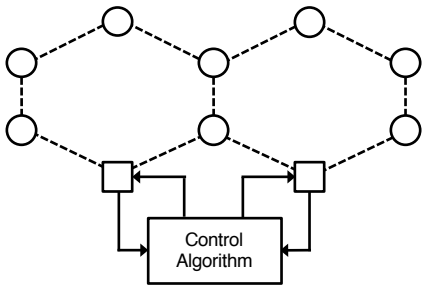


Figure 6.1: An example multiagent network as a system, where *circles* denote the floating agents, *squares* denote the driver agents, *dashed lines* denote the local information exchange between all agents (graph topology), and *solid lines* denote the input-output (feedback) interaction between driver agents and a given control algorithm of interest.

smooth dynamical systems. There exist many studies in the multiagent networks literature that utilize and generalize the results in these two (and similar) papers, where the finite-time convergence depends on the initial conditions of agents. The studies documented in [33–39] address this problem by upper bounding the finite-time convergence time and the studies documented in [40, 41, 43–52] propose system-theoretic tools for guaranteeing user-defined finite-time convergence regardless of the initial conditions of dynamical systems.

As noted in Section 6.1.1, the contribution of this paper builds on the novel time transformation method introduced in [40, 41] that results in smooth control algorithms (the studies in [46–52] are more related than the other aforementioned ones to the contributions documented in these two papers, where we refer to [40, 41] for important differences). Our motivation behind in utilizing and generalizing the results in [40, 41] is primarily owing to the fact that their time transformation method allows one to use well-established system-theoretical tools proposed over infinite-time intervals $[0,\infty)$ for reaching guarantees over the user-defined prescribed time interval $[0,T)$. This key aspect here allows us to analytically show *i)* and *ii)* outlined in Section 6.1.1.

6.1.3 Organization

The content of this paper is as follows. Section 6.2 introduces the necessary mathematical preliminaries for the main results of this paper. The proposed finite-time control approach for multiagent networks as systems is introduced and analyzed in Section 6.3. In addition, a numerical example is presented to demonstrate the proposed system-theoretical results. Section 6.4 discusses some practical considerations and validates the efficacy of the proposed architectures with experimental results. Finally, concluding remarks are summarized in Section 6.5.

6.2 Mathematical Preliminaries

The notation used in this paper is fairly standard. Specifically, $\mathbb{R}$, $\mathbb{R}^n$, and $\mathbb{R}^{n\times m}$ respectively denote the set of real numbers, $n\times 1$ real column vectors, and $n\times m$ real matrices; $\mathbb{R}_+$ and $\mathbb{R}_+^{n\times n}$ (resp., $\overline{\mathbb{R}}_+^{n\times n}$) respectively denote the set of positive real numbers and $n\times n$ positive-definite (resp., positive semi-definite) real matrices; and 0_n, $\mathbf{1}_n$, $0_{n\times n}$, and I_n respectively denote the $n\times 1$ vector of all zeros, the $n\times 1$ vector of all ones, the $n\times n$ zero matrix, and the $n\times n$ identity matrix. In addition, we write $(\cdot)^{\mathrm{T}}$ for transpose, $(\cdot)^{-1}$ for inverse, $\|\cdot\|_2$ for the Euclidian norm, $\lambda_{\min}(A)$ (resp., $\lambda_{\max}(A)$) for the minimum (resp., maximum)

eigenvalue of the symmetric matrix A, $\lambda_i(A)$ for the i-th eigenvalue of A (A is symmetric and the eigenvalues are ordered from least to greatest value), $[A]_{ij}$ for the entry of the of the matrix A on the i-th row and j-th column.

We now concisely overview several notions from graph theory (we refer to, e.g., [5] for details). In particular, graphs are broadly utilized to encode interactions in multiagent networks. To this end, an *undirected* graph $\mathcal{G}$ is defined by a set $\mathcal{V}_\mathcal{G} = \{1,\dots,N\}$ of *nodes* and a set $\mathcal{E}_\mathcal{G} \subset \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ of *edges*. Repeated edges and self-loops are not allowed. If $(i,j) \in \mathcal{E}_\mathcal{G}$, then the nodes i and j are *neighbors* and the neighboring relation is indicated with $i \sim j$. The *degree* of a node is given by the number of its neighbors. Letting d_i be the degree of node i, then the *degree* matrix of a graph $\mathcal{G}$, $\mathcal{D}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is given by $\mathcal{D}(\mathcal{G}) = \mathrm{diag}(d)$, $d = [d_1,\dots,d_N]^{\mathrm{T}}$. A *path* $i_0 i_1 \dots i_L$ is a finite sequence of nodes such that $i_{k-1} \sim i_k$, $k = 1,\dots,L$, and a graph $\mathcal{G}$ is *connected* if there is a path between any pair of distinct nodes. The *adjacency* matrix of a graph $\mathcal{G}$, $\mathcal{A}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is given by $[\mathcal{A}(\mathcal{G})]_{ij} = 1$ if $(i,j) \in \mathcal{E}_\mathcal{G}$ and $[\mathcal{A}(\mathcal{G})]_{ij} = 0$ otherwise. The *Laplacian* matrix of a graph, $\mathcal{L}(\mathcal{G}) \in \overline{\mathbb{R}}_+^{N\times N}$, is given by $\mathcal{L}(\mathcal{G}) = \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$. For the multiagent networks as systems study considered in this paper, we consider that a given multiagent network can be modeled by a connected, undirected graph $\mathcal{G}$ with nodes and edges respectively representing agents and inter-agent communication links.

Finally, the following lemma is necessary for one of the results in this paper.

Lemma 6.2.1 (Fact 2.17.1, [117]). *Let $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$, and $D \in \mathbb{R}^{m\times m}$. Assume that A and D are nonsingular. Then,*

$$\begin{bmatrix} A & B \\ 0_{m\times n} & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & -A^{-1}BD^{-1} \\ 0_{m\times n} & D^{-1} \end{bmatrix}. \tag{6.1}$$

6.3 Finite-Time Control of Multiagent Networks as Systems

6.3.1 Multiagent Networks as Systems Setup

We first introduce the *multiagent networks as systems* setup considered in this paper. As discussed in Section 6.1 (see also Figure 6.1), this class of multiagent networks consists of *floating agents* and *driver agents* (see, e.g., [5, Chapter 10]), where dynamics of each agent satisfies a single integrator form. Specifically, we propose that the floating agents execute the dynamics given by $\dot{x}_i(t) = \alpha\lambda(t)\big[-\sum_{i\sim j}\big(x_i(t) -$

$x_j(t))\big]$ to locally exchange their state information $x_i(t)$. In addition, we propose that the driver agents execute the dynamics given by $\dot{x}_i(t) = \alpha\lambda(t)\big[-\sum_{i\sim j}\big(x_i(t)-x_j(t)\big)+u_i(t)\big]$ and $y_i(t) = x_i(t)$ to not only locally exchange their state information $x_i(t)$ but also take input and output roles in the system, where $u_i(t)$ denotes their control inputs and $y_i(t)$ denotes their output measurements. Building on the finite-time control results documented in [40, 41], we consider that the resulting system evolves over the user-defined prescribed time interval $[0,T)$ with $T \in \mathbb{R}_+$ being a given user-defined convergence time. In addition, $\alpha \in \mathbb{R}_+$ is a gain and $\lambda(t) = 1/(T-t)$ in the above expressions. While we do not constraint the number of floating and driver agents in the system, the number of floating agents is often larger than the number of driver agents in practical applications.

The above multiagent network as a system setup, which consists of a total of N agents that exchange information using their local measurements according to a connected, undirected graph $\mathcal{G}$, can be compactly written as

$$\dot{x}(t) = \alpha\lambda(t)\big(-\mathcal{L}(\mathcal{G})x(t)+Bu(t)\big),\ x(0)=x_0, \tag{6.2}$$

$$y(t) = B^{\mathrm{T}}x(t), \tag{6.3}$$

where $x(t) = \Big[x_1(t),\ x_2(t),\ \ldots,\ x_N(t)\Big]^{\mathrm{T}} \in \mathbb{R}^N$ denotes the aggregated state vector that captures the individual states of floating and driver agents, $u(t) \in \mathbb{R}^p$ denotes the aggregated control vector that captures the inputs applied to the set of driver agents, and $y(t) \in \mathbb{R}^p$ denotes the aggregated output vector that captures the output measurements received from the set of driver agents. Here, $\mathcal{L}(\mathcal{G}) \in \mathbb{R}^{N\times N}$ is the resulting Laplacian matrix (see Section 6.2). In addition, $B \in \mathbb{R}^{N\times p}$ (resp., $B^{\mathrm{T}} \in \mathbb{R}^{p\times N}$) is the input (resp., output) matrix of the form $B = \Big[e_i \quad e_j \quad e_k \quad \ldots\Big]$, where $i,j,k,\ldots$ are the corresponding indices of driver agents in the system and e_i is the column vector with i-th element being equal to one and other elements being equal to zero (we refer to Section 6.3.5 for an example).

It is clear from the above discussion that a driver agent takes both an input and output role in the system. It should be noted that one can also extend the results of this paper to the cases where a subset of driver agents can take input roles only, another subset of driver agents can take output roles only, and the remaining driver agents can take both input and output roles. The reason why this is not considered here is owing to the following practical consideration: When an operator connects to the set of driver agents in a multiagent network from, for example, a ground station, then this operator generally has the ability to receive

output measurement data from this set and inject control inputs to the same set. Finally, for the following results in this paper, it is considered that the pair $(-\mathcal{L}(\mathcal{G}), B)$ is stabilizable. Since $\mathcal{L}(\mathcal{G})$ is symmetric, the pair $(-\mathcal{L}(\mathcal{G}), B^{\mathrm{T}})$ is detectable by duality.

6.3.2 Control Algorithm for Driver Agents

We next introduce the control algorithm, which is applied to the actuators of the driver agents (i.e., "$Bu(t)$" in (6.2)) based on the measurements collected from their sensors (i.e., "$B^{\mathrm{T}}x(t)$" in (6.3)) for the purpose of influencing the overall behavior of the resulting system (see Figure 6.1). Specifically, we propose the finite-time control algorithm over user-defined prescribed time interval $[0, T)$ given by[9]

$$u(t) = K_1\hat{x}(t) + K_2 z(t) + K_3 c(t), \tag{6.4}$$

$$\dot{\hat{x}}(t) = \alpha\lambda(t)\big(-\mathcal{L}(\mathcal{G})\hat{x}(t) + Bu(t) + H(y(t) - B^{\mathrm{T}}\hat{x}(t)\big), \quad \hat{x}(0) = \hat{x}_0, \tag{6.5}$$

$$\dot{z}(t) = \alpha\lambda(t)\big(A_c z(t) + B_{c1}\hat{x}(t) + B_{c2}c(t)\big), \quad z(0) = z_0, \tag{6.6}$$

where $K_1 \in \mathbb{R}^{p\times N}$, $K_2 \in \mathbb{R}^{p\times p}$, $K_3 \in \mathbb{R}^{p\times p}$, $H \in \mathbb{R}^{N\times p}$, $A_c \in \mathbb{R}^{p\times p}$, $B_{c1} \in \mathbb{R}^{p\times N}$, and $B_{c2} \in \mathbb{R}^{p\times p}$. In the execution of the control signal given by (6.4) since the aggregated state vector is not available, one needs to reconstruct the aggregated state vector through the state estimation algorithm given by (6.5) with $\hat{x}(t) \in \mathbb{R}^N$ denoting the estimated state. In addition, to give a designer the flexibility in achieving different sets of control objectives, (6.6) denotes the dynamic compensator with $z(t) \in \mathbb{R}^p$ denoting the dynamic compensator state and $c(t) \in \mathbb{R}^p$ denoting a command of interest. Here, we consider that $c(t)$ and $\dot{c}(t)$ are bounded for $t \geq 0$ and $\dot{c}(t)$ is a piecewise continuous function.

Remark 6.3.1. *The finite-time control algorithm given by (6.4), (6.5), and (6.6) are inside the bottom "box" shown in Figure 6.1. That is, based on the measurements $y(t)$ collected from the sensors of driver agents, an operator executes this control algorithm through injecting $u(t)$ to the actuators of these agents. The aforementioned sets of control objectives that can be achieved with this control algorithm over user-defined prescribed time interval $[0, T)$ is discussed in Section 6.3.4.*

[9] The control architecture given by (6.4), (6.5), and (6.6) without $\lambda(t)$ is the generalized dynamic output feedback (see, e.g., [128, Section 1.3]) .

6.3.3 Stability Analysis Based on Time Transformation and Separation Principle

We now show the stability analysis of the controlled multiagent network as a system based on time transformation and discuss the separation principle. In particular, let the state estimation error be $\tilde{x}(t) \triangleq x(t) - \hat{x}(t)$. This error evolves according to the dynamics given by

$$\begin{aligned}
\dot{\tilde{x}}(t) &= \alpha\lambda(t)\big(-\mathcal{L}(\mathcal{G})x(t) + Bu(t) + \mathcal{L}(\mathcal{G})\hat{x}(t) - Bu(t) - H(y(t) - B^{\mathrm{T}}\hat{x}(t))\big) \\
&= \alpha\lambda(t)\big(-\mathcal{L}(\mathcal{G})\tilde{x}(t) - HB^{\mathrm{T}}\tilde{x}(t)\big) \\
&= -\alpha\lambda(t)\big(\mathcal{L}(\mathcal{G}) + HB^{\mathrm{T}}\big)\tilde{x}(t) \\
&= -\alpha\lambda(t)F\tilde{x}(t),\ \tilde{x}(0) = \tilde{x}_0
\end{aligned} \tag{6.7}$$

where $F \triangleq \big(\mathcal{L}(\mathcal{G}) + HB^{\mathrm{T}}\big) \in \mathbb{R}^{N\times N}$. Here, we assume that $-F$ is Hurwitz by a proper selection of the observer gain matrix H, where the following remark is immediate.

Remark 6.3.2. *Since $(-\mathcal{L}(\mathcal{G}), B^{\mathrm{T}})$ is detectable, one can always find an observer gain matrix H such that $-F$ is Hurwitz. Since $\mathcal{L}(\mathcal{G})$ is also symmetric, a proper selection of H provides that all eigenvalues of F (resp., $-F$) are positive (resp., negative). For example, one trivial selection is $H = B$ that results in $F = \big(\mathcal{L}(\mathcal{G}) + HB^{\mathrm{T}}\big) = \big(\mathcal{L}(\mathcal{G}) + BB^{\mathrm{T}}\big)$ with BB^{T} being a diagonal matrix with ones and zeros on its diagonal. From Lemma 2 in [70] and Lemma 3.3 in [16], it follows that this selection of the observer gain matrix creates a positive-definite F matrix owing to the connected, undirected graph topology.*

Next, using the proposed control algorithm (6.4) in (6.2), one can write

$$\begin{aligned}
\dot{x}(t) &= \alpha\lambda(t)\Big(-\mathcal{L}(\mathcal{G})x(t) + B\big(K_1\hat{x}(t) + K_2 z(t) + K_3 c(t)\big)\Big) \\
&= \alpha\lambda(t)\Big(-\mathcal{L}(\mathcal{G})x(t) + B\big(K_1(x(t) - \tilde{x}(t)) + K_2 z(t) + K_3 c(t)\big)\Big) \\
&= \alpha\lambda(t)\Big(-(\mathcal{L}(\mathcal{G}) - BK_1)x(t) - BK_1\tilde{x}(t) + BK_2 z(t) + BK_3 c(t)\Big), \quad x(0) = x_0.
\end{aligned} \tag{6.8}$$

In addition, the dynamic compensator given by (6.6) can be rewritten as

$$\dot{z}(t) = \alpha\lambda(t)\big(A_c z(t) + B_{c1}x(t) - B_{c1}\tilde{x}(t) + B_{c2}c(t)\big), \quad z(0) = z_0. \tag{6.9}$$

At this point, let $r(t) \triangleq \Big[x^{\mathrm{T}}(t),\ z^{\mathrm{T}}(t),\ \tilde{x}^{\mathrm{T}}(t)\Big]^{\mathrm{T}} \in \mathbb{R}^{2N+p}$. From (6.7), (6.8), and (6.9), one can now write

$$\begin{aligned}\dot{r}(t) &= \alpha\lambda(t)\underbrace{\begin{bmatrix}-(\mathcal{L}(\mathcal{G})-BK_1) & BK_2 & -BK_1\\ B_{c1} & A_c & -B_{c1}\\ 0_{N\times N} & 0_{N\times p} & -F\end{bmatrix}}_{M} r(t)+\alpha\lambda(t)\underbrace{\begin{bmatrix}BK_3\\ B_{c2}\\ 0_{N\times p}\end{bmatrix}}_{N} c(t)\\ &= \alpha\lambda(t)\big(Mr(t)+Nc(t)\big),\quad r(0)=r_0. \end{aligned} \tag{6.10}$$

Here, our finite-time control goal[10] is to show $\lim_{t\to T^-}\big(r(t)+M^{-1}Nc(t)\big)=0$, where "$-M^{-1}Nc(t)$" can capture different sets of control objectives (see Sections 6.3.4 and 6.3.5). In order to achieve this goal, one needs to make the system matrix M in (6.10) Hurwitz (see Theorem 6.3.1), where the following remark is immediate.

Remark 6.3.3. *We start with partitioning the system matrix M in (6.10). For this purpose, let $M \triangleq \begin{bmatrix}M_1 & M_2\\ M_3 & M_4\end{bmatrix}$, where $M_1 \triangleq \begin{bmatrix}-(\mathcal{L}(\mathcal{G})-BK_1) & BK_2\\ B_{c1} & A_c\end{bmatrix}$, $M_2 \triangleq \begin{bmatrix}-BK_1\\ -B_{c1}\end{bmatrix}$, $M_3 \triangleq 0_{N\times(N+p)}$ and $M_4 \triangleq -F$. Obviously, the spectrum of M is equal to the union of the spectrums of M_1 and M_4 owing to the upper block triangular structure of M. This highlights the separation principle. Because, one can judiciously select the controller gain matrices K_1 and K_2 to make M_1 Hurwitz (see below) and select the observer gain matrix H to make M_4 Hurwitz (see Remark 6.3.2). Hence, the design processes for making M_1 and M_4 both Hurwitz are independent.*

We now further elaborate how the controller gain matrices K_1 and K_2 can be selected to render M_1 Hurwitz. To this end, we first present the following lemma.

Lemma 6.3.1. *If $(-\mathcal{L}(\mathcal{G}),B)$ is stabilizable and*

$$rank\begin{bmatrix}-\mathcal{L}(\mathcal{G})-\lambda \mathrm{I}_N & B\\ B_{c1} & 0_{p\times p}\end{bmatrix} = N+p, \tag{6.11}$$

for all $\lambda\in\sigma(A_c)$ with $\sigma(A_c)$ denoting the spectrum of A_c, then the pair

[10] As discussed in, for example, [41] and [40], we are only interested in the response of the multiagent network over the user-defined prescribed time interval $[0,T)$. There are a broad set of applications that support this viewpoint such as time-critical simultaneous strike, multiagent automation, and formation reconfiguration, to name but a few examples.

$$\left(\begin{bmatrix} -\mathcal{L}(\mathcal{G}) & 0_{N\times p} \\ B_{c1} & A_c \end{bmatrix}, \begin{bmatrix} B \\ 0_{p\times p} \end{bmatrix} \right), \tag{6.12}$$

is stabilizable.

Proof. The result is immediate from Lemma 1.26 of [128] with $A \triangleq -\mathcal{L}(\mathcal{G})$, $G_1 \triangleq A_c$, $G_2 \triangleq \mathrm{I}_p$, $D \triangleq 0_{p\times p}$ and $C \triangleq B_{c1}$. ■

The next remark shows that one can always find the controller gain matrices K_1 and K_2 to render M_1 Hurwitz.

Remark 6.3.4. *Under the conditions given in Lemma 6.3.1, there always exist controller gain matrices K_1 and K_2 such that M_1 is Hurwitz. To elucidate this, one can rewrite M_1 as*

$$\begin{aligned} M_1 &= \begin{bmatrix} -(\mathcal{L}(\mathcal{G}) - BK_1) & BK_2 \\ B_{c1} & A_c \end{bmatrix} \\ &= \begin{bmatrix} -\mathcal{L}(\mathcal{G}) & 0_{N\times p} \\ B_{c1} & A_c \end{bmatrix} + \begin{bmatrix} BK_1 & BK_2 \\ 0_{p\times N} & 0_{p\times p} \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} -\mathcal{L}(\mathcal{G}) & 0_{N\times p} \\ B_{c1} & A_c \end{bmatrix}}_{\tilde{A}} + \underbrace{\begin{bmatrix} B \\ 0_{p\times p} \end{bmatrix}}_{\tilde{B}} \underbrace{\begin{bmatrix} K_1 & K_2 \end{bmatrix}}_{\tilde{K}} \\ &= \tilde{A} + \tilde{B}\tilde{K}. \end{aligned} \tag{6.13}$$

The result is now immediate from Lemma 6.3.1.

We are now ready to state the following theorem to achieve our finite-time control goal.

Theorem 6.3.1. *Consider a multiagent network as a system given by (6.2) and (6.3), where agents exchange information using their local measurements according to a connected, undirected graph $\mathcal{G}$. In addition, if M is Hurwitz*[11] *and the driver agents execute the controller architecture given by (6.4), (6.5) and (6.6), the*

[11] See Remarks 6.3.2, 6.3.3, and 6.3.4.

solution to (6.10) is then bounded for $t \in [0,\ T)$ *and*

$$\lim_{t \to T^-} \big(r(t) + M^{-1}Nc(t)\big) = 0. \tag{6.14}$$

Proof. Let $q(t) \triangleq r(t) + M^{-1}Nc(t)$. One can then write

$$\begin{aligned} \dot{q}(t) &= \dot{r}(t) + M^{-1}N\dot{c}(t) \\ &= \alpha\lambda(t)\big(M(q(t) - M^{-1}Nc(t)) + Nc(t)\big) + M^{-1}N\dot{c}(t) \\ &= \alpha\lambda(t)Mq(t) + M^{-1}Nc_d(t), \quad q(0) = q_0, \end{aligned} \tag{6.15}$$

where $c_d(t) \triangleq \dot{c}(t)$. We now use the time transformation method utilized in [41] and [40] (see also [129, Section 1.1.1.4]) to analyze the error dynamics given by (6.15). For this purpose, let $t \triangleq \theta(p) \triangleq T(1 - e^{-p})$ denote the time transformation with $p \in [0, \infty)$ being the argument on stretched, infinite-time interval $[0, \infty)$. Note that

$$\frac{dt}{dp} = \frac{d\theta(p)}{dp} = Te^{-p} \triangleq h(p). \tag{6.16}$$

Next, let $\xi(t)$ denote the solution to the dynamical system given by (6.15), and define $\psi(p) \triangleq \xi(t)$. In addition, let $\psi'(p) \triangleq \frac{d\psi}{dp}$ and use the chain rule to obtain

$$\begin{aligned} \psi'(p) &= \frac{dt}{dp}\frac{d\psi}{dt} \\ &= Te^{-p}\Big(\frac{\alpha}{T - T(1 - e^{-p})}M\psi(p) + M^{-1}Nc_d\big(\theta(p)\big)\Big) \\ &= \alpha M\psi(p) + Te^{-p}M^{-1}Nc_d\big(\theta(p)\big) \\ &= \alpha M\psi(p) + h(p)c_d^*\big(\theta(p)\big), \quad \psi(0) = q_0, \end{aligned} \tag{6.17}$$

where $c_d^*\big(\theta(p)\big) \triangleq M^{-1}Nc_d\big(\theta(p)\big)$. Utilizing (6.16) and (6.17), we have the dynamics of the system in the stretched, infinite-time interval as[12]

$$\psi'(p) = \alpha M\psi(p) + h(p)c_d^*\big(\theta(p)\big), \quad \psi(0) = q_0, \tag{6.18}$$

$$h'(p) = -h(p), \quad h(0) = h_0. \tag{6.19}$$

From this point, one can use standard Lyapunov stability theory over $p \in [0,\infty)$ to show that the origin of the system $\big(\psi,h\big) \equiv (0,0)$ is exponentially stable on this stretched interval. Since M is Hurwitz, there exists a unique positive-definite matrix P such that

$$0 = M^{\mathrm{T}}P + PM + \mathrm{I}. \tag{6.20}$$

We now consider the Lyapunov function candidate given by

$$V(\psi,h) \quad = \quad \psi^{\mathrm{T}}P\psi + \eta h^2, \tag{6.21}$$

where $\eta \in \mathbb{R}_+$ and $P \in \mathbb{R}_+^{(2N+p)\times(2N+p)}$ is the solution of (6.20). Note that $V(0,0) = 0$ and $V(\psi,h) > 0$ for all $(\psi,h) \neq (0,0)$. The derivative of (6.21) with respect to $p \in [0,\infty)$ along the trajectories of (6.18) and (6.19) is given by

$$\begin{aligned} V'(\psi(p),h(p)) &= \psi^{\mathrm{T}}(p)P\big(\alpha M\psi(p) + h(p)c_d^*\big(\theta(p)\big)\big) \\ &\quad + \big(\alpha M\psi(p) + hc_d^*\big(\theta(p)\big)\big)^{\mathrm{T}}P\psi(p) - 2\eta h^{\mathrm{T}}(p)h(p), \\ &= \alpha\psi^{\mathrm{T}}(p)(PM + M^{\mathrm{T}}P)\psi(p) + 2\psi^{\mathrm{T}}(p)Ph(p)c_d^*\big(\theta(p)\big) - 2\eta h^2(p), \\ &= -\alpha\psi^{\mathrm{T}}(p)\psi(p) - 2\eta h^2(p) + 2\psi^{\mathrm{T}}(p)Ph(p)c_d^*\big(\theta(p)\big), \end{aligned} \tag{6.22}$$

[12]Note that the initial condition of (6.19) $h(0) = h_0$ is arbitrary while the solution of (6.17) is a subset of the solutions of the system (6.18) and (6.19) corresponding to the initial condition where $h(0) = T$. Therefore, if the origin of the system (6.18) and (6.19) is exponentially stable, then the solution of (6.17) is bounded and converges to 0.

where $V'(\psi(p),h(p)) \triangleq \big(dV(\psi(p),h(p))\big)/dp$. By applying Young's equality to the term "$2\psi^{\mathrm{T}}(p)Ph(p)c_d^*\big(\theta(p)\big)$", we obtain

$$\begin{aligned} 2\psi^{\mathrm{T}}(p)Ph(p)c_d^*\big(\theta(p)\big) &\leq 2\lambda_{max}(P)|h(p)|\|\psi(p)\|_2\|c_d^*\big(\theta(p)\big)\|_2 \\ &\leq 2\lambda_{max}(P)|h(p)|\|\psi(p)\|_2\bar{c}_d^* \\ &\leq \frac{1}{d}\|\psi(p)\|_2 + d|h(p)|^2\lambda_{max}^2(P)(\bar{c}_d^*)^2 \end{aligned} \tag{6.23}$$

where $\|c_d^*\big(\theta(p)\big)\|_2 \leq \bar{c}_d^*$ follows from the boundedness of $\dot{c}(t)$ and the parameter $d \triangleq \big(\lambda_{max}^2(P)(\bar{c}_d^*)^2\big)^{-1}\eta \in \mathbb{R}_+$.

Next, it follows from (6.22) and (6.23) that

$$\begin{aligned} V'(\psi(p),h(p)) &\leq -\alpha\|\psi(p)\|_2^2 - 2\eta|h(p)|^2 + \frac{1}{d}\|\psi(p)\|_2 + d|h(p)|^2\lambda_{max}^2(P)(\bar{c}_d^*)^2 \\ &= -\|\psi(p)\|_2^2\big(\alpha - \frac{1}{d}\big) - |h(p)|^2\big(2\eta - d\lambda_{max}^2(P)(\bar{c}_d^*)^2\big) \\ &= -\|\psi(p)\|_2^2\big(\alpha - \frac{\big(\lambda_{max}^2(P)(\bar{c}_d^*)^2\big)}{\eta}\big) \\ &\quad -|h(p)|^2\Big(2\eta - \frac{\eta\lambda_{max}^2(P)(\bar{c}_d^*)^2}{\lambda_{max}^2(P)(\bar{c}_d^*)^2}\Big) \\ &= -\|\psi(p)\|_2^2\big(\alpha - \frac{\big(\lambda_{max}^2(P)(\bar{c}_d^*)^2\big)}{\eta}\big) - \eta|h(p)|^2, \end{aligned} \tag{6.24}$$

By choosing $\eta = \big(2\lambda_{max}^2(P)(\bar{c}_d^*)^2\big)/\alpha$, (6.24) becomes

$$V'(\psi(p),h(p)) \quad \leq \quad -\frac{\alpha}{2}\|\psi(p)\|_2^2 - \eta|h(p)|^2. \tag{6.25}$$

Thus, the origin of the dynamics given by (6.18) and (6.19) is globally exponentially stable [119, Theorem 4.10] on the stretched, infinite-time interval $p \in [0,\infty)$ (i.e., $\lim_{p\to\infty}\big(\psi(p),h(p)\big) = (0,0)$ holds). Finally, when we fix the initial condition of (6.19) to $h(0) = T$, the solution of (6.17) $\psi(p) = \xi(t)$ is a subset of the solutions of the system (6.18) and (6.19), where $\xi(t)$ is the solution to (6.15). In addition, t approaches T as p approaches ∞; hence, $\lim_{t\to T^-}\xi(t) = 0$. As a result, (6.14) is now immediate for all initial conditions of $r(t)$. ■

Theorem 6.3.1 shows that $r(t)$ converges to $-M^{-1}Nc(t)$ in T seconds in the sense of (6.14), where T is the user-defined convergence time determining the time-interval of operation $[0,T)$. As discussed earlier, M and N can be chosen to capture different sets of control objectives (see Sections 6.3.4 and 6.3.5).

Remark 6.3.5. *Theorem 6.3.1 also establishes the boundedness of the control signal $u(t)$ given by (6.4) over $[0,T)$. Note that one can also view the term "$\alpha\lambda(t)(-\mathcal{L}(\mathcal{G})x(t)+Bu(t))$" in (6.2) as the total control signal. Motivated from this standpoint, the next theorem establishes the boundedness of this signal through showing that $\dot{r}(t)$ is bounded over $[0,T)$.*

Theorem 6.3.2. *Consider a multiagent network as a system given by (6.2) and (6.3), where agents exchange information using their local measurements according to a connected, undirected graph $\mathcal{G}$. In addition, if M and*

$$\mathcal{M} \triangleq \mathrm{I}+\alpha M \in \mathbb{R}^{2N+p}, \tag{6.26}$$

are both Hurwitz, and the driver agents execute the controller architecture given by (6.4), (6.5) and (6.6), $\dot{r}(t)$ is then bounded for $t \in [0,T)$.

Proof. Taking time derivative of (6.10), one can write

$$\begin{aligned}
\ddot{r}(t) &= \alpha\lambda^2(t)\big(Mr(t)+Nc(t)\big)+\alpha\lambda(t)(M\dot{r}(t)+N\dot{c}(t)) \\
&= \alpha\lambda^2(t)(\alpha\lambda(t))^{-1}\dot{r}(t)+\alpha\lambda(t)(M\dot{r}(t)+N\dot{c}(t)) \\
&= \lambda(t)\dot{r}(t)+\alpha\lambda(t)M\dot{r}(t)+\alpha\lambda(t)N\dot{c}(t) \\
&= \lambda(t)(I+\alpha M)\dot{r}(t)+\alpha\lambda(t)N\dot{c}(t), \;\; r(0)=r_0, \;\; \dot{r}(0)=\dot{r}_0,
\end{aligned} \tag{6.27}$$

where the first equation comes from the fact that $\dot{\lambda}(t)=\lambda^2(t)$. Next, let $s(t) \triangleq \dot{r}(t)$ and rewrite (6.27) as

$$\dot{s}(t)=\lambda(t)(\mathrm{I}+\alpha M)s(t)+\alpha\lambda(t)Nc_d(t), \quad s(0)=\dot{r}_0. \tag{6.28}$$

Let $v(t)$ denote a solution to (6.28). In addition, let $t \triangleq \theta(p) \triangleq T(1-e^{-p})$ denote the time transformation with $p \in [0,\infty)$ being the argument on stretched, infinite-time interval $[0,\infty)$, and define $\zeta(p) \triangleq v(t)$. We also let $\zeta'(p) \triangleq \frac{d\zeta}{dp}$ and use the chain rule to obtain

$$\begin{aligned}
\zeta'(p) &= \frac{dt}{dp}\frac{d\zeta}{dt} \\
&= Te^{-p}\big(\frac{1}{Te^{-p}}(\mathrm{I}+\alpha M)\zeta(p)+\frac{\alpha}{Te^{-p}}Nc_d\big(\theta(p)\big)\big) \\
&= \mathcal{M}\zeta(p)+\alpha Nc_d\big(\theta(p)\big), \quad \zeta(\theta^{-1}(0))=\dot{r}_0,
\end{aligned} \tag{6.29}$$

Since $c_d\big(\theta(p)\big)$ is bounded, and $\mathcal{M}$ is Hurwitz by assumption (6.26), $\zeta(p)$ is a bounded solution to the dynamical system given by (6.29) on the stretched, infinite-time interval $p \in [0,\infty)$. Thus, $v(t)$ is bounded on $[0,T)$ and so is $\dot{r}(t)$. The result is now complete. ■

Remark 6.3.6. *In Theorem 6.3.2, it is assumed that both M and $\mathcal{M}$ are Hurwitz. It should be noted that if $\mathcal{M}$ is Hurwitz and since α is a positive design parameter, then it can be readily shown through Lyapunov equation arguments that M is Hurwitz. On the other hand, if M is Hurwitz, then it can be shown by directly checking the eigenvalue of $\mathcal{M}$ that $\alpha > -1/\mathrm{Re}(\lambda_{max}(M))$ makes $\mathcal{M}$ Hurwitz. To summarize, since M can always be made Hurwitz (see Remark 6.3.2 and Remark 6.3.4), α can be judiciously selected according to $\alpha > -1/\mathrm{Re}(\lambda_{max}(M))$ for $\mathcal{M}$ to be Hurwitz.*

Remark 6.3.7. *Theorem 6.3.2 establishes the boundedness of $\dot{r}(t)$, $t \in [0,T)$. Since $\dot{r}(t) \triangleq \left[\dot{x}^{\mathrm{T}}(t),\, \dot{z}^{\mathrm{T}}(t),\, \dot{\hat{x}}^{\mathrm{T}}(t)\right]^{\mathrm{T}}$ by definition and the total control signal (i.e., "$\alpha\lambda(t)(-\mathcal{L}(\mathcal{G})x(t)+Bu(t))$" in (6.2)) equals to the first entry in $\dot{r}(t)$, then this theorem directly gives the boundedness of this total control signal. Under the condition given in Theorem 6.3.2, one can also show that $\dot{u}(t)$ is bounded. To see this, consider the time derivative of (6.4) given by $\dot{u}(t) = K_1\dot{\hat{x}}(t) + K_2\dot{z}(t) + K_3\dot{c}(t)$. Now, from the boundedness of $\dot{c}(t)$ by definition and Theorem 6.3.2, the result is immediate. If, in addition, $c(t)$ is a constant command, it then follows that $\lim_{t\to T^-}\dot{u}(t) = 0$. In this constant command case and using the steps highlighted in the proofs of Theorem 6.3.2, one can also show that $u^{(\xi)}(t) \triangleq d^{\xi}u(t)/dt^{\xi}$ is bounded and $\lim_{t\to T^-}u^{(\xi)}(t) = 0$ when $\mathcal{M}_{\xi} \triangleq \xi \mathrm{I} + \alpha M$ is Hurwitz with ξ being a positive integer. Similar to the discussion in Remark 6.3.6, note that $\alpha > -\xi/\mathrm{Re}(\lambda_{max}(M))$ makes $\mathcal{M}_{\xi}$ Hurwitz when M is Hurwitz.*

6.3.4 Discussion on the Structure of "$-M^{-1}N$"

Theorem 6.3.1 shows that $r(t)$ converges to $-M^{-1}Nc(t)$ in the sense of (6.14) at the user-defined convergence time T. Here, we elaborate the structure of "$-M^{-1}N$" since it can be designed to capture different sets of control objectives. Recall that we consider above the partitioned matrix given by $M = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}$. One can also partition the matrix N as $N = \begin{bmatrix} N_1 \\ N_2 \end{bmatrix}$ with $N_1 = \begin{bmatrix} BK_3 \\ B_{c2} \end{bmatrix}$ and $N_2 = 0_{N\times p}$. Under the conditions of

Lemma 6.2.1, note that one can write

$$M^{-1} = \begin{bmatrix} M_1^{-1} & -M_1^{-1}M_2M_4^{-1} \\ 0_{N\times(N+p)} & M_4^{-1} \end{bmatrix}. \tag{6.30}$$

As a result, we obtain

$$\begin{aligned} -M^{-1}N &= \begin{bmatrix} -M_1^{-1} & M_1^{-1}M_2M_4^{-1} \\ 0_{N\times(N+p)} & -M_4^{-1} \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} \\ &= \begin{bmatrix} -M_1^{-1}N_1 + M_1^{-1}M_2M_4^{-1}N_2 \\ 0_{N\times(N+p)}N_1 - M_4^{-1}N_2 \end{bmatrix} \\ &= \begin{bmatrix} -M_1^{-1}N_1 \\ 0_{N\times p} \end{bmatrix}, \end{aligned} \tag{6.31}$$

where the last equality results from the fact that $N_2 = 0_{N\times p}$. This implies that $\lim_{t\to T^-}\tilde{x}(t) = 0$.

Next, let $\bar{r}(t) \triangleq \begin{bmatrix} x(t),\ z(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{N+p}$. From (6.31), it is clearly that $\lim_{t\to T^-}\left(\bar{r}(t) + M_1^{-1}N_1c(t)\right) = 0$ holds. To further elaborate the structure of the term "$-M_1^{-1}N_1$", let

$$M_1 \triangleq \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = \begin{bmatrix} -(\mathcal{L}(\mathcal{G}) - BK_1) & BK_2 \\ B_{c1} & A_c \end{bmatrix}, \tag{6.32}$$

$$N_1 \triangleq \begin{bmatrix} N_{11} \\ N_{21} \end{bmatrix} = \begin{bmatrix} BK_3 \\ B_{c2} \end{bmatrix}, \tag{6.33}$$

with $M_{11}, M_{12}, M_{21}, M_{22}, N_{11}, N_{21}$ being the corresponding matrices. Since M_1 can always be made Hurwitz (see Remark 6.3.4), we define $M_1^{-1} \triangleq \begin{bmatrix} \bar{M}_{11} & \bar{M}_{12} \\ \bar{M}_{21} & \bar{M}_{22} \end{bmatrix}$. As a result, "$-M_1^{-1}N_1$" can be equivalently written as

$$-M_1^{-1}N_1 = -\begin{bmatrix} \bar{M}_{11}N_{11} + \bar{M}_{12}N_{21} \\ \bar{M}_{21}N_{11} + \bar{M}_{22}N_{21} \end{bmatrix}. \tag{6.34}$$

Depending on the considered control objective, one can design the structure of (6.34) accordingly (see the following remark and Section 6.3.5 for a specific example).

Remark 6.3.8. *Consider, for example, N agents on a connected, undirected path graph with the first agent being the driver agent. Assume that one would like all agents in the multiagent network to converge to a spatial location, $c(t)$, at the user-defined convergence time T (i.e., rendezvous). In this case, we can set $K_1 = -\frac{1}{N}\mathbf{1}_N^{\mathrm{T}}$, $K_2 = 0$, $K_3 = 1$, $A_c = -1$, $B_{c1} = 0$, and $B_{c2} = 0$, where we assume $-(\mathcal{L}(\mathcal{G}) + \frac{1}{N}B\mathbf{1}_N^{\mathrm{T}})$ is Hurwitz, then these selections result in $-M_1^{-1}N_1c(t) = [\mathbf{1}_N, 0]^{\mathrm{T}}c(t)$. As a consequence, $\lim_{t\to T^-}\big(x(t) - \mathbf{1}_N c(t)\big) = 0$ and $\lim_{t\to T} z(t) = 0$, where the former convergence yields $\lim_{t\to T^-}\big(x_i(t) - c(t)\big) = 0$.*

6.3.5 Illustrative Numerical Example

In this section, we show a numerical example to illustrate the efficacy of the proposed control architecture. Specifically, consider a multiagent network with 4 agents (i.e., $N = 4$) subject to a connected, undirected path graph, where agents 2 and 3 are the driver agents ($p = 2$) while agents 1 and 4 are the floating agents. For (6.2) and (6.3), this selection of driver and floating agents yields

$$B^{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{6.35}$$

Note that the pair $(-\mathcal{L}(\mathcal{G}), B)$ is controllable in this case. In what follows, random initial conditions are assigned to all agents.

Here, our objective is to design a control algorithm to split the network such that agents 1 and 2 reach to a desired command ($c_1(t) = 3 + 3\sin(0.5t) + \sin(4t)$ is used) and agent 3 and 4 reach to another desired command ($c_2(t) = -2 + 1.5\cos(0.8t)$ is used) at $T = 5$ seconds. It should be noted that based on Theorem 6.3.1 and (6.34) that $\lim_{t\to T^-}\big(x(t) + (\bar{M}_{11}N_{11} + \bar{M}_{12}N_{21})c(t)\big) = 0$; hence, with this given objective, we need to select gain matrices according to $S = -(\bar{M}_{11}N_{11} + \bar{M}_{12}N_{21})$ with

$$S^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{6.36}$$

We now explain the process of designing the gain matrices $K_1, K_2, K_3, H, A_c, B_{c1}$, and B_{c2} for the proposed control architecture. Specifically, recall that $F = \big(\mathcal{L}(\mathcal{G}) + HB^{\mathrm{T}}\big)$, where H needs to be designed such that F is a matrix with positive real part eigenvalues. As discussed in Remark 6.3.2, a choice of $H = B$ makes F a positive definite matrix. In this example, we do not adopt a dynamic compensator (6.6); hence,

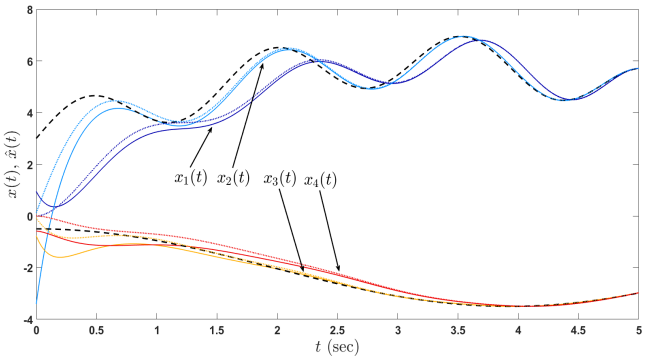


Figure 6.2: Response of the multiagent network as a system in Example 2 (the dashed lines denote the trajectories of the commands $c_1(t)$ and $c_2(t)$, the solid lines denote actual states $x(t)$ of all agents, and the dotted lines denotes the state estimation $\hat{x}(t)$).

$K_2 = 0_{p\times p}$ for the control input $u(t)$ given by (6.4)[13]. In addition, without loss of generality we can choose $A_c = -\mathrm{I}_p$, $B_{c1} = 0_{p\times N}$ and $B_{c2} = 0_{p\times p}$. Since the pair $(-\mathcal{L}(\mathcal{G}), B)$ is controllable, we can always find K_1 such that $-\mathcal{L}(\mathcal{G}) + BK_1$ is Hurwitz with

$$K_1 = \begin{bmatrix} -1.1956 & -1.0722 & -0.3942 & -0.1387 \\ -0.1387 & -0.3942 & -1.0722 & -1.1956 \end{bmatrix}. \tag{6.37}$$

With these choices, M_1 in (6.13) is a block diagonal matrix and the matrices M_1 and M are Hurwitz. Once again, in order to meet the objective, we need to design the gain matrices such that $\lim_{t\to T}\big(x(t) + (\bar{M}_{11}N_{11} + \bar{M}_{12}N_{21})c(t)\big) = \lim_{t\to T^-}\big(x(t) - (\mathcal{L}(\mathcal{G}) - BK_1)^{-1}BK_3c(t)\big) = \lim_{t\to T^-}\big(x(t) - Sc(t)\big) = 0$, where S is given by (6.36). By choosing

$$K_3 = \begin{bmatrix} 3.2678 & -0.4671 \\ -0.4671 & 3.2678 \end{bmatrix}, \tag{6.38}$$

one can satisfy (6.36). With the choice of $\alpha = 10$ and the above gain matrices, the assumption in Theorem 6.3.2 given by (6.26) is also satisfied.

The response of the multiagent network as a system in Example 2 given by (6.2) and (6.3) under the proposed control architecture (6.4) and (6.5) is shown in Figure 6.2, where the dashed lines denote the trajectories of the commands $c_1(t)$ and $c_2(t)$, the solid lines denote the actual states $x(t)$ of all agents and the dotted lines denote the state estimation $\hat{x}(t)$. As expected from Theorem 6.3.1 along with the above selection

[13] An example that considers a dynamic compensator can be found in [127].

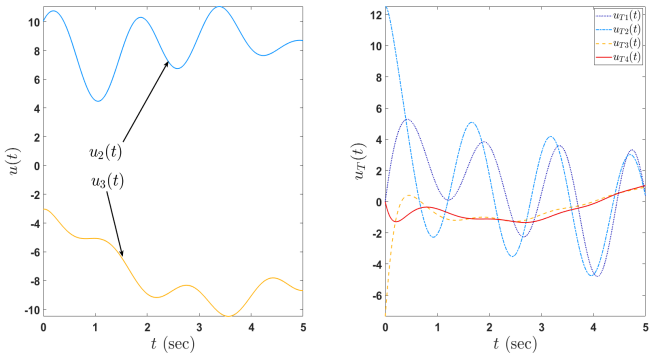


Figure 6.3: The control signals of driver agents (i.e., agents 2 and 3 in the considered multiagent network as a system) and the total control signals of all agents in Example 2, where $u_T(t)$ is the total control signal depicted by the right hand side of (6.2).

of the gain matrices, the states of agents 1 and 2 converge to $c_1(t)$ and the states of agents 3 and 4 converge to $c_2(t)$ at $T = 5$ seconds. The resulting control signals are also shown in Figure 6.3.

6.4 Practical Considerations and Experiments

6.4.1 Practical Considerations

In Section 6.3, we show that the proposed algorithm can meet the objective in a user-defined finite time T. However, when t gets sufficiently close to T or when T is chosen as a small value, the controller can have a high-gain effect that is not desired in practice. To address this issue, the finite time gain $\lambda(t) = 1/(T-t)$ needs to be bounded at some point, but the question is when? In this section, we investigate the following two cases: i) Find the time T_s such that agents are sufficiently close to meet the objective. ii) If the finite time gain is assigned an upper bound ahead of time, find the time T_s such that agents are sufficiently close to meet the objective and the finite time gain does not exceed the given bound before that time.

We now start with case i) by reconsidering the system error dynamics given by (6.18) and (6.19) on the stretched, infinite-time interval

$$\psi'(p) = \alpha M\psi(p) + h(p)c_d^*\big(\theta(p)\big), \quad \psi(0) = q_0, \tag{6.39}$$

$$h'(p) = -h(p), \quad h(0) = h_0, \tag{6.40}$$

where $h(p) = Te^{-p}$. Here, the explicit solution of (6.39) can be written as

$$\begin{aligned}\psi(p) &= e^{p\alpha M}\psi(0) + \int_0^p e^{(p-s)\alpha M} h(s) c_d^*\big(\theta(s)\big) ds \\ &= e^{p\alpha M}\psi(0) + \int_0^p e^{((p-s)\alpha M - s)} T c_d^*\big(\theta(s)\big) ds. \end{aligned} \tag{6.41}$$

Note that $\|e^{p\alpha M}\|_2 \leq k e^{-\alpha\bar{\lambda}p}$ for some $k, \bar{\lambda} \in \mathbb{R}_+$ such that $\bar{\lambda} \leq -\mathrm{Re}(\lambda_{\max}(M))$ [130, Chapter 8][14]. As discussed in Theorem 6.3.2 and Remarks 6.3.6 and 6.3.7, to guarantee the boundedness of $\dot{r}(t)$ over $[0, T)$, one should choose $\alpha > -1/\mathrm{Re}(\lambda_{\max}(M))$, which can be readily satisfied by choosing $\alpha > 1/\bar{\lambda}$ (i.e., $\alpha\bar{\lambda} - 1 > 0$). In addition, we have $\|c_d^*\big(\theta(p)\big)\|_2 \leq \bar{c}_d^*$, so the solution (6.41) can be upper bounded by

$$\begin{aligned}\|\psi(p)\|_2 &\leq k e^{-\alpha\bar{\lambda}p}\|\psi(0)\|_2 + T\bar{c}_d^* \int_0^p e^{\left(-\alpha\bar{\lambda}(p-s)-s\right)} ds \\ &= k e^{-\alpha\bar{\lambda}p}\|\psi(0)\|_2 + T\bar{c}_d^* e^{-\alpha\bar{\lambda}p} \int_0^p e^{(\alpha\bar{\lambda}-1)s} ds \\ &= k e^{-\alpha\bar{\lambda}p}\|\psi(0)\|_2 + \frac{T\bar{c}_d^* e^{-\alpha\bar{\lambda}p}}{\alpha\bar{\lambda}-1}\left(e^{(\alpha\bar{\lambda}-1)p} - 1\right) \\ &\leq k e^{-\alpha\bar{\lambda}p}\|\psi(0)\|_2 + \frac{T\bar{c}_d^* e^{-\alpha\bar{\lambda}p}}{\alpha\bar{\lambda}-1} e^{(\alpha\bar{\lambda}-1)p} \\ &= k e^{-\alpha\bar{\lambda}p}\|\psi(0)\|_2 + \frac{T\bar{c}_d^* e^{-p}}{\alpha\bar{\lambda}-1} \\ &= e^{-p} \underbrace{\left(k e^{-(\alpha\bar{\lambda}-1)p}\|\psi(0)\|_2 + \frac{T\bar{c}_d^*}{\alpha\bar{\lambda}-1}\right)}_{H(p)} \\ &= e^{-p} H(p). \end{aligned} \tag{6.42}$$

This explicit solution not only justifies the result of Theorem 6.3.1 but also can be used to estimate the sufficient time $T_s < T$ for the error approaching the zero neighborhood. It can be seen that $\lim_{p\to\infty} H(p) = \frac{T\bar{c}_d^*}{\alpha\bar{\lambda}-1}$. Let $\varepsilon \in \mathbb{R}_+$ such that $\varepsilon > \frac{T\bar{c}_d^*}{\alpha\bar{\lambda}-1}$ and at $p = p_s$ we have

$$H(p_s) = k e^{-(\alpha\bar{\lambda}-1)p_s}\|\psi(0)\|_2 + \frac{T\bar{c}_d^*}{\alpha\bar{\lambda}-1} = \varepsilon. \tag{6.43}$$

One can directly solve (6.43) for p_s and obtain

[14] When all Jordan blocks of M have multiplicity equal to 1, we can choose $\bar{\lambda} = -\lambda_{\max}(M)$. Otherwise, $\bar{\lambda}$ has to strictly satisfy $\bar{\lambda} < -\lambda_{\max}(M)$.

$$p_s = \frac{1}{-(\alpha\bar{\lambda}-1)} \ln \underbrace{\left(\frac{\varepsilon - \frac{T\bar{c}_d^*}{\alpha\bar{\lambda}-1}}{k\|\psi(0)\|_2} \right)}_{\delta} = \frac{-\ln(\delta)}{\alpha\bar{\lambda}-1}. \tag{6.44}$$

When the error of the system gets sufficiently small, the term $\varepsilon - \frac{T\bar{c}_d^*}{\alpha\bar{\lambda}-1}$ approaches to 0; that is, δ also approaches to 0. Therefore, δ can be used as an alternative quantity for the error to measure how close agents are to meet the objective. Mathematically, by choosing $\delta \in \mathbb{R}_+$ and $\delta \ll 1$, one can estimate the equivalent value of T_s in the stretch time domain, which is $p_s = \frac{-\ln(\delta)}{\alpha\bar{\lambda}-1}$, without the need to explicitly know the parameters $\varepsilon, k, \psi(0)$, and $\bar{c}_d^*$. Subsequently, the time transformation $t = T(1-e^{-p})$ can be utilized to calculate

$$T_s = T(1-e^{-p_s}) = T(1-\delta^{\frac{1}{\alpha\bar{\lambda}-1}}). \tag{6.45}$$

As discussed earlier, at time $t = T_s$, agents are sufficiently close to meet the objective; and hence, we can bound the finite time gain from this time onward to prevent the possible high-gain affect; that is, $\lambda(t) = 1/(T-T_s)$ for $t \geq T_s$.

For case ii), we define an upper bound $\beta \in \mathbb{R}_+$ ahead of time for the finite time gain such that $\lambda(t) = 1/(T-t) \leq \beta$ (for the purpose of bounding it at $t = T_s$). However, if T is chosen as a small value, the finite time gain could have been bounded when agents were still too far away from the objective. Therefore, we first need to calculate the required time T_s for the system to sufficiently close to meet the objective under this extra condition and then choose T accordingly. In particular, T_s has to satisfy $T_s \leq T - \frac{1}{\beta} \in \mathbb{R}_+$. Utilizing the result of (6.45) to get T in term of T_s and substituting to the above inequality yield

$$T_s \leq \frac{T_s}{1-\delta^{\frac{1}{\alpha\bar{\lambda}-1}}} - \frac{1}{\beta}. \tag{6.46}$$

Once again, by choosing $\delta \in \mathbb{R}_+$ and $\delta \ll 1$, one can guarantee that $\left(1-\delta^{\frac{1}{\alpha\bar{\lambda}-1}}\right) > 0$ and can acquire T_s from (6.46) as

$$\begin{aligned} T_s &\geq \frac{1}{\beta}\left(1-\delta^{\frac{1}{\alpha\bar{\lambda}-1}}\right)\delta^{\frac{-1}{\alpha\bar{\lambda}-1}} \\ &= \frac{1}{\beta}\left(\delta^{\frac{-1}{\alpha\bar{\lambda}-1}} - 1\right). \end{aligned} \tag{6.47}$$

At this point, one can obtain

$$T = \frac{T_s}{1 - \delta^{\frac{1}{\alpha\bar{\lambda}-1}}}. \tag{6.48}$$

Remark 6.4.1. *Note that in case i) the finite time gain $\lambda(t)$ has no predefined upper bound and is bounded at time $t = T_s$, when the error gets sufficiently small (by selecting a small value for δ); hence, we still have the freedom to choose T arbitrarily. In case ii), the finite time gain $\lambda(t)$ is bounded under two conditions: i) it reaches the upper bound and ii) the error is sufficiently small. To satisfy both conditions simultaneously, T_s is required to satisfy the inequality (6.47) and therefore the freedom to choose T is reduced, as the choice of T now depends on the choice of T_s.*

Remark 6.4.2. *It should be noted that the smaller the choice of of δ is, the larger the right hand sides of (6.45) and (6.47) are; hence, the larger the value of T_s.*

6.4.2 Experiments

In this section, we present two experimental results to justify the efficacy of the control architecture proposed in Section 6.3 together with the practical considerations discussed in Section 6.4.1. Specifically, the control architecture is implemented on a group of four ground mobile robots communicating in a path graph with robots 2 and 3 are driver agents and robots 1 and 4 are floating agents. The objective of the robots is the same as the illustrative numerical example in Section 6.3.5, where the group is split into two so that the robots 1 and 2 track a target $c_1(t)$ while the robots 3 and 4 track a target $c_2(t)$. Note that for the experiments, a two-level control hierarchy is considered in such a way that the proposed control architecture plays the role as a high-level guidance law for the robots and each robot also has a low-level control law for tracking commands generated from the high-level one.

The differential drive robot platform used in the experiments is Khepera IV (Figure 6.4). In addition, a motion capture system with Vicon Vero 2.2 cameras is used to obtain the position and orientation of each robot. Furthermore, the control architecture is implemented in a pseudo-distributed manner; that is, a workstation computer is utilized to calculate guidance commands, translate into control signals (i.e., the left and right wheel speeds) and send them to the robots, yet the information for generating guidance commands is regulated as if the robots were exchanging information according to the path graph.

Figure 6.4: Khepera IV robot.

In both experiments, the designed matrices are used exactly the same as presented in the numerical example of Section 6.3.5 for x and y directions. Moreover, the robots are initially placed at arbitrary positions. The targets are also two Khepera IV robots, which are respectively programmed to track the commands in x-direction as $c_{1x}(t) = 250 + 500\cos(0.1t)$ and $c_{2x}(t) = -300 + 500\cos(0.1t)$ and in y-direction as $c_{1y}(t) = 400 + 500\sin(0.1t)$ and $c_{2y}(t) = -150 + 500\sin(0.1t)$. Since the matrix M is Hurwitz and has $\mathrm{Re}(\lambda_{\max}(M)) = -0.382$, the choice of $\bar{\lambda} = 0.3$ for both experiments satisfies the condition $\bar{\lambda} \leq -\mathrm{Re}(\lambda_{\max}(M))$ discussed in Section 6.4.1. We note also that in the experiments the robots are set to stop when $t = T_s$.

6.4.2.1 First Experiment

In this experiment, the robots follow the first case discussed in Section 6.4.1; that is, the finite time convergence is given and the robots stop when they are sufficiently close to meet the objective. In particular, we choose the finite time convergence to be $T = 15$ seconds and $\delta = 0.002$, which is an alternative measurement for the error (see equation (6.44) for its definition). Then, using (6.45), one can find $T_s = 14.3292$ seconds. Figure 6.5 shows the trajectories of the robots and the targets during the mission (top plot) and the time evolution of the robots in x and y directions, respectively (bottom plots). It can be seen that the robots are split into two to track the two targets, and they are close to the target at time $t = T_s$. Figure 6.6 shows the control history of robots generated by the proposed control architecture, where the top plots are the control history of driver robots and the bottom plots are the total control history depicted by the right hand side of (6.2).

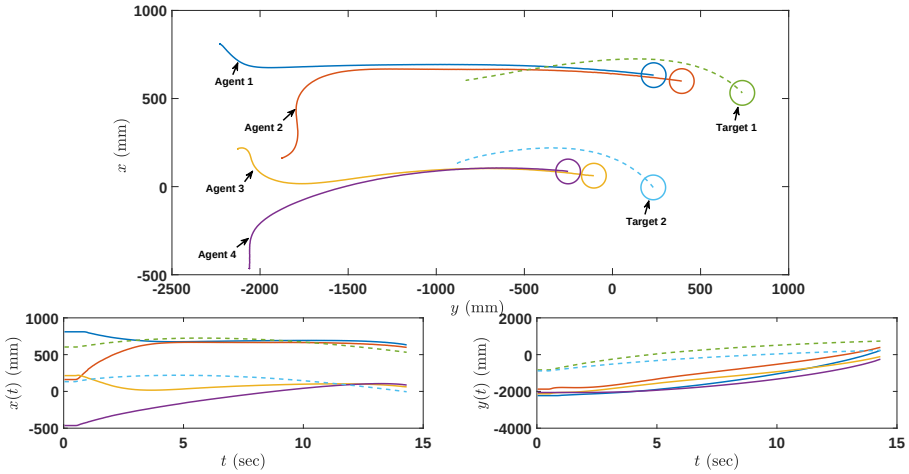


Figure 6.5: Trajectory of the robots and targets during the mission in Experiment 1 (top plot) and the time evolution of the robots in x and y directions (bottom plots).

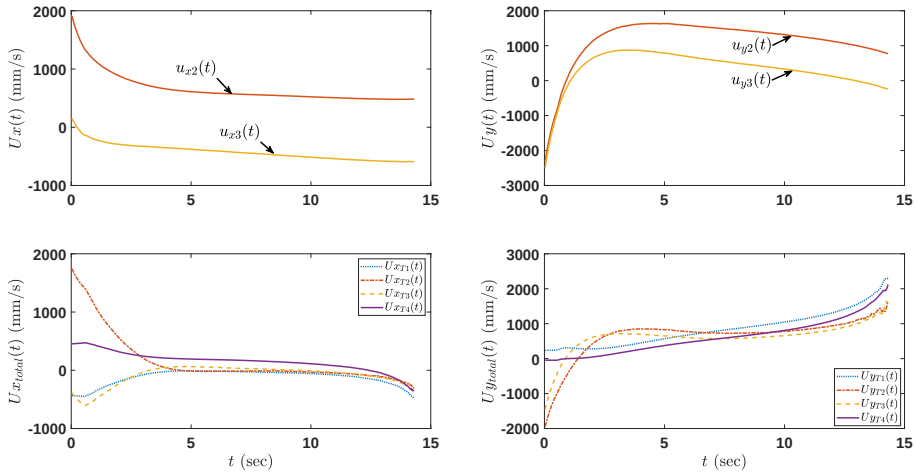


Figure 6.6: The high-level control history of robots generated by the proposed control architecture in Experiment 1. Top plots are the control history of driver robots. Bottom plots are the total control history of all robots depicted by the right hand side of (6.2).

6.4.2.2 Second Experiment

In this experiment, the robots follow the second case discussed in Section 6.4.1; that is, the upper bound of the finite time gain $\lambda(t)$ is assigned to be $\beta = 2$ and the robots stop when they are sufficiently close to meet the objective with $\delta = 0.002$. From (6.47), we need to choose $T_s \geq 10.6803$, and $T_s = 15$ is chosen for this case. Next, (6.48) is used to obtained the finite time convergence $T = 15.7022$ seconds for implementation purpose. Figure 6.7 shows the trajectories of the robots and the targets during the mission

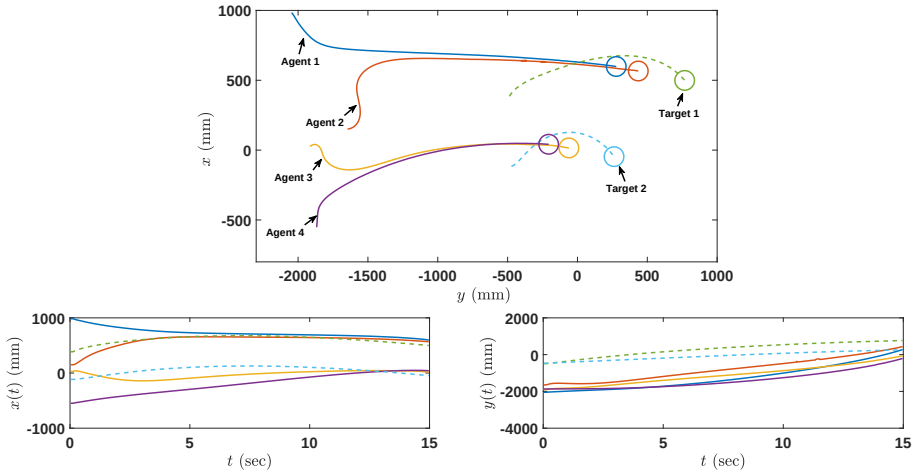


Figure 6.7: Trajectory of the robots and targets during the mission in Experiment 2 (top plot) and the time evolution of the robots in x and y directions (bottom plots).

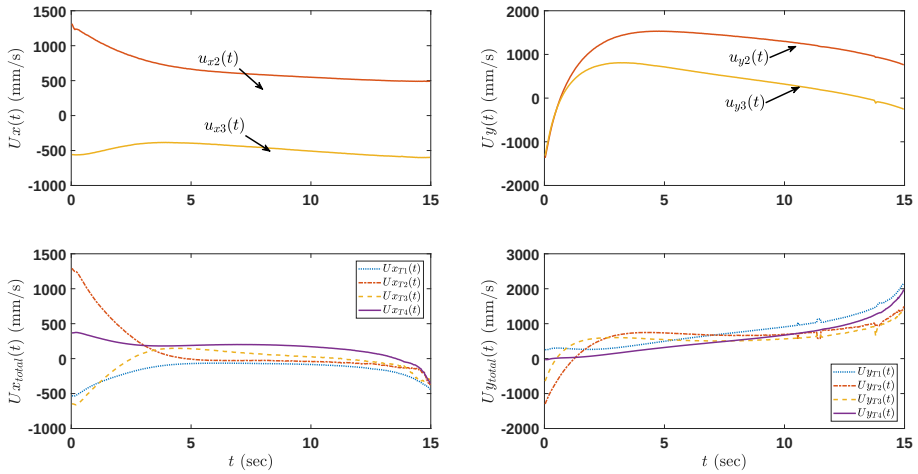


Figure 6.8: The high-level control history of robots generated by the proposed control architecture in Experiment 2. Top plots are the control history of driver robots. Bottom plots are the total control history of all robots depicted by the right hand side of (6.2).

(top plot) and the time evolution of the robots in x and y directions, respectively (bottom plots). It can be seen that the robots are split into two to track the two targets, and at the stop time $t = T_s$, they are close to the target. In addition, Figure 6.9 shows that the finite time gain $\lambda(t)$ does not exceed the given upper bound $\beta = 2$ during the mission. Figure 6.8 also shows the control history of robots generated by the proposed control architecture, where the top plots are the control history of driver robots and the bottom plots are the total control history depicted by the right hand side of (6.2).

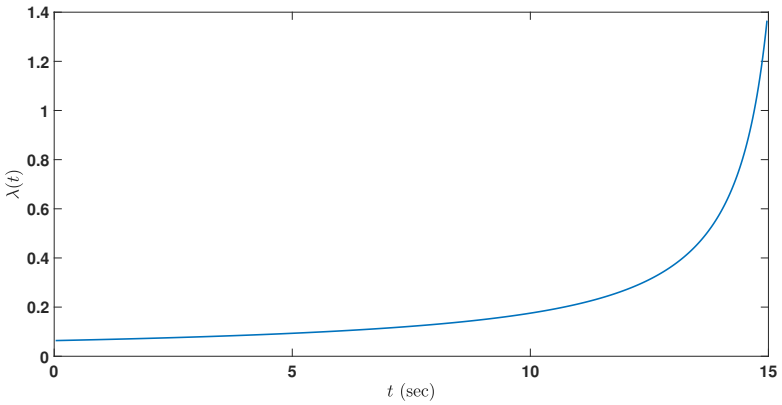


Figure 6.9: The evolution of the finite time gain $\lambda(t)$.

6.5 Conclusion

In this paper, we focused on multiagent networks as systems and proposed a new finite-time control algorithm using a recent time transformation method. Specifically, based on a given user-defined finite-time interval $[0,T)$, we showed that the proposed algorithm guarantees the time-critical completion of a given system-level control objective at T seconds regardless of the initial conditions of agents. In addition, it was shown that the separation principle holds for the proposed finite-time control algorithm in the sense that one can select the observer and controller gain matrices independently. Finally, numerical examples demonstrated the efficacy of our theoretical results.

Chapter 7: Finite-Time Control of Perturbed Dynamical Systems Based on a Generalized Time Transformation Approach*

We study finite-time control of perturbed dynamical systems based on the time transformation approach. For addressing time-critical applications, where the execution of a control algorithm over a prescribed time interval $[0,\tau)$ is necessary with τ being a user-defined convergence time, we introduce a new class of scalar, time-varying gain functions entitled as "generalized finite-time gain functions" that have the capability to convert an original baseline control algorithm into a time-varying one. Based on these generalized finite-time gain functions, in particular, the corresponding "generalized time transformation functions" are obtained and used to transform a resulting algorithm over the prescribed time interval $[0,\tau)$ to an equivalent algorithm over the stretched infinite-time interval $[0,\infty)$ for stability analysis, where the connection between the generalized finite-time gain functions and their corresponding generalized time transformation functions are investigated in detail. Specifically, we show all the conditions on the proposed generalized finite-time gain functions that guarantee the boundedness and convergence of the state and control signals. We also present an application of our theoretical findings to the distributed control of networked multiagent systems problem over a prescribed time interval.

7.1 Introduction

In many practical applications such as engagement of a guided missile with a target, landing of an aerial vehicle at a non-stationary carrier, and sequential execution of given complex tasks, finite-time control algorithms play an important role (see, for example, [26, 27, 40, 41] and the references therein). These time-critical applications are often performed over a time interval $[0,\tau)$, where the utilized finite-time control algorithms are expected to guarantee a task completion at a user-defined convergence time τ.

While there is a rich literature with regard to finite-time control, the finite-time convergence with the standard algorithms depends on initial conditions of dynamical systems (see, for example, [26–32] and

*This chapter has been submitted to the *System & Control Letters* for possible publication.

the references therein), and therefore, τ may not be readily assigned by a control designer. To provide a remedy to this problem, several results focus on finding an upper bound on the finite-time convergence (see, for example, [33–39] and references therein). Recently, there are also new results such as [40–54] (and references therein) that have the ability to directly assign a user-defined convergence time τ to the finite-time algorithms utilized in time-critical applications. For example, the authors of [53] propose a class of distributed control protocols to solve the consensus problem of linear multiagent systems within a pre-specified time by reducing the sampling time as time progresses. As another example, the authors of [54] propose a methodology for designing autonomous and non-autonomous pre-defined time systems; however, their results still require some knowledge on initial conditions. The results of this paper are particularly related to and generalize the recent studies in [40, 41] that utilize the time transformation approach.

In this paper, we study finite-time control of perturbed dynamical systems based on the time transformation approach. For addressing time-critical applications, where the execution of a control algorithm over a prescribed time interval $[0, \tau)$ is necessary with τ being a user-defined convergence time, we introduce a new class of scalar, time-varying gain functions entitled as “generalized finite-time gain functions" that have the capability to convert an original baseline control algorithm into a time-varying one. Based on these generalized finite-time gain functions, in particular, the corresponding “generalized time transformation functions" are obtained and used to transform a resulting algorithm over the prescribed time interval $[0, \tau)$ to an equivalent algorithm over the stretched infinite-time interval $[0, \infty)$ for stability analysis, where the connection between the generalized finite-time gain functions and their corresponding generalized time transformation functions are investigated in detail. Specifically, we show all the conditions on the proposed generalized finite-time gain functions that guarantee the boundedness and convergence of the state and control signals.

The organization of this paper is as follows. In Section 7.2, we state the necessary mathematical preliminaries and a key lemma for our main results. The proposed generalized time transformation functions-based finite-time control problem over the prescribed time interval $[0, \tau)$ is introduced and analyzed in Section 7.3. We also present an application of our theoretical findings to the distributed control of networked multiagent systems problem over a prescribed time interval in Section 7.4. Finally, our concluding remarks are summarized in Section 7.5. Note that a preliminary conference version of this paper is appeared in [131]. The present paper considerably expands on [131] by providing the detailed proofs of all the results together with additional remarks and discussions.

7.2 Mathematical Preliminaries

A standard mathematical notation is used in this paper. Specifically, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ denotes the set of $n \times 1$ real column vectors, $\mathbb{R}^{n\times m}$ denotes the set of $n \times m$ real matrices, $\mathbb{R}_+^{n\times n}$ (resp., $\overline{\mathbb{R}}_+^{n\times n}$) denotes the set of $n \times n$ positive-definite (resp., positive semi-definite) real matrices, $\mathbf{1}_n$ denotes the $n \times 1$ vector of all ones, and I_n denotes the $n \times n$ identity matrix. In addition, we write $(\cdot)^{\mathrm{T}}$ for transpose, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ respectively for the minimum and maximum eigenvalue of a matrix A, $\lambda_i(A)$ for the i-th eigenvalue of A, where A is symmetric and the eigenvalues are ordered from least to greatest value, $\mathrm{diag}(a)$ for the diagonal matrix with the vector a on its diagonal, $[x]_i$ for the entry of the vector x on the i-th row, and A_{ij} for the entry of the matrix A on the i-th row and j-th column.

We next summarize the basic graph-theoretical notions used in this paper (see, for example, [5] and [92] for details). In particular, an undirected graph $\mathcal{G}$ is defined by a set $\mathcal{E}_\mathcal{G} \subset \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ of edges and a set $\mathcal{V}_\mathcal{G} = \{1,\dots,N\}$ of nodes. We utilize $(i,j) \in \mathcal{E}_\mathcal{G}$ for cases when a pair of nodes i and j are neighbors, where $i \sim j$ indicates the neighboring relation. Furthermore, the degree of a node is determined by the number of its neighbors. Denoting d_i as the degree of node i, the degree matrix of a graph $\mathcal{G}$, $\mathcal{D}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is defined by $\mathcal{D}(\mathcal{G}) \triangleq \mathrm{diag}(d)$, $d = [d_1,\dots,d_N]^{\mathrm{T}}$. A path $i_0 i_1 \dots i_L$ is a finite sequence of nodes such that $i_{k-1} \sim i_k$, $k = 1,\dots,L$, and a graph $\mathcal{G}$ is said to be connected when there is a path between any pair of distinct nodes. The adjacency matrix of a graph $\mathcal{G}$, $\mathcal{A}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is defined by $[\mathcal{A}(\mathcal{G})]_{ij} = 1$ when $(i,j) \in \mathcal{E}_\mathcal{G}$ and $[\mathcal{A}(\mathcal{G})]_{ij} = 0$ otherwise. The Laplacian matrix of a graph, $\mathcal{L}(\mathcal{G}) \in \overline{\mathbb{R}}_+^{N\times N}$, is then defined by $\mathcal{L}(\mathcal{G}) \triangleq \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$.

Finally, the following key lemma from [119, Theorem 4.14] is necessary for the results in this paper.

Lemma 7.2.1. *For a given dynamical system $\dot{x}(t) = f\big(x(t)\big)$ with $f : \mathbb{R}^n \to \mathbb{R}^n$ being continuously differentiable over $D = \{\|x\|_2 < r\}$ and $x(t) \in \mathbb{R}^n$, let its origin be an exponentially stable equilibrium point. Furthermore, let k, λ, and r_0 be positive constants subject to $r_0 < r/k$ such that $\|x(t)\|_2 \le k\|x(0)\|_2 e^{-\lambda t}$ for all $x(0) \in D_0$ and $t \ge 0$, where $D_0 = \{\|x\|_2 < r_0\}$. Then, there is a continuously differentiable function $V(x)$ satisfying the inequalities given by*

$$c_1\|x\|_2^2 \le V(x) \le c_2\|x\|_2^2, \tag{7.1}$$

$$\frac{\partial V}{\partial x} f(x) \le -c_3\|x\|_2^2, \tag{7.2}$$

$$\Big\|\frac{\partial V}{\partial x}\Big\|_2 \le c_4\|x\|_2, \tag{7.3}$$

for all $x \in D_0$ with positive constants c_1, c_2, c_3, and c_4. If, in addition, f is continuously differentiable for all x, globally Lipchitz, and the origin is globally exponentially stable, then $V(x)$ is defined and satisfies the aforementioned inequalities for all $x \in \mathbb{R}^n$.

7.3 Generalized Time Transformation Approach-Based Finite-Time Control

Consider the perturbed dynamical system given by

$$\dot{x}(t) = \alpha(t) f\big(x(t)\big) + g(t, x(t)), \;\; x(0) = x_0, \tag{7.4}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $\alpha(t) \in \mathbb{R}_+$ is a positive and time-varying scalar function entitled as "generalized finite-time gain function" (details below), $g(t, x(t)) \in \mathbb{R}^n$ is a bounded perturbation term satisfying $\|g(t, x(t))\|_2 \leq g^*$, and $f\big(x(t)\big)$ is a continuously differentiable and globally Lipschitz function. In addition, let the origin of the nominal dynamical system $\dot{x}(t) = f\big(x(t)\big)$ be globally exponentially stable. Note that the nominal dynamical system $\dot{x}(t) = f\big(x(t)\big)$ can represent a controlled dynamics and it can be also considered as the error dynamics resulting from an original baseline control algorithm, where the perturbation is set to zero and $\alpha(t)$ is neglected as $\alpha(t) = 1$. To elucidate the latter point, we now provide an example.

Example 7.3.1. *Consider a simple-yet-illustrative baseline scalar command following control algorithm given by $\dot{z}(t) = u(t)$ with $u(t) = -\big(z(t) - c(t)\big)$, where $z(t)$ is the state, $u(t)$ is the control, and $c(t)$ is a time-varying bounded command with bounded time rate of change. Defining the error as $x(t) \triangleq z(t) - c(t)$, one can write the corresponding error dynamics in the form given by $\dot{x}(t) = -x(t) - \dot{c}(t)$. If $c(t)$ is constant (i.e., $\dot{c}(t) = 0$), then the error dynamics reduces to $\dot{x}(t) = -x(t)$, where this is the so-called nominal dynamical system with $f\big(x(t)\big) = -x(t)$ for this example. Here, if we choose the control as $u(t) = -\alpha(t)\big(z(t) - c(t)\big)$ through multiplying the right hand side of the baseline algorithm with the generalized finite-time gain function $\alpha(t)$, we obtain its time-varying version as $\dot{z}(t) = \alpha(t)\big(-z(t) + c(t)\big)$. In this case, the resulting error dynamics can be written in the form given by (7.4); that is, $\dot{x}(t) = \alpha(t)\big(-x(t)\big) - \dot{c}(t)$ with $g\big(t, x(t)\big) = -\dot{c}(t)$.*

The objective of this paper is to establish a class of generalized finite-time gain functions $\alpha(t)$ and the corresponding conditions in order to guarantee that the solution $x(t)$ of (7.4) converges to zero as $t \to \tau$,

where $\tau \in \mathbb{R}_+$ is a user-defined convergence time. Motivated by this standpoint, we first introduce the following assumption.

Assumption 7.3.1. *The generalized finite-time gain function $\alpha(t)$ satisfies the following properties:*

- *$\alpha(t)$ is continuous differentiable on $t \in [0, \tau)$.*
- *$\alpha(t) > m$ for all $t \in [0, \tau)$ and for some $m > 0$.*
- *$\lim_{t \to \tau} \alpha(t) = \infty$.*

If one chooses a generalized finite-time gain function $\alpha(t)$ according to Assumption 7.3.1, then its corresponding generalized time transformation function $t = \theta(s)$ can be obtained based on the next lemma.

Lemma 7.3.1. *Consider a generalized finite-time gain function $\alpha(t)$ subject to Assumption 7.3.1 and the following conditions:*

i) $\frac{\mathrm{d}t}{\mathrm{d}s} = \frac{\mathrm{d}(\theta(s))}{\mathrm{d}s} = \frac{1}{\alpha(\theta(s))}$ *(i.e,* $\alpha(\theta(s))\mathrm{d}(\theta(s)) = \mathrm{d}s$*).*

ii) $\theta(0) = 0$.

iii) $\lim_{s \to \infty} \theta(s) = \tau$.

If the generalized time transformation function $\theta(s)$ is obtained by solving the differential equation in i) along with the conditions ii) and iii), then the following statements hold:

a) $\theta(s)$ is continuous differentiable and strictly increasing over $s \in [0, \infty)$.

b) Let $h(s) \triangleq \frac{\mathrm{d}(\theta(s))}{\mathrm{d}s}$. Then, $h(s)$ is bounded and $\lim_{s \to \infty} h(s) = 0$.

Proof. To show *a)*, we need to show that $\mathrm{d}(\theta(s))/\mathrm{d}s$ is well-defined and greater than 0 over $s \in [0, \infty)$. We first note that the time transformation function $\theta(s)$ is actually a change of time from the stretched infinite-time interval $s \in [0, \infty)$ into the regular prescribed time interval $t \in [0, \tau)$; and vice versa. Therefore, $\alpha(t) = \alpha(\theta(s))$. In addition, from *i)*, we have

$$\frac{\mathrm{d}(\theta(s))}{\mathrm{d}s} = \frac{1}{\alpha(\theta(s))} = \frac{1}{\alpha(t)}. \tag{7.5}$$

Since $\alpha(t) > 0$ and it is continuous differentiable over $t \in [0, \tau)$, it is well-defined on $t \in [0, \tau)$; hence, $\mathrm{d}(\theta(s))/\mathrm{d}s > 0$ holds and also well-defined for all $s \in [0, \infty)$. As a result, $\theta(s)$ is continuous differentiable and strictly increasing over $s \in [0, \infty)$ (see, for example, [132]). Thus, the proof of *a)* is now complete.

To show *b)*, by the definition of $h(s)$ and (7.5), we have

$$h(s) = \frac{1}{\alpha(\theta(s))} = \frac{1}{\alpha(t)}. \tag{7.6}$$

Since $\alpha(t)$ is positive definite and lower bounded by m, $h(s)$ is upper bounded by $h^* \triangleq m^{-1}$. In addition, *ii)* and *iii)* indicates that $t \to \tau$ as $s \to \infty$ and recall that $\alpha(t)$ satisfies $\lim_{t\to\tau} \alpha(t) = \infty$. Therefore, we have

$$\lim_{s\to\infty} h(s) = \lim_{s\to\infty} \frac{1}{\alpha\big(\theta(s)\big)} = \lim_{t\to\tau} \frac{1}{\alpha(t)} = 0. \tag{7.7}$$

Note also that

$$\int_0^\infty |h(r)|\mathrm{d}r = \int_0^\infty h(r)\mathrm{d}r = \int_0^\tau \mathrm{d}\big(\theta(r)\big) = \int_0^\tau \mathrm{d}t = \tau. \tag{7.8}$$

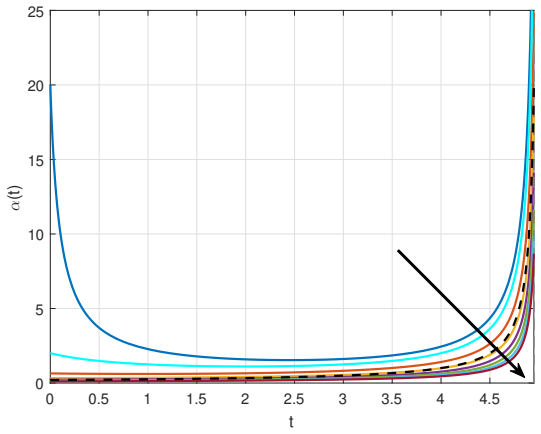
Thus, $h(s) \in \mathcal{L}_1$ on stretched infinite-time interval $s \in [0, \infty)$. Hence, the proof of *2)* is also now complete.

■

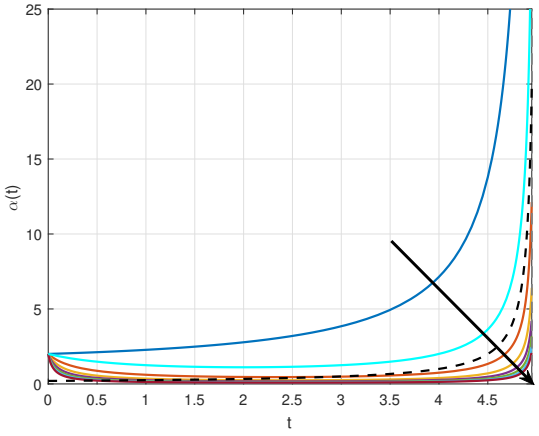
Remark 7.3.1. *Lemma 7.3.1 establishes the theoretical connection between the generalized finite-time gain functions and their corresponding generalized time transformation functions. Therefore, in a reverse manner to Lemma 7.3.1, one can also start with a time transformation function $t = \theta(s)$ that satisfies the conditions ii) and iii) and the properties a) and b) of Lemma 7.3.1, and then solve the differential equation i) to obtain the generalized finite-time gain function $\alpha(t)$ subject to Assumption 7.3.1. As an example, the authors of [40] and [41] choose $\theta(s) = \tau(1 - e^{-s})$ and then obtain $\alpha(t) = 1/(\tau - t)$.*

In order to elucidate Lemma 7.3.1, we next provide candidate generalized finite-time gain functions $\alpha(t)$.

Example 7.3.2. *A common finite-time gain function is $\alpha(t) = 1/(\tau - t)$ with its corresponding time transformation function $\theta(s) = \tau(1 - e^{-s})$. We now consider a family of generalized finite-time gain functions defined by $\alpha(t) \triangleq \frac{1}{(\tau-t)(mt+a)}$, where $m \in \mathbb{R}_+$ and $a \in \mathbb{R}_+$. Here, $\alpha(t)$ satisfies the conditions in Assumption 1. After solving the differential equation i) of Lemma 7.3.1, one can obtain $\theta(s) = \frac{\tau a(e^{(a+m\tau)s}-1)}{ae^{(a+m\tau)s}+m\tau}$ that satisfies*

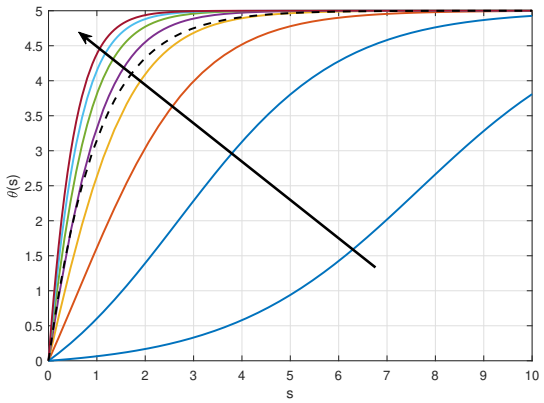


(a) $a \in [0.01, 2]$, $m = 0.1$, and $\tau = 5$.

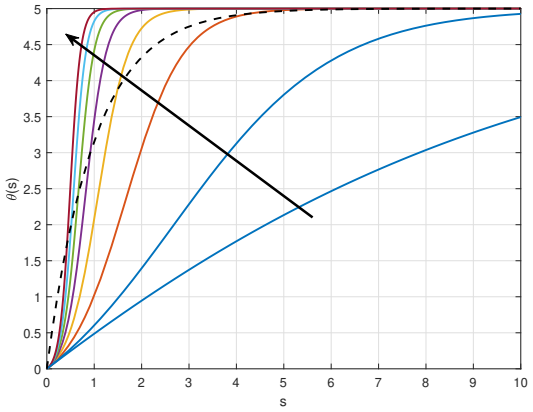


(b) $a = 0.1$, $m \in [0.01, 2]$ and $\tau = 5$.

Figure 7.1: Plots of the family of $\alpha(t)$ in Example 7.3.2 as a is increasing (left) and m is increasing (right), where the arrow pointing in the increasing direction of a and m. The dashed line represents $\alpha(t) = 1/(\tau - t)$ for comparison purpose.



(a) $a \in [0.01, 2]$, $m = 0.1$, and $\tau = 5$.



(b) $a = 0.1$, $m \in [0.01, 2]$, and $\tau = 5$

Figure 7.2: Plots of the family of $\theta(s)$ in Example 7.3.2 as a is increasing (left) and m is increasing (right), where the arrow pointing in the increasing direction of a and m. The dashed line represents $\theta(s) = \tau(1 - e^{-s})$ for comparison purpose.

the conditions ii) and iii). Figures 7.1 and 7.2 respectively show the plots of $\alpha(t)$ and the corresponding $\theta(s)$, as a is increasing while m is fixed (Figures 7.1(a) and 7.2(a)) and as m is increasing while a is fixed (Figures 7.1(b) and 7.2(b)). In these figures, the arrows point in the increasing direction of a and m. In addition, the common finite-time gain function $\alpha(t) = 1/(\tau - t)$ and its corresponding time transformation function $\theta(s) = \tau(1 - e^{-s})$ are also plotted for reference (dashed lines). Note that parameter a affects the initial gain of $\alpha(t)$ and parameter m affects the time rate of change of $\alpha(t)$ during the transient stage. Note

also that different generalized finite-time gain functions $\alpha(t)$ lead to different transient behaviors of the system, we refer to Section 7.4 for an illustrative numerical example.

Building on the result of Lemma 7.3.1, we now show the convergence of the solution of the perturbed dynamical system given by (7.4) to zero over the prescribed regular time interval $[0,\tau)$.

Theorem 7.3.1. *Consider the perturbed dynamical system given by (7.4). If the generalized finite-time gain function $\alpha(t)$ satisfies Assumption 7.3.1 and there exists a corresponding generalized time transformation function $\theta(s)$ as stated in Lemma 7.3.1, then* $\lim_{t\to\tau} x(t) = 0$.

Proof. Since $x(t) = x(\theta(s))$, define $\bar{x}(s) \triangleq x(\theta(s))$. Then, the perturbed dynamical system given by (7.4) can be rewritten in the stretched infinite-time interval $s \in [0,\infty)$ as

$$\begin{aligned}\bar{x}'(s) \triangleq \frac{\mathrm{d}\bar{x}(s)}{\mathrm{d}s} &= \frac{\mathrm{d}\theta(s)}{\mathrm{d}s}\frac{\mathrm{d}\bar{x}(s)}{\mathrm{d}\theta(s)} \\ &= \frac{1}{\alpha(\theta(s))}\Big(\alpha(\theta(s))f\big(\bar{x}(s)\big) + g\big(\theta(s),\bar{x}(s)\big)\Big) \\ &= f\big(\bar{x}(s)\big) + \frac{1}{\alpha(\theta(s))}g\big(\theta(s),\bar{x}(s)\big) \\ &= f\big(\bar{x}(s)\big) + h(s)g\big(\theta(s),\bar{x}(s)\big),\ \bar{x}(0) = x_0, \end{aligned} \tag{7.9}$$

where $h(s) \triangleq 1/\alpha(\theta(s)) = 1/\alpha(t) = \mathrm{d}(\theta(s))/\mathrm{d}s$ as shown in Lemma 7.3.1. Define now $p(s,\bar{x}(s)) \triangleq h(s)g\big(\theta(s),\bar{x}(s)\big) \in \mathbb{R}^n$. Note that both $h(s)$ and $g\big(\theta(s),\bar{x}(s)\big)$ are respectively bounded by h^* and g^*; hence, $p(s,\bar{x}(s))$ is also bounded; that is, $\|p(s,\bar{x}(s))\|_2 \leq p^* \triangleq h^*g^*$. Now, one can rewrite (7.9) as

$$\bar{x}'(s) = f\big(\bar{x}(s)\big) + p(s,\bar{x}(s)),\ \ \bar{x}(0) = x_0. \tag{7.10}$$

Since the origin of the nominal dynamical system $\dot{x}(t) = f\big(x(t)\big)$ of (7.4) is globally exponentially stable, the result of this theorem follows directly from Lemma 4.6 of [119]. Yet, we explicitly derive it here for the further analysis later. Specifically, for the nominal dynamical system, there exists a continuous function $V(x)$ satisfying the inequalities (7.1), (7.2) and (7.3) by Lemma 7.2.1. Utilizing this Lyapunov function and taking its derivative with respect to $s \in [0,\infty)$ along the trajectories of (7.10), we have

$$
\begin{aligned}
V'(\bar{x}) &= \frac{\partial V}{\partial \bar{x}}\Big(f\big(\bar{x}(s)\big) + p(s,\bar{x}(s))\Big) \\
&= \frac{\partial V}{\partial \bar{x}} f\big(\bar{x}(s)\big) + \frac{\partial V}{\partial \bar{x}} p(s,\bar{x}(s)) \\
&\leq -c_3\|\bar{x}(s)\|_2^2 + c_4\|\bar{x}(s)\|_2\|p(s,\bar{x}(s))\|_2 \\
&\leq -(1-\theta)c_3\|\bar{x}(s)\|_2^2 - \theta c_3\|\bar{x}(s)\|_2^2 + c_4\|\bar{x}(s)\|_2\|p(s,\bar{x}(s))\|_2 \\
&\leq -(1-\theta)c_3\|\bar{x}(s)\|_2^2, \ \forall\ \|\bar{x}(s)\|_2 \geq \frac{c_4\|p(s,\bar{x}(s))\|_2}{\theta c_3},
\end{aligned} \tag{7.11}
$$

where $\theta \in (0,1)$ and the third inequality comes from (7.2) and (7.3). By Theorem 4.19 of [119], the system (7.10) is input-to-state stable. Note that input-to-state stability implies that when the input converges to zero as $s \to \infty$, so does the state (see, for example, Exercise 4.58 in [119]). From Lemma 7.3.1, $\lim_{s\to\infty} h(s) = 0$; hence, $\lim_{s\to\infty} p(s,\bar{x}) = 0$. As a result, $\bar{x}(s) \to 0$ as $s \to \infty$. Finally, since $t \to \tau$ as $s \to \infty$, $\lim_{t\to\tau} x(t) = 0$ follows. ■

Remark 7.3.2. *Although the dynamical system given by (7.4) is perturbed, Theorem 7.3.1 shows that its state vector still converges to zero in a user-defined convergence time τ owing to the generalized finite-time gain function $\alpha(t)$. In addition, the perturbed dynamical system given by (7.4) often represents the error dynamics as discussed in Example 7.3.1 (see also, for example, [40, 41, 49] and references therein). In particular, the dynamics in, for example, [40, 41, 49] are linear; hence, the origins of their nominal systems are globally exponentially stable and readily satisfy the conditions of the perturbed dynamical system given by (7.4). Thus, for time-critical applications, if one designs a control algorithm for the dynamical system such that its error dynamics can be put into the form given by (7.4), its finite-time convergence is then guaranteed. To summarize, consistent with the discussion given in Example 7.3.1, the following three-step procedure can be adopted for designing a control algorithm for time-critical applications:*

- *Design a baseline control algorithm to exponentially satisfy the given objectives of a considered application over $[0,\infty)$.*
- *Find a generalized finite-time gain function $\alpha(t)$ that satisfies Assumption 7.3.1 and its corresponding generalized time transformation function $\theta(s)$ along the lines stated in Lemma 7.3.1.*
- *Obtain then the finite-time control algorithm over $[0,\tau)$ by multiplying the baseline control algorithm with the generalized finite-time gain function $\alpha(t)$.*

Remark 7.3.3. *From the third line in (7.11), we have*

$$V(\bar{x}(\infty)) - V(\bar{x}(0)) \leq -c_3 \int_0^\infty \|\bar{x}(r)\|_2^2 \mathrm{d}r + c_4 \int_0^\infty \|\bar{x}(r)\|_2 \|p\big(r,\bar{x}(r)\big)\|_2 \mathrm{d}r, \tag{7.12}$$

where $V(\bar{x}(\infty)) \triangleq \lim_{s\to\infty} V(\bar{x}(s)) = V(0) = 0$ *from the result of Theorem 7.3.1. In addition, the result of Theorem 7.3.1 indicates that* $\bar{x}(s)$ *is bounded; hence, we can consider* $\|\bar{x}(s)\|_2 \leq \bar{x}^*$*. Therefore, from (7.12), one can write*

$$\begin{aligned}
\int_0^\infty \|\bar{x}(r)\|_2^2 \mathrm{d}r &\leq \frac{1}{c_3}\Big(V(\bar{x}(0)) + c_4\bar{x}^* \int_0^\infty \|p\big(r,\bar{x}(r)\big)\|_2 \mathrm{d}r\Big) \\
&\leq \frac{1}{c_3}\Big(V(\bar{x}(0)) + c_4\bar{x}^* \int_0^\infty |h(r)| \|g\big(\theta(r),\bar{x}(r)\big)\|_2 \mathrm{d}r\Big) \\
&\leq \frac{1}{c_3}\Big(V(\bar{x}(0)) + c_4\bar{x}^* g^* \int_0^\infty |h(r)| \mathrm{d}r\Big) \\
&= \frac{V(\bar{x}(0))}{c_3} + \frac{c_4\bar{x}^* g^* \tau}{c_3},
\end{aligned} \tag{7.13}$$

where the last equality results from (7.8). We note that the left hand side of (7.13) can be considered as an energy of $\bar{x}(s)$*, and therefore is of* $x(t)$*, and the right hand side of (7.13) is a constant. Thus,* $\bar{x}(s)$ *has a finite energy signal. The upper bound of this energy depends on* $V(\bar{x}(0))$ *and the product of* τ *with the bounds of both the state and the perturbation term. In particular, if the user-defined convergence time* τ *is selected to be large, then the upper bound of this energy given by the right hand side of (7.13) also increases to suppress the effect of the perturbation term. Yet,* $V(\bar{x}(0))$ *is independent of* τ *and can be interpreted as the energy required to drive the system from* $\bar{x}_0$ *to* 0*. In other words, when the perturbations on the dynamics are negligible, the system requires a fixed amount of energy to go from an initial value to zero equilibrium point regardless of the chosen of* τ*. One can also consider the limit of a dynamical system's actuator when choosing* τ*. A workaround to prevent exceeding an actuator's performance is discussed in the next remark.*

Remark 7.3.4. *A practical approach to overcome the problem of exceeding actuator's performance is to first drive the system to a point or a region, where we know that it is feasible to achieve the objective in a user-defined convergence time* τ *without exceeding actuator's limit, and then activate the finite-time algorithm. Consider now that a feasible region* Ψ *is theoretically defined. The generalized finite-time gain*

function can then be redefined as

$$\beta(t) \triangleq \begin{cases} \alpha(0), & x \notin \Psi, \\ \alpha(t-t_0), & x \in \Psi, \end{cases} \tag{7.14}$$

where t_0 is the time when the system enters the region Ψ providing that the system is capable of keeping track of execution time and detecting whether or not it is in Ψ. Note that $\beta(t)$ is a continuous function and identical to $\alpha(t)$ when $t_0 = 0$. By replacing $\alpha(t)$ by $\beta(t)$ in (7.4), the total execution time of the system is now $t_0 + \tau$. In particular, for the first t_0 seconds, the dynamics is time-invariant and is heading toward the feasible region Ψ. For $t \geq t_0$, the dynamics becomes time-varying and meets the objective in τ seconds. An example of the feasible region is $\Psi = \{x(t) \in \mathbb{R}^n : \|x\|_\infty \leq x^*\}$ with $x^* \in \mathbb{R}_+$ being a known defined threshold.

Finally, the next theorem establishes the boundedness of $\dot{x}(t)$ over $t \in [0, \tau)$.[15]

Theorem 7.3.2. *Consider the perturbed dynamical system given by (7.4). Consider, in addition, the following conditions:*

i) $\frac{\dot{\alpha}(t)}{\alpha^2(t)}$ *is bounded on* $t \in [0, \tau)$, *and* $\lim_{t \to \tau} \frac{\dot{\alpha}(t)}{\alpha^2(t)} = \kappa < \infty$.

ii) $\bar{r}'(s) = \left(\frac{\mathrm{d}f\left(\bar{x}(s)\right)}{\mathrm{d}\bar{x}} + \frac{\mathrm{d}\alpha(\theta(s))}{\mathrm{d}\theta(s)} h^2(s) \mathrm{I}_n \right) \bar{r}(s)$ *is globally exponentially stable, where* $r(t) = r\big(\theta(s)\big)$ *and* $\bar{r}(s) \triangleq r\big(\theta(s)\big)$.

Then, $\dot{x}(t)$ is bounded for $t \in [0, \tau)$.

Proof. From (7.4), if we show that $r(t) \triangleq \alpha(t) f\big(x(t)\big)$ is bounded over $t \in [0, \tau)$, then we can directly conclude that $\dot{x}(t)$ is bounded over $t \in [0, \tau)$ since $g\big(t, x(t)\big)$ is bounded over $t \in [0, \tau)$. For this purpose, we now write the time derivative of $r(t)$ as

$$\begin{aligned} \dot{r}(t) &= \dot{\alpha}(t) f\big(x(t)\big) + \alpha(t) \dot{f}\big(x(t)\big) \\ &= \dot{\alpha}(t) f\big(x(t)\big) + \alpha(t) \frac{\mathrm{d}f\big(x(t)\big)}{\mathrm{d}x} \frac{\mathrm{d}x}{\mathrm{d}t} \\ &= \dot{\alpha}(t) \left(\frac{1}{\alpha(t)} r(t) \right) + \alpha(t) \frac{\mathrm{d}f\big(x(t)\big)}{\mathrm{d}x} \big(r(t) + g(t, x(t)) \big) \\ &= \left(\alpha(t) \frac{\mathrm{d}f\big(x(t)\big)}{\mathrm{d}x} + \frac{\dot{\alpha}(t)}{\alpha(t)} \mathrm{I}_n \right) r(t) + \alpha(t) \frac{\mathrm{d}f\big(x(t)\big)}{\mathrm{d}x} g(t, x(t)), \end{aligned} \tag{7.15}$$

[15] Similar to the discussion in Example 7.3.1, if we consider the control signal as $u(t) = \alpha(t) f\big(x(t)\big)$, then it is also bounded as a result of the following theorem as well as from the boundedness of $g\big(t, x(t)\big)$.

where the third equality comes from the definition of $r(t)$ and (7.4). Since $r(t) = r\big(\theta(s)\big)$, define $\bar{r}(s) \triangleq r\big(\theta(s)\big)$. Similar to the proof of Theorem 7.3.1, one can rewrite (7.15) in the stretched infinite-time interval $s \in [0,\infty)$ as

$$\bar{r}'(s) = \left(\frac{\mathrm{d}f\big(\bar{x}(s)\big)}{\mathrm{d}\bar{x}} + \frac{\mathrm{d}\alpha(\theta(s))}{\mathrm{d}\theta(s)} h^2(s)\mathrm{I}_n\right)\bar{r}(s) + \frac{\mathrm{d}f\big(\bar{x}(s)\big)}{\mathrm{d}\bar{x}} g\big(\theta(s),\bar{x}(s)\big), \tag{7.16}$$

where $h(s) \triangleq 1/\alpha(\theta(s))$. Since $f\big(x(t)\big)$ is globally Lipschitz, the second term of (7.16) is bounded. In addition, by conditions *i)* and *ii)*, we conclude that $\bar{r}(s)$ is a bounded solution to the dynamical system (7.16) on the stretched infinite-time interval $s \in [0,\infty)$. Therefore, $r(t)$ is bounded over $t \in [0,\tau)$, where this implies the boundedness of $\dot{x}(t)$ over $t \in [0,\tau)$. ■

Remark 7.3.5. *In the proof of Theorem 7.3.2, $\dot{x}(t)$ is bounded when $r(t) = \alpha(t)f\big(x(t)\big)$ is bounded. Note that $\lim_{t\to\tau}\alpha(t) = \infty$ by Assumption 7.3.1 and $\lim_{t\to\tau} f\big(x(t)\big) = 0$ by Theorem 7.3.1. Therefore, intuitively, $r(t)$ is bounded when $f\big(x(t)\big)$ decays to zero before $\alpha(t)$ approaches to ∞. Indeed, assumptions i) and ii) of Theorem 7.3.2 capture this phenomenon. Specifically, the term $\alpha(t)\frac{\mathrm{d}f\left(x(t)\right)}{\mathrm{d}x}$ in (7.15) represents the rate of change of $r(t)$ along $f\big(x(t)\big)$ and the term $\frac{\dot{\alpha}(t)}{\alpha(t)}\mathrm{I}_n$ in (7.15) is the rate of change of $r(t)$ along $\alpha(t)$. When transforming these terms into the stretched infinite-time interval $s \in [0,\infty)$, the assumption ii) apprehends the above requirement. In addition, we note that*

$$\frac{\mathrm{d}\alpha(\theta(s))}{\mathrm{d}\theta(s)} = \frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} = \dot{\alpha}(t), \tag{7.17}$$

while $h^2(s) = 1/\alpha^2(t)$; hence, $\frac{\mathrm{d}\alpha(\theta(s))}{\mathrm{d}\theta(s)}h^2(s) = \frac{\dot{\alpha}(t)}{\alpha^2(t)}$. This induces assumption i). We refer to Remark 7.3.7 and Figure 7.3 for an illustration of this point.

Remark 7.3.6. *The conditions i) and ii) of Theorem 7.3.2 are the generalized forms of the conditions in, for example, [40, 41] and [42]. Specifically, in these papers, $\alpha(t) = 1/(\tau - t)$ and $\dot{\alpha}(t) = \alpha^2(t)$; hence, the condition i) automatically holds. In addition, these papers consider linear systems; thus, $\mathrm{d}f\big(\bar{x}(s)\big)/\mathrm{d}\bar{x}$ is often depicted by a Hurwitz matrix such as $\mathrm{d}f\big(\bar{x}(s)\big)/\mathrm{d}\bar{x} = \gamma M$ with $\gamma \in \mathbb{R}_+$ being a design parameter. As a consequence, assumption ii) simplifies to the requirement that the matrix $(\gamma M + \mathrm{I}_n)$ is Hurwitz. In this case, since M is Hurwitz, it is straightforward to show that by choosing $\gamma > -1/\lambda_{\max}(M)$, the matrix $(\gamma M + \mathrm{I}_n)$ is guaranteed to be Hurwitz.*

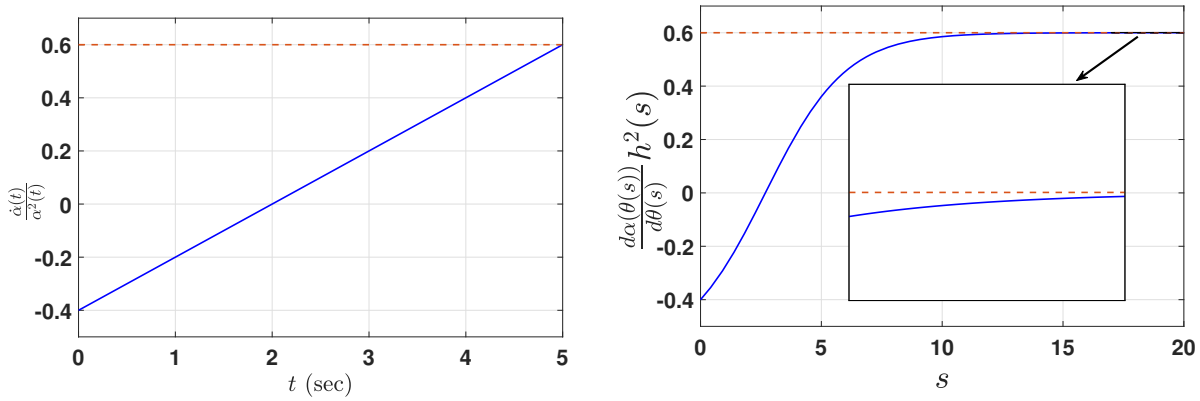


Figure 7.3: Plot of $\frac{\dot{\alpha}(t)}{\alpha^2(t)}$ over the regular prescribed time interval $[0,\tau)$ (left) and plot of its identical version $\frac{\mathrm{d}\alpha(\theta(s))}{\mathrm{d}\theta(s)}h^2(s)$ over the stretched infinite-time interval $[0,\infty)$ (right) for $\alpha(t) \triangleq \frac{1}{(\tau-t)(mt+a)}$ with $a=0.1$, $m=0.1$, and $\tau=5$. The dashed line represents the upper bound κ.

Remark 7.3.7. *Note for the candidate family of generalized finite-time gain functions utilized in Example 7.3.2,* $\alpha(t) \triangleq \frac{1}{(\tau-t)(mt+a)}$, *that* $\dot{\alpha}(t) = (2mt - m\tau + a)\alpha^2(t)$; *hence, the condition i) of Theorem 7.3.2 is satisfied with* $\kappa = m\tau + a$ *and this is also the upper bound of* $\dot{\alpha}(t)/\alpha^2(t)$ *over* $t \in [0,\tau)$. *Note also that* $\lim_{s\to\infty} \frac{\mathrm{d}\alpha(\theta(s))}{\mathrm{d}\theta(s)}h^2(s) = \kappa$. *To illustrate this point, Figure 7.3 shows the plot of* $\frac{\dot{\alpha}(t)}{\alpha^2(t)}$ *over the regular prescribed time interval* $[0,\tau]$ *(left) and the plot of its identical version* $\frac{\mathrm{d}\alpha(\theta(s))}{\mathrm{d}\theta(s)}h^2(s)$ *over the stretched infinite-time interval* $[0,\infty)$ *(right) with* $a=0.1$, $m=0.1$, *and* $\tau=5$, *and the dashed line represents* κ. *In addition, similar to Remark 7.3.6, as applied to linear systems with* $\mathrm{d}f\big(\bar{x}(s)\big)/\mathrm{d}\bar{x} \triangleq \gamma M$ *being a Hurwitz matrix, the assumption ii) is satisfied when the matrix* $(\gamma M + \kappa \mathrm{I}_n)$ *is Hurwitz (Interested readers can refer to Example 9.6, Corollary 9.1 and Lemma 9.5 of [119] for similar analysis). Once again, since M is Hurwitz, when* $\gamma > -\kappa/\lambda_{\max}(M)$, $(\gamma M + \kappa \mathrm{I}_n)$ *is guaranteed to be Hurwitz.*

7.4 Finite-Time Distributed Control of Networked Multiagent Systems

During the last two decades, networked multiagent systems have started to become a relatively mature research field addressing important problems through local interactions such as consensus, leader-follower, flocking, formation control, containment control; to name but a few examples (see, for example, [5, 76, 133, 134] and references therein). In this section, we present a theoretical application of our findings in Section 7.3 to the distributed control of networked multiagent systems to illustrate their efficacy. In particular, we consider the leader-follower problem with a networked multiagent system containing N agents

subject to a connected and undirected graph $\mathcal{G}$. To this end, let the baseline distributed control algorithm of agents be (see, for example, [16])

$$\dot{x}_i(t) \quad = \quad -\gamma\left(\sum_{i\sim j}\big(x_i(t)-x_j(t)\big)+k_i\big(x_i(t)-c(t)\big)\right), \;\; x_i(0)=x_{i0}, \tag{7.18}$$

where $x_i(t)\in\mathbb{R}$ is the state of agent i, $i=1,\ldots,N$, $c(t)\in\mathbb{R}$ is the bounded command with bounded time rate of change, $\gamma\in\mathbb{R}_+$ is a scalar gain, $k_i=1$ if agent i is the leader, and $k_i=0$ otherwise.

Defining the error as $\tilde{x}_i(t)\triangleq x_i(t)-c(t)$, one can write

$$\dot{\tilde{x}}_i(t) \quad = \quad -\gamma\left(\sum_{i\sim j}\big(\tilde{x}_i(t)-\tilde{x}_j(t)\big)+k_i\tilde{x}_i(t)\right)-\dot{c}(t), \quad \tilde{x}_i(0)=\tilde{x}_{i0}. \tag{7.19}$$

By defining $\tilde{x}(t)\triangleq[\tilde{x}_1(t),\ldots,\tilde{x}_N(t)]^{\mathrm{T}}$, one can obtain the compact form of the error dynamics in (7.19) as

$$\dot{\tilde{x}}(t)=-\gamma(\mathcal{L}(\mathcal{G})+K)\tilde{x}(t)-\mathbf{1}_N\dot{c}(t), \;\; \tilde{x}(0)=\tilde{x}_0, \tag{7.20}$$

where $\mathcal{L}(\mathcal{G})\in\mathbb{R}^{N\times N}$ is the Laplacian matrix of the communication graph $\mathcal{G}$ and $K=\mathrm{diag}([k_1,k_2,\ldots,k_N]^{\mathrm{T}})$. Here, $-(\mathcal{L}(\mathcal{G})+K)$ is a Hurwitz matrix (see, for example, Lemma 3.3 of [16]).

Next, by multiplying the baseline algorithm in (7.18) with the generalized finite-time gain function $\alpha(t)$, we obtain a time-varying distributed control algorithm for time-critical applications in the form

$$\dot{x}_i(t)=u_i(t), \;\; x_i(0)=x_{i0}, \tag{7.21}$$

$$u_i(t)=-\gamma\alpha(t)\left(\sum_{i\sim j}\big(x_i(t)-x_j(t)\big)+k_i\big(x_i(t)-c(t)\big)\right). \tag{7.22}$$

In this case, the resulting error dynamics becomes

$$\dot{\tilde{x}}(t)=-\gamma\alpha(t)\,(\mathcal{L}(\mathcal{G})+K)\tilde{x}(t)-\mathbf{1}_N\dot{c}(t), \;\; \tilde{x}(0)=\tilde{x}_0, \tag{7.23}$$

which clearly satisfies the form of the perturbed dynamical system given by (7.4) with $f\big(x(t)\big)=-\gamma(\mathcal{L}(\mathcal{G})+K)\tilde{x}(t)$ and $g\big(t,x(t)\big)=-\mathbf{1}_N\dot{c}(t)$. The next corollary is now immediate.

Corollary 7.4.1. *Consider the networked multiagent system with N agents given by (7.21) and (7.22), where agents exchange information using their local measurements through an undirected and connected graph*

topology $\mathcal{G}$. In addition, consider that the generalized finite-time gain function $\alpha(t)$ satisfies Assumption 7.3.1, $\lim_{t\to\tau} \frac{\dot{\alpha}(t)}{\alpha^2(t)} = \kappa < \infty$*, and there exists a corresponding generalized time transformation function $\theta(s)$ as stated in Lemma 7.3.1. By defining* $M \triangleq -(\mathcal{L}(\mathcal{G}) + K)$ *and choosing* $\gamma > -\kappa/\lambda_{\max}(M)$*, then* $\lim_{t\to\tau} x(t) = c(\tau)$ *holds and $\dot{x}(t)$ is bounded for $t \in [0, \tau)$.*

Proof. The results directly follows from Theorems 7.3.1 and 7.3.2 as well as Remark 7.3.7. ■

We are now ready to present an example to illustrate the result in Corollary 7.4.1.

Example 7.4.1. *In this example, we consider a multiagent system under the algorithm given by (7.21) and (7.22) with 5 agents subject to a ring graph, where the first agent is the leader and the rest are followers. Here, we choose the user-defined convergence time as $\tau = 5$ seconds and the command as $c(t) = 5 + 0.75sin(t)$. Along the lines of the discussion in Example 7.3.2 of Section 7.3, the family of generalized finite-time gain functions defined by $\alpha(t) \triangleq \frac{1}{(\tau - t)(mt + a)}$ that satisfies Assumption 1 and there exists a corresponding family of generalized time transformation functions $\theta(s)$ as stated in Lemma 7.3.1. In what follows, we consider the two cases: We choose $a = 0.5$ and $m = 0.005$ for the first case and choose $a = 0.1$ and $m = 0.085$ for the second case. Based on Remark 7.3.7, note that $M = -(\mathcal{L}(\mathcal{G}) + K)$ is Hurwitz and the upper bounded of $\dot{\alpha}(t)/\alpha^2(t)$ on $t \in [0, \tau)$ is $\kappa = m\tau + a = 0.525$ for both cases; hence, we obtain $-\kappa/\lambda_{\max}(M) = 3.7717$. By choosing $\gamma = 4$ for both cases, the assumptions of Theorem 7.3.2 are now satisfied; hence, $\dot{x}(t)$ is bounded over $t \in [0, \tau)$. The same random initial conditions for agents are utilized for both cases.*

Figure 7.4 shows the response of the networked multiagent system under the control algorithm given by (7.22) with $\alpha(t) \triangleq \frac{1}{(\tau - t)(mt + a)}$, $a = 0.5$, $m = 0.005$, $\tau = 5$ seconds, and $\gamma = 4$, where the solid lines are the states of agents (left) and their time derivative (right), and the dashed line shows the command. Similarly, Figure 7.5 shows the response of the networked multiagent system under algorithm (7.22) with $\alpha(t) \triangleq \frac{1}{(\tau - t)(mt + a)}$, $a = 0.1$, $m = 0.085$, $\tau = 5$ seconds, and $\gamma = 4$, where the solid lines are the states of agents (left) and their time derivative (right), and the dashed line shows the command. As expected from the result of Corollary 7.4.1, the states of all agents approach to the command $c(t)$ as $t \to \tau = 5$ seconds. Note that when $t = 0$, $\alpha(0) = 1/(\tau a)$; hence, the parameter a affects the initial value of $\alpha(t)$. Moreover, from discussion in Remark 7.3.7, $\dot{\alpha}(t) = (2mt - m\tau + a)\alpha^2(t)$; hence, m affects the time rate of change of $\alpha(t)$. Since the second case has a smaller value for a, the initial value of $\alpha(t)$ in the second case is larger than in the first case. This is depicted by the higher initial values of $\dot{x}(t)$ in the right plot of Figure 7.5 compared to the one in Figure 7.4. Finally, as expected from Theorem 7.3.2, the right plots show that $\dot{x}_i(t)$ remains

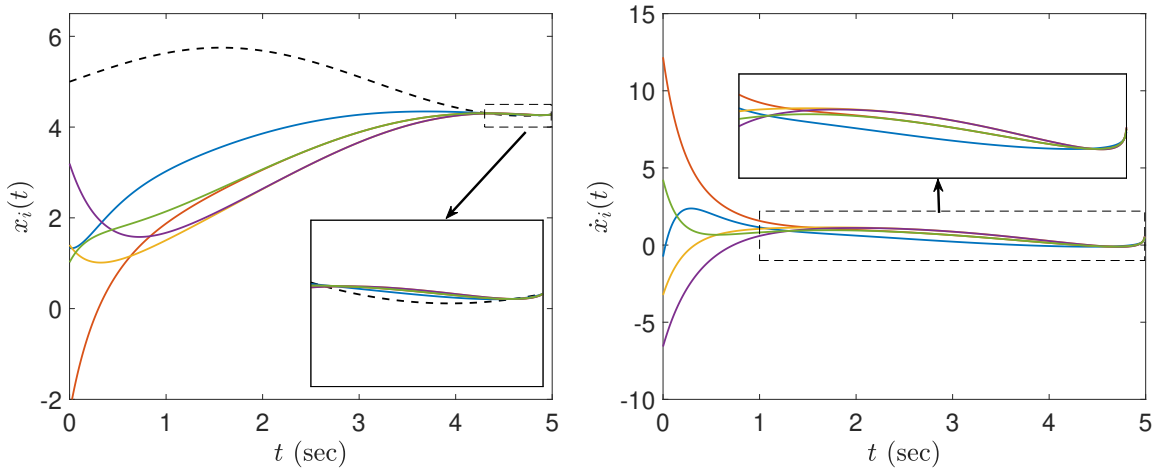


Figure 7.4: Response of the networked multiagent system under algorithm (7.22) with $\alpha(t) \triangleq \frac{1}{(\tau-t)(mt+a)}$, $a = 0.5$, $m = 0.005$, $\tau = 5$ seconds, and $\gamma = 4$, where the solid lines are the states of agents (left) and the time derivative of agents (right), and the dashed line shows the tracking command.

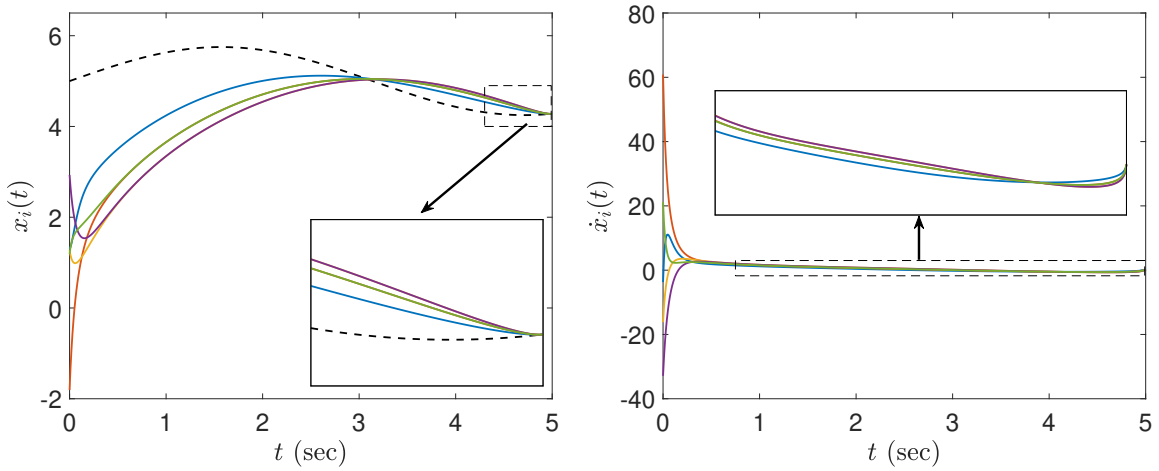


Figure 7.5: Response of the networked multiagent system under algorithm (7.22) with $\alpha(t) \triangleq \frac{1}{(\tau-t)(mt+a)}$, $a = 0.1$, $m = 0.085$, $\tau = 5$ seconds, and $\gamma = 4$, where the solid lines are the states of agents (left) and the time derivative of agents (right), and the dashed line shows the tracking command.

bounded over $t \in [0, \tau)$. In general, different values of a and m in both cases lead to different transient behaviors of the resulting networked multiagent system.

7.5 Conclusion

For contributing to the recent studies on finite-time control based on the time transformation approach, we investigated a new class of scalar, time-varying gain functions entitled as "generalized finite-

time gain functions". We showed how these functions have the capability to convert an original baseline control algorithm into a time-varying one to allow for its execution over a prescribed time interval. We also rigorously showed all the stability conditions with regard to the proposed family of generalized finite-time gain functions that guarantee the boundedness and convergence of the state and control signals when the considered class of dynamical systems are subject to perturbations. Finally, we presented an application of our theoretical findings to distributed control of networked multiagent systems over a prescribed time interval.

Chapter 8: Distributed Input and State Estimation Using Local Information in Heterogeneous Sensor Networks*

A new distributed input and state estimation architecture is introduced and analyzed for heterogeneous sensor networks. Specifically, nodes of a given sensor network are allowed to have heterogeneous information roles in the sense that a subset of nodes can be active (that is, subject to observations of a process of interest) and the rest can be passive (that is, subject to no observation). Both fixed and varying active and passive roles of sensor nodes in the network are investigated. In addition, these nodes are allowed to have nonidentical sensor modalities under the common underlying assumption that they have complimentary properties distributed over the sensor network to achieve collective observability. The key feature of our framework is that it utilizes local information not only during the execution of the proposed distributed input and state estimation architecture but also in its design in that global uniform ultimate boundedness of error dynamics is guaranteed once each node satisfies given local stability conditions independent from the graph topology and neighboring information of these nodes. As a special case (e.g., when all nodes are active and a positive real condition is satisfied), the asymptotic stability can be achieved with our algorithm. Several illustrative numerical examples are further provided to demonstrate the efficacy of the proposed architecture.

8.1 Introduction

As technological advances have boosted the development of integrated microsystems that combine sensing, computing, and communication on a single platform, we are rapidly moving toward a future in which large numbers of integrated microsensors will have the capability to operate in both civilian and military environments. Such large-scale sensor networks will support applications with dramatically increasing levels of complexity including situational awareness, environment monitoring, scientific data gathering, collaborative information processing, and search and rescue; to name but a few examples. One of

*This chapter is previously published in [1]. Permission is included in Appendix I.

the important areas of research in sensor networks is the development of distributed estimation algorithms for dynamic information fusion. Because, these algorithms are reliable to possible loss of a subset of nodes and communication links and they are flexible in the sense that nodes can be added and removed by making only local changes to the sensor network.

There are two common ways to do distributed dynamic information fusion. Specifically, one classical way include decentralized data fusion, for example, see [55–57], where these methods have been shown to work well in practice for many applications without formal stability guarantees. Unlike these methods, system-theoretic dynamic information fusion involve equations of motion to describe time behavior of the information fusion process and they also offer stability guarantees (e.g. [58–60]). The contribution of this paper builds on system-theoretic dynamic information fusion approaches.

Although distributed estimation algorithms have had strong appeal owing to their reliability and flexibility as outlined above, a critical roadblock to achieve correct dynamic information fusion with these algorithms is heterogeneity. Heterogeneity in sensor networks is unavoidable in real-world applications. To elucidate this fact, consider a target estimation problem as a motivating example. Specifically, nodes of a given sensor network can have heterogeneous information roles in the target estimation problem such that a subset of nodes can be subject to observations of this target (active nodes) and the rest can be subject to no observation (passive nodes). Thus, only active nodes have to be taken into account during the dynamic information fusion process. In addition, note that nodes of a sensor network can also have nonidentical sensor modalities; for example, a subset of nodes can sense the target position and others can sense the target velocity. This case also needs to be considered in the dynamic information fusion process.

Dealing with these classes of heterogeneity in sensor networks to achieve correct and reliable dynamic information fusion is a challenging task using distributed estimation algorithms. Toward this end, notable contributions in the literature include [2, 3, 58–75]. Specifically, the authors of [58–65] propose dynamic consensus algorithms that are suitable for sensor networks with all nodes being active. However, as discussed above, a subset of nodes in a sensor network can be passive in that they may not be able to sense a process of interest and collect information. While the authors of [66–68] present methods that cover specific applications when a subset of nodes are passive (and the remaining nodes are active), their results are in the context of static consensus, and hence, they are not suitable in their presented form for dynamic data-driven applications.

To address this challenge, the authors of [69–75] introduce the concept of sensor networks with active and passive nodes in the context of dynamic consensus. However, nodes of the considered class of sensor networks are implicitly assumed to have identical sensor modalities since each node is modeled using single integrator dynamics. Finally, the authors of [3] and [2] consider dynamic information fusion for sensor networks having nonidentical sensor modalities, where the former contribution requires each node to be active via sensing some states of a process of interest. While this is implicitly not assumed in the latter contribution, global information is required during the distributed algorithm design in terms of guaranteeing global asymptotic stability (although the proposed algorithm can be executed by solely relying on local information exchange between neighboring nodes).

The contribution of this paper is to introduce and analyze a new distributed input and state estimation architecture for heterogeneous sensor networks. Specifically, nodes of a given sensor network are allowed to have heterogeneous information roles in the sense that a subset of nodes can be active (that is, subject to observations of a process of interest) and the rest can be passive (that is, subject to no observation). Both fixed and varying active and passive roles of sensor nodes in the network are investigated. In addition, these nodes are allowed to have nonidentical sensor modalities under the common underlying assumption that they have complimentary properties distributed over the sensor network to achieve collective observability (see, for example, [3] and [2], and references therein). The key feature of our framework is that it utilizes local information not only during the execution of the proposed distributed input and state estimation architecture but also in its design unlike the results in [2]; that is, global uniform ultimate boundedness of error dynamics is guaranteed once each node satisfies given local stability conditions independent from the graph topology and neighboring information of these nodes. Several illustrative numerical examples are further provided to demonstrate the efficacy of the proposed architecture.

8.2 Notation and Mathematical Preliminaries

The notation used in this paper is fairly standard. Specifically, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ denotes the set of $n \times 1$ real column vectors, $\mathbb{R}^{n\times m}$ denotes the set of $n \times m$ real matrices, $\mathbb{S}_+^{n\times n}$ (resp., $\overline{\mathbb{S}}_+^{n\times n}$) denotes the set of $n \times n$ positive-definite (resp., positive-semidefinite) real matrices, 0_n denotes the $n \times 1$ vector of all zeros, $\mathbf{1}_n$ denotes the $n \times 1$ vector of all ones, and I_n denotes the $n \times n$ identity matrix. In addition, we write $(\cdot)^{\mathrm{T}}$ for transpose, $(\cdot)^{\dagger}$ for generalized inverse, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ for the minimum and maximum eigenvalue of the Hermitian matrix A, respectively, $\lambda_i(A)$ for the i-th eigenvalue of A, where

A is Hermitian and the eigenvalues are ordered from least to greatest value, $\mathrm{diag}(a)$ for the diagonal matrix with the vector a on its diagonal, $[x]_i$ for the entry of the vector x on the i-th row, and $[A]_{ij}$ for the entry of the of the matrix A on the i-th row and j-th column. Note that, throughout the paper, we use $A > 0$ (resp., $A \geq 0$) to indicate $A \in \mathbb{S}_+^{n\times n}$ (resp., $A \in \overline{\mathbb{S}}_+^{n\times n}$).

We now recall some basic notions from graph theory and refer to textbooks [5] and [92] for details. Specifically, graphs are broadly adopted in the sensor networks literature to encode interactions between nodes. An undirected graph $\mathcal{G}$ is defined by a set $\mathcal{V}_\mathcal{G} = \{1,\ldots,N\}$ of nodes and a set $\mathcal{E}_\mathcal{G} \subset \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ of edges. If $(i,j) \in \mathcal{E}_\mathcal{G}$, then the nodes i and j are neighbors and the neighboring relation is indicated with $i \sim j$. The degree of a node is given by the number of its neighbors. Letting d_i be the degree of node i, then the degree matrix of a graph $\mathcal{G}$, $\mathcal{D}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is given by

$$\mathcal{D}(\mathcal{G}) \triangleq \mathrm{diag}(d), \quad d = [d_1,\ldots,d_N]^{\mathrm{T}}. \tag{8.1}$$

A path $i_0 i_1 \ldots i_L$ is a finite sequence of nodes such that $i_{k-1} \sim i_k$, $k = 1,\ldots,L$, and a graph $\mathcal{G}$ is connected if there is a path between any pair of distinct nodes. The adjacency matrix of a graph $\mathcal{G}$, $\mathcal{A}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is given by

$$[\mathcal{A}(\mathcal{G})]_{ij} \triangleq \begin{cases} 1, & \text{if } (i,j) \in \mathcal{E}_\mathcal{G}, \\ 0, & \text{otherwise.} \end{cases} \tag{8.2}$$

The Laplacian matrix of a graph, $\mathcal{L}(\mathcal{G}) \in \overline{\mathbb{S}}_+^{N\times N}$, playing a central role in many graph-theoretic treatments of sensor networks, is given by

$$\mathcal{L}(\mathcal{G}) \triangleq \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G}). \tag{8.3}$$

The spectrum of the Laplacian of an undirected and connected graph can be ordered as

$$0 = \lambda_1(\mathcal{L}(\mathcal{G})) < \lambda_2(\mathcal{L}(\mathcal{G})) \leq \cdots \leq \lambda_N(\mathcal{L}(\mathcal{G})), \tag{8.4}$$

with $\mathbf{1}_N$ as the eigenvector corresponding to the zero eigenvalue $\lambda_1(\mathcal{L}(\mathcal{G}))$ and $\mathcal{L}(\mathcal{G})\mathbf{1}_N = 0_N$ and $\mathrm{e}^{\mathcal{L}(\mathcal{G})}\mathbf{1}_N = \mathbf{1}_N$. Throughout this paper, we assume that the graph $\mathcal{G}$ of a given sensor network is undirected and connected.

To develop the main results of this paper, the following lemmas are necessary.

Lemma 8.2.1 (**Proposition 8.1.2, [117].**). *Let $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$. If $A \geq 0$ and $B > 0$, then $A + B > 0$.*

Lemma 8.2.2 ((**Proposition 8.2.4, [117]**).). *Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times m}$, and*

$$X = \begin{bmatrix} A & B \\ B^{\mathrm{T}} & C \end{bmatrix}.$$

Then, the following statements are equivalent:

i) $X \geq 0$.

ii) $A \geq 0,\ C - B^{\mathrm{T}}A^{\dagger}B \geq 0,\ (I - AA^{\dagger})B = 0.$

iii) $C \geq 0,\ A - BC^{\dagger}B^{\mathrm{T}} \geq 0,\ (I - CC^{\dagger})B^{\mathrm{T}} = 0.$

Finally, **Co**Ω is defined as a polytope or a bounded polyhedron, which is the intersection of a finite number of half space and hyperplanes ([135]). For the following lemma, let $P \in \mathbb{R}^{n\times n}$, $A(t) \in \mathbb{R}^{n\times n}$, **Co**$\Omega \triangleq$ **Co**$\{A_1, \ldots, A_L\}$, and $A(t) \in$ **Co**$\{A_1, \ldots, A_L\}$ where **Co** denotes the convex hull and $A_i \in \mathbb{R}^{n\times n}$ are the vertices of the polytope.

Lemma 8.2.3 (**[136]**).). *If $P > 0$, $A_i^{\mathrm{T}}P + PA_i \leq 0$ holds, then $A^{\mathrm{T}}(t)P + PA(t) \leq 0$ holds.*

By letting $P = I_n$, it follows from Lemma 8.2.3 that $A^{\mathrm{T}}(t) + A(t) \leq 0$ holds, when $A_i^{\mathrm{T}} + A_i \leq 0$ holds. If, in addition, $A(t)$ is symmetric, then it further follows that $A(t) \leq 0$ holds, if $A_i \leq 0$

8.3 Distributed Input and State Estimation for Active-Passive Sensor Networks with Fixed Node Roles

In this section, we introduce and analyze a distributed input and state estimation architecture for heterogeneous sensor networks, where the active and passive role of each node is fixed. For this purpose, consider a process of interest with the (open-loop or closed-loop) dynamics given by

$$\dot{x}(t) = Ax(t) + Bw(t), \quad x(0) = x_0, \tag{8.5}$$

where $x(t) \in \mathbb{R}^n$ denotes an unmeasurable process state vector, $w(t) \in \mathbb{R}^p$ denotes an unknown bounded input (e.g., command) to this process with a bounded time rate of change, $A \in \mathbb{R}^{n\times n}$ denotes the Hurwitz system matrix necessary for internal process stability, and $B \in \mathbb{R}^{n\times p}$ denotes the system input matrix.

Next, consider a sensor network with N nodes exchanging information among each other using their local measurements according to an undirected and connected graph $\mathcal{G}$. In the sense of [69–75], if a node i, $i = 1,\ldots,N$, is subject to observations of the process (8.5) given by

$$y_i(t) = C_i x(t), \tag{8.6}$$

where $y_i(t) \in \mathbb{R}^m$ and $C_i \in \mathbb{R}^{m\times n}$ denote the measurable process output and the system output matrix for node i, $i = 1,\ldots,N$, respectively, then we say that it is an active node. Similarly, if a node i, $i = 1,\ldots,N$, has no observations, then we say that it is a passive node. Notice that the above formulation allows for nonidentical sensor modalities since C_i of active nodes can be different. Here, as standard in the literature, we assume that each active node has complimentary properties distributed over the sensor network to guarantee collective observability (see, for example, [3] and [2], and references therein), although the pairs (A,C_i), $i = 1,\ldots,N$, may not be locally observable. In mathematical sense, collective observability condition means the pair (A,C) is observable, where $C = [C_1^{\mathrm{T}}, C_2^{\mathrm{T}}, \ldots, C_N^{\mathrm{T}}]^{\mathrm{T}}$ (e.g., see [2]).

Here, we are interested in the problem of distributively estimating the unmeasurable state $x(t)$ and the unknown input $w(t)$ of the process given by (8.5) using a sensor network, where active nodes are subject to the observations given by (8.6). For this purpose, the rest of this section is divided into two parts, where we first introduce the proposed distributed estimation architecture and then analyze its stability in detail using tools and methods from system theory.

8.3.1 Proposed Distributed Estimation Architecture

For node i, $i = 1,\ldots,N$, consider the distributed estimation algorithm given by

$$\dot{\hat{x}}_i(t) = (A-\gamma P_i^{-1})\hat{x}_i(t) + B\hat{w}_i(t) + g_i L_i\big(y_i(t) - C_i\hat{x}_i(t)\big) - \alpha P_i^{-1}\sum_{i\sim j}(\hat{x}_i(t)-\hat{x}_j(t)),$$
$$\hat{x}_i(0) = \hat{x}_{i0}, \tag{8.7}$$

$$\dot{\hat{w}}_i(t) = g_i J_i(y_i(t) - C_i\hat{x}_i(t)) - (\sigma_i K_i + \gamma \mathrm{I}_p)\hat{w}_i(t) - \alpha\sum_{i\sim j}(\hat{w}_i(t)-\hat{w}_j(t)), \quad \hat{w}_i(0) = \hat{w}_{i0}, \tag{8.8}$$

where $\hat{x}_i(t) \in \mathbb{R}^n$ is a local state estimate of $x(t)$ for node i, $\hat{w}_i \in \mathbb{R}^p$ is a local input estimate of $w(t)$ for node i, $L_i \in \mathbb{R}^{n\times m}$, $J_i \in \mathbb{R}^{p\times m}$ and $K_i \in \mathbb{S}_+^{p\times p}$ are design matrices of node i, and α, γ, and $\sigma_i \in \mathbb{R}$ are positive design coefficients for node i. Here, $g_i = 1$ for active nodes and otherwise $g_i = 0$. In addition, $P_i > 0$ is the

consensus gain satisfying the linear matrix inequality given by

$$R_i = \begin{bmatrix} \bar{A}_i^{\mathrm{T}} P_i + P_i \bar{A}_i & -P_i B + g_i C_i^{\mathrm{T}} J_i^{\mathrm{T}} \\ -B^{\mathrm{T}} P_i + g_i J_i C_i & -2\sigma_i K_i \end{bmatrix} \leq 0, \tag{8.9}$$

where

$$\bar{A}_i \triangleq A - g_i L_i C_i. \tag{8.10}$$

Remark 8.3.1. *The local condition given by (8.9) for node i, $i = 1, \ldots, N$, plays a central role in the stability analysis presented in the next section. Specifically, if the proposed input and state estimation architecture given by (8.7) and (8.8) satisfies the local condition given by (8.9) for each node, then the global uniform ultimate boundedness of error dynamics is guaranteed for the overall sensor network. In addition, note that the local condition given by (8.9) is well-posed. To see this, for example, let P_i satisfy the linear matrix inequality given by $\bar{A}_i^{\mathrm{T}} P_i + P_i \bar{A}_i < 0$, $i = 1, \ldots, N$. Then, it can be readily shown that there can exist a sufficiently large σ_i, $i = 1, \ldots, N$, such that (8.9) holds. As a special case, if all nodes are active and a well-known positive real condition $P_i B = C_i^{\mathrm{T}} J_i^{\mathrm{T}}$ holds (see, for example, [137–141], and references therein), then it can be easily seen that (8.9) holds even for small values of σ_i, $i = 1, \ldots, N$. From this standpoint, it should be also mentioned that (8.9) relaxes this condition $P_i B = C_i^{\mathrm{T}} J_i^{\mathrm{T}}$ similar in spirit to how the authors of [142–146] and [147] relax similar conditions. Finally, once again, for the special case when all nodes are active, if $\mathcal{H}(s) \triangleq J_i C_i (s\mathrm{I} - \bar{A}_i)^{-1} B + \sigma_i K_i$ is passive, $i = 1, \ldots, N$, then (8.9) is feasible and vice versa [136].*

8.3.2 Stability Analysis

Let $\tilde{x}_i(t) \triangleq x(t) - \hat{x}_i(t)$ and $\tilde{w}_i(t) \triangleq \hat{w}_i(t) - w(t)$. Then, based on (8.7) and (8.8),

$$\begin{aligned} \dot{\tilde{x}}_i(t) &= Ax(t) + Bw(t) - (A - \gamma P_i^{-1})\hat{x}_i(t) - B\hat{w}_i(t) - g_i L_i (y_i(t) - C_i \hat{x}_i(t)) + \alpha P_i^{-1} \sum_{i \sim j} (\hat{x}_i(t) - \hat{x}_j(t)) \\ &= (A - g_i L_i C_i)\tilde{x}_i(t) - B\tilde{w}_i(t) + \alpha P_i^{-1} \sum_{i \sim j} (\hat{x}_i(t) - \hat{x}_j(t)) + \gamma P_i^{-1} \hat{x}_i(t) \\ &= \bar{A}_i \tilde{x}_i(t) - B\tilde{w}_i(t) - \alpha P_i^{-1} \sum_{i \sim j} (\tilde{x}_i(t) - \tilde{x}_j(t)) - \gamma P_i^{-1} \big(\tilde{x}_i(t) - x(t)\big), \quad \tilde{x}_i(0) = \tilde{x}_{i0}, \end{aligned} \tag{8.11}$$

$$\begin{aligned} \dot{\tilde{w}}_i(t) &= g_i J_i C_i \tilde{x}_i(t) - \sigma_i K_i (\tilde{w}_i(t) + w(t)) - \alpha \sum_{i \sim j} (\tilde{w}_i(t) - \tilde{w}_j(t)) - \gamma (\tilde{w}_i(t) + w(t)) - \dot{w}(t), \\ &\qquad \tilde{w}_i(0) = \tilde{w}_{i0}. \end{aligned} \tag{8.12}$$

Now, considering the aggregated vectors given by

$$\tilde{x}(t) \triangleq [\tilde{x}_1^{\mathrm{T}}(t), \tilde{x}_2^{\mathrm{T}}(t), \ldots, \tilde{x}_N^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{Nn}, \tag{8.13}$$

$$\tilde{w}(t) \triangleq [\tilde{w}_1^{\mathrm{T}}(t), \tilde{w}_2^{\mathrm{T}}(t), \ldots, \tilde{w}_N^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{Np}, \tag{8.14}$$

we can write the error dynamics as

$$\begin{aligned}\dot{\tilde{x}}(t) = &\begin{bmatrix} \bar{A}_1 & 0 & \cdots & 0 \\ 0 & \bar{A}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{A}_N \end{bmatrix} \tilde{x}(t) - \begin{bmatrix} B & 0 & \cdots & 0 \\ 0 & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B \end{bmatrix} \tilde{w}(t) \\ &-\alpha \begin{bmatrix} \mathcal{L}_{11}P_1^{-1} & \mathcal{L}_{12}P_1^{-1} & \cdots & \mathcal{L}_{1N}P_1^{-1} \\ \mathcal{L}_{21}P_2^{-1} & \mathcal{L}_{22}P_2^{-1} & \cdots & \mathcal{L}_{2N}P_2^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}_{N1}P_N^{-1} & \mathcal{L}_{N2}P_N^{-1} & \cdots & \mathcal{L}_{NN}P_N^{-1} \end{bmatrix} \tilde{x}(t) - \gamma \begin{bmatrix} P_1^{-1} & 0 & \cdots & 0 \\ 0 & P_2^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_N^{-1} \end{bmatrix} \tilde{x}(t) + \gamma \begin{bmatrix} P_1^{-1} \\ P_2^{-1} \\ \vdots \\ P_N^{-1} \end{bmatrix} x(t), \end{aligned} \tag{8.15}$$

$$\begin{aligned}\dot{\tilde{w}}(t) = &\begin{bmatrix} g_1J_1C_1 & 0 & \cdots & 0 \\ 0 & g_2J_2C_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_NJ_NC_N \end{bmatrix} \tilde{x}(t) - \begin{bmatrix} \sigma_1K_1 & 0 & \cdots & 0 \\ 0 & \sigma_2K_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_NK_N \end{bmatrix} \tilde{w}(t) - \begin{bmatrix} \sigma_1K_1 \\ \sigma_2K_2 \\ \vdots \\ \sigma_NK_N \end{bmatrix} w(t) \\ &-\alpha \begin{bmatrix} \mathcal{L}_{11}I_p & \mathcal{L}_{12}I_p & \cdots & \mathcal{L}_{1N}I_p \\ \mathcal{L}_{21}I_p & \mathcal{L}_{22}I_p & \cdots & \mathcal{L}_{2N}I_p \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}_{N1}I_p & \mathcal{L}_{N2}I_p & \cdots & \mathcal{L}_{NN}I_p \end{bmatrix} \tilde{w}(t) - \gamma \begin{bmatrix} I_p & 0 & \cdots & 0 \\ 0 & I_p & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_p \end{bmatrix} \tilde{w}(t) + \begin{bmatrix} -\gamma w(t) - \dot{w}(t) \\ -\gamma w(t) - \dot{w}(t) \\ \vdots \\ -\gamma w(t) - \dot{w}(t) \end{bmatrix}, \end{aligned} \tag{8.16}$$

where $\mathcal{L}_{ij} \in \mathbb{R}$ is the entry in the i-th row and j-th column of the Laplacian matrix.

The error dynamics now can be written a compact form as

$$\dot{\tilde{x}}(t) = \bar{A}\tilde{x}(t) - (\mathrm{I}_N \otimes B)\tilde{w}(t) - P^{-1}(F \otimes \mathrm{I}_n)\tilde{x}(t) + \gamma P^{-1}(\mathbf{1}_N \otimes \mathrm{I}_n)x(t), \tag{8.17}$$

$$\dot{\tilde{w}}(t) = M\tilde{x}(t) - \bar{K}\big(\tilde{w}(t) + (\mathbf{1}_N \otimes \mathrm{I}_p)w(t)\big) - (F \otimes \mathrm{I}_p)\tilde{w}(t) - \gamma(\mathbf{1}_N \otimes \mathrm{I}_p)w(t) - (\mathbf{1}_N \otimes \mathrm{I}_p)\dot{w}(t), \quad (8.18)$$

where

$$\bar{A} \triangleq \mathrm{diag}([\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_N]), \quad (8.19)$$

$$M \triangleq \mathrm{diag}([g_1 J_1 C_1, g_2 J_2 C_2, \ldots, g_N J_N C_N]), \quad (8.20)$$

$$\bar{K} \triangleq \mathrm{diag}([\sigma_1 K_1, \sigma_2 K_2, \ldots, \sigma_N K_N]), \quad (8.21)$$

$$F \triangleq \alpha\mathcal{L}(\mathcal{G}) + \gamma \mathrm{I}_N, \quad (8.22)$$

$$P \triangleq \mathrm{diag}([P_1, P_2, \ldots, P_N]), \quad (8.23)$$

with $\mathcal{L}(\mathcal{G})$ being the Laplacian matrix. Note that $P > 0$ readily follows from $P_i > 0$.

Theorem 8.3.1. *Consider the process given by (8.5) and the distributed input and state estimation architecture given by (8.7) and (8.8). Assume (8.9) holds and nodes exchange information using local measurements subject to an undirected and connected graph $\mathcal{G}$. Then, the error dynamics given by (8.17) and (8.18) are uniformly ultimately bounded.*

Proof. Consider the Lyapunov function candidate given by

$$V(\tilde{x}, \tilde{w}) = \tilde{x}^{\mathrm{T}} P \tilde{x} + \tilde{w}^{\mathrm{T}} \tilde{w}. \quad (8.24)$$

Note that $V(0,0) = 0$ and $V(\tilde{x}, \tilde{w}) > 0$ for all $(\tilde{x}, \tilde{w}) \neq (0,0)$. Taking time-derivative of $V(\tilde{x}, \tilde{w})$ along the trajectories of (8.17) and (8.18) yields

$$\begin{aligned}
\dot{V}(\cdot) &= \tilde{x}^{\mathrm{T}}(t)(\bar{A}^{\mathrm{T}}P + P\bar{A})\tilde{x}(t) - 2\tilde{x}^{\mathrm{T}}(t)P(\mathrm{I}_N \otimes B)\tilde{w}(t) - 2\tilde{x}^{\mathrm{T}}(t)(F \otimes \mathrm{I}_n)\tilde{x}(t) \\
&\quad + 2\gamma\tilde{x}^{\mathrm{T}}(t)(\mathbf{1}_N \otimes \mathrm{I}_n)x(t) + 2\tilde{w}^{\mathrm{T}}(t)M\tilde{x}(t) - 2\tilde{w}^{\mathrm{T}}(t)\bar{K}\tilde{w}(t) - 2\tilde{w}^{\mathrm{T}}(t)(F \otimes \mathrm{I}_p)\tilde{w}(t) \\
&\quad - 2\tilde{w}^{\mathrm{T}}(t)(\bar{K} + \gamma \mathrm{I}_{Np})(\mathbf{1}_N \otimes \mathrm{I}_p)w(t) - 2\tilde{w}^{\mathrm{T}}(t)(\mathbf{1}_N \otimes \mathrm{I}_p)\dot{w}(t) \\
&= \tilde{x}^{\mathrm{T}}(t)(\bar{A}^{\mathrm{T}}P + P\bar{A})\tilde{x}(t) - 2\tilde{w}^{\mathrm{T}}(t)\bar{K}\tilde{w}(t) - 2\tilde{x}^{\mathrm{T}}(t)\big(P(\mathrm{I}_N \otimes B) - M^{\mathrm{T}}\big)\tilde{w}(t) \\
&\quad - 2\tilde{x}^{\mathrm{T}}(t)(F \otimes \mathrm{I}_n)\tilde{x}(t) - 2\tilde{w}^{\mathrm{T}}(t)(F \otimes \mathrm{I}_p)\tilde{w}(t) + 2\gamma\tilde{x}^{\mathrm{T}}(t)(\mathbf{1}_N \otimes \mathrm{I}_n)x(t) \\
&\quad - 2\tilde{w}^{\mathrm{T}}(t)(\bar{K} + \gamma \mathrm{I}_{Np})(\mathbf{1}_N \otimes \mathrm{I}_p)w(t) - 2\tilde{w}^{\mathrm{T}}(t)(\mathbf{1}_N \otimes \mathrm{I}_p)\dot{w}(t)
\end{aligned}$$

$$
\begin{aligned}
&= \begin{bmatrix} \tilde{x}^{\mathrm{T}}(t) & \tilde{w}^{\mathrm{T}}(t) \end{bmatrix} \begin{bmatrix} \bar{A}^{\mathrm{T}}P + P\bar{A} & -P(\mathrm{I}_N \otimes B) + M^{\mathrm{T}} \\ -(\mathrm{I}_N \otimes B^{\mathrm{T}})P + M & -2\bar{K} \end{bmatrix} \begin{bmatrix} \tilde{x}(t) \\ \tilde{w}(t) \end{bmatrix} \\
&\quad + \begin{bmatrix} \tilde{x}^{\mathrm{T}}(t) & \tilde{w}^{\mathrm{T}}(t) \end{bmatrix} \begin{bmatrix} -2(F \otimes \mathrm{I}_n) & 0 \\ 0 & -2(F \otimes \mathrm{I}_p) \end{bmatrix} \begin{bmatrix} \tilde{x}(t) \\ \tilde{w}(t) \end{bmatrix} \\
&\quad + 2 \begin{bmatrix} \tilde{x}^{\mathrm{T}}(t) & \tilde{w}^{\mathrm{T}}(t) \end{bmatrix} \begin{bmatrix} \gamma(\mathbf{1}_N \otimes \mathrm{I}_n)x(t) \\ -(\bar{K} + \gamma \mathrm{I}_{Np})(\mathbf{1}_N \otimes \mathrm{I}_p)w(t) - (\mathbf{1}_N \otimes \mathrm{I}_p)\dot{w}(t) \end{bmatrix} \\
&= z^{\mathrm{T}}(t)R_A z(t) + z^{\mathrm{T}}(t)R_B z(t) + 2z^{\mathrm{T}}(t)\phi \\
&= z^{\mathrm{T}}(t)Rz(t) + 2z^{\mathrm{T}}(t)\phi, \qquad (8.25)
\end{aligned}
$$

where

$$
\begin{aligned}
z(t) &\triangleq [\tilde{x}^{\mathrm{T}}(t),\ \tilde{w}^{\mathrm{T}}(t)]^{\mathrm{T}}, \qquad (8.26) \\
R_A &\triangleq \begin{bmatrix} \bar{A}^{\mathrm{T}}P + P\bar{A} & -P(\mathrm{I}_N \otimes B) + M^{\mathrm{T}} \\ -(\mathrm{I}_N \otimes B^{\mathrm{T}})P + M & -2\bar{K} \end{bmatrix}, \qquad (8.27) \\
R_B &\triangleq \begin{bmatrix} -2(F \otimes \mathrm{I}_n) & 0 \\ 0 & -2(F \otimes \mathrm{I}_p) \end{bmatrix}, \qquad (8.28) \\
R &\triangleq R_A + R_B \\
&= \begin{bmatrix} \bar{A}^{\mathrm{T}}P + P\bar{A} - 2(F \otimes \mathrm{I}_n) & -P(\mathrm{I}_N \otimes B) + M^{\mathrm{T}} \\ -(\mathrm{I}_N \otimes B^{\mathrm{T}})P + M & -2\bar{K} - 2(F \otimes \mathrm{I}_p) \end{bmatrix}, \qquad (8.29) \\
\phi &\triangleq \begin{bmatrix} \gamma(\mathbf{1}_N \otimes \mathrm{I}_n)x(t) \\ -(\bar{K} + \gamma \mathrm{I}_{Np})(\mathbf{1}_N \otimes \mathrm{I}_p)w(t) - (\mathbf{1}_N \otimes \mathrm{I}_p)\dot{w}(t) \end{bmatrix}. \qquad (8.30)
\end{aligned}
$$

Note that $(F \otimes \mathrm{I}_n) > 0$ and $(F \otimes \mathrm{I}_p) > 0$ readily follow from $F > 0$, and hence, $R_B < 0$.

Next, since the linear matrix inequality given by (8.9) holds, it follows that

$$
\begin{aligned}
\bar{A}_i^{\mathrm{T}}P_i + P_i\bar{A}_i &\leq 0, \qquad (8.31) \\
N_i \triangleq -2\sigma_i K_i - (-B^{\mathrm{T}}P_i + g_i J_i C_i)(\bar{A}_i^{\mathrm{T}}P_i + P_i\bar{A}_i)^{\dagger}(-P_i B + g_i C_i^{\mathrm{T}}J_i^{\mathrm{T}}) &\leq 0, \qquad (8.32) \\
Q_i \triangleq \left(\mathrm{I}_n - (\bar{A}_i^{\mathrm{T}}P_i + P_i\bar{A}_i)(\bar{A}_i^{\mathrm{T}}P_i + P_i\bar{A}_i)^{\dagger}\right)(-P_i B + g_i C_i^{\mathrm{T}}J_i^{\mathrm{T}}) &= 0, \qquad (8.33)
\end{aligned}
$$

by applying Lemma 8.2.2 to (8.9). Note that

$$\bar{A}^{\mathrm{T}}P + P\bar{A} = \begin{bmatrix} \hat{A}_1 & 0 & \cdots & 0 \\ 0 & \hat{A}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{A}_N \end{bmatrix} \leq 0, \tag{8.34}$$

as a consequence of (8.31), where $\hat{A}_i \triangleq \bar{A}_i^{\mathrm{T}}P_i + P_i\bar{A}_i$ for $i = 1,\ldots,N$. Furthermore, it follows from (8.32) that

$$\begin{aligned} \mathcal{N} &\triangleq -2\bar{K} - \big(-(\mathrm{I}_N \otimes B^{\mathrm{T}})P + M\big)(\bar{A}^{\mathrm{T}}P + P\bar{A})^{\dagger}(-P(\mathrm{I}_N \otimes B) + M^{\mathrm{T}}) \\ &= \begin{bmatrix} N_1 & 0 & \cdots & 0 \\ 0 & N_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N_N \end{bmatrix} \leq 0, \end{aligned} \tag{8.35}$$

holds. Finally, (8.33) leads to

$$\begin{aligned} Q &\triangleq \big(\mathrm{I}_{Nn} - (\bar{A}^{\mathrm{T}}P + P\bar{A})(\bar{A}^{\mathrm{T}}P + P\bar{A})^{\dagger}\big)(-P(\mathrm{I}_N \otimes B) + M^{\mathrm{T}}) \\ &= \begin{bmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & Q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_N \end{bmatrix} = 0. \end{aligned} \tag{8.36}$$

Now, by Lemma 8.2.2, $R_A \leq 0$ as a direct consequence of (8.34), (8.35) and (8.36). Thus, by Lemma 8.2.1, $R = R_A + R_B < 0$.

Note that since A is Hurwitz, and $\|w(t)\|_2 \leq \bar{w}$, we have $\|x(t)\|_2 \leq \bar{x}$, where $\bar{w}$ and $\bar{x}$ are upper bounds of the input and the state, respectively. With this and $\|\dot{w}(t)\|_2 \leq \bar{\dot{w}}$, where $\bar{\dot{w}}$ is the upper bound of the time rate of change of input, we have $\|\phi\|_2 \leq \bar{\phi}$ with

$$\begin{aligned} \bar{\phi} &\triangleq \sqrt{\gamma^2\|(\mathbf{1}_N \otimes \mathrm{I}_n)\|_2^2\bar{x}^2 + \|\bar{K} + \gamma\mathrm{I}_{Np}\|_2^2\|\mathbf{1}_N \otimes \mathrm{I}_p\|_2^2\bar{w}^2 + \|\mathbf{1}_N \otimes \mathrm{I}_p\|_2^2\bar{\dot{w}}^2} \\ &= \sqrt{N\gamma^2\bar{x}^2 + \|\bar{K} + \gamma\mathrm{I}_{Np}\|_2^2 N\bar{w}^2 + N\bar{\dot{w}}^2}. \end{aligned} \tag{8.37}$$

Now, one can write

$$\begin{aligned}\dot{V}(\cdot) &= z^{\mathrm{T}}(t)Rz(t)+2z^{\mathrm{T}}(t)\phi \\ &\leq \lambda_{\max}(R)\|z(t)\|_2^2+2\|z(t)\|_2\bar{\phi} \\ &\leq (1-\theta)\lambda_{\max}(R)\|z(t)\|_2^2+\theta\lambda_{\max}(R)\|z(t)\|_2^2+2\|z(t)\|_2\bar{\phi}, \end{aligned} \tag{8.38}$$

with $\lambda_{\max}(R)<0$ and $\theta\in(0,1)$. Letting $\mu_1 \triangleq \frac{-2\bar{\phi}}{\theta\lambda_{\max}(R)}>0$ and $\Omega_1 \triangleq \{z(t):\|z(t)\|_2\leq\mu_1\}$, it follows that $\dot{V}(\cdot)\leq(1-\theta)\lambda_{\max}(R)\|z(t)\|_2^2<0$ outside the compact set Ω_1, and hence, the error dynamics given by (8.17) and (8.18) are uniformly ultimately bounded by Theorem 4.18 of [119]. ■

The following corollary to the above theorem is now immediate.

Corollary 8.3.1. *Consider the process given by (8.5) and the distributed input and state estimation architecture given by (8.7) and (8.8). Assume (8.9) holds and nodes exchange information using local measurements subject to an undirected and connected graph $\mathcal{G}$. Then, for all $z(0)\in\mathbb{R}^{N(n+p)}$, there exists $T=T\left(z(0),\mu_1\right)\geq 0$ such that*

$$\|\tilde{x}(t)\|_2 \leq \xi_1 \triangleq \sqrt{\frac{\lambda_{\max}(\bar{P})}{\lambda_{\min}(\bar{P})}}\max\left\{\|z(0)\|_2\, e^{((1-\theta)\lambda_{\max}(R)/2\lambda_{\max}(\bar{P}))t},\mu_1\right\},\quad \forall t\geq 0, \tag{8.39}$$

$$\|\tilde{w}(t)\|_2 \leq \xi_1,\quad \forall t\geq 0, \tag{8.40}$$

where

$$\bar{P}=\begin{bmatrix}P & 0\\ 0 & \mathrm{I}_{Np}\end{bmatrix}, \tag{8.41}$$

and

$$\|\tilde{x}(t)\|_2 \leq \psi_1 \triangleq \sqrt{\frac{\lambda_{\max}(\bar{P})}{\lambda_{\min}(P)}}\mu_1,\quad t\geq T, \tag{8.42}$$

$$\|\tilde{w}(t)\|_2 \leq \zeta_1 \triangleq \sqrt{\lambda_{\max}(\bar{P})}\mu_1,\quad t\geq T. \tag{8.43}$$

Proof. Note that

$$\begin{aligned}V(\cdot) &= \tilde{x}^{\mathrm{T}}(t)P\tilde{x}(t)+\tilde{w}^{\mathrm{T}}(t)\tilde{w}(t) \\ &= \begin{bmatrix}\tilde{x}^{\mathrm{T}}(t) & \tilde{w}^{\mathrm{T}}(t)\end{bmatrix}\begin{bmatrix}P & 0\\ 0 & \mathrm{I}_{Np}\end{bmatrix}\begin{bmatrix}\tilde{x}(t)\\ \tilde{w}(t)\end{bmatrix} \\ &= z^{\mathrm{T}}(t)\bar{P}z(t). \end{aligned} \tag{8.44}$$

Let $c_1 \triangleq \lambda_{\min}(\bar{P})$, $c_2 \triangleq \lambda_{\max}(\bar{P})$ and $c_3 \triangleq -(1-\theta)\lambda_{\max}(R)$. From (8.44), we have

$$c_1\|z(t)\|_2^2 \leq V(\cdot) \leq c_2\|z(t)\|_2^2. \tag{8.45}$$

In addition, $\dot{V}(\cdot) \leq -c_3\|z(t)\|_2^2$ for all $\|z(t)\|_2 \geq \mu_1$. By Theorem 4.5 of [148], since the domain $D = \mathbb{R}^{N(n+p)}$, for every initial state $z(0)$, the bound of the overall system is

$$\|z(t)\|_2 \leq \sqrt{\frac{c_2}{c_1}} \max\left\{\|z(0)\|_2\, e^{(-c_3/2c_2)t}, \mu_1\right\} = \xi_1, \quad \forall t \geq 0. \tag{8.46}$$

Using the fact that $\|\tilde{x}(t)\|_2 \leq \|z(t)\|_2$ and $\|\tilde{w}(t)\|_2 \leq \|z(t)\|_2$, (8.39) and (8.40) follow immediate.

In the proof of Theorem 8.3.1, we show that $V(\cdot)$ cannot grow outside the compact set Ω_1, thus (8.42) follows from $\lambda_{\min}(P)\|\tilde{x}(t)\|_2^2 \leq V\left(\tilde{x}(t), \tilde{w}(t)\right) \leq \lambda_{\max}(\bar{P})\|z(t)\|_2^2 \leq \lambda_{\max}(\bar{P})\mu_1^2$. Identically, (8.43) follows from $\|\tilde{w}(t)\|_2^2 \leq V\left(\tilde{x}(t), \tilde{w}(t)\right) \leq \lambda_{\max}(\bar{P})\|z(t)\|_2^2 \leq \lambda_{\max}(\bar{P})\mu_1^2$. The proof is now complete. ■

Remark 8.3.2. *While this paper shows the uniform ultimate boundedness of the error dynamics, the provided results (8.42) and (8.43) can be used to tune the design parameters to achieve acceptable performance criteria. The uniform ultimate boundedness can be considered as a result of the considered complex problem that we address here, which we can recap the main points as:*

i) The proposed algorithm only utilizes local information for designing agent-wise dynamics to achieve the stability, unlike existing results in [2].

ii) With regard to the considered problem in this paper, for the first time, we allow a subset of nodes to be passive (that is, subject to no observation).

iii) The sensing capability of active nodes can be different among sensors.

iv) Not only the states of the process are unknown but also the inputs are unknown.

v) We do not assume a common positive real condition, e.g., $P_iB = C_i^{\mathrm{T}} J_i^{\mathrm{T}}$, which in practice may not be easy to satisfy.

vi) The inputs are not constant.

If we relax some of these conditions, the asymptotic stability can be obtained with a version of the proposed algorithm (see Appendix A).

To summarize, the nature of the distributed estimation problem subjected to *i)-vi)* is challenging. In order to solve this problem using only local information and without the positive real condition, the condition (8.9) is required by the nature of the problem. In addition, we need the assumption that A is Hurwitz to make (8.9) feasible especially for passive nodes. Furthermore, when the input $w(t)$ is time-varying, adding leakage terms is unavoidable to prove the stability (e.g, see [149]).

Remark 8.3.3. *Since the ultimate bounds given by (8.42) and (8.43) depend on the design parameters of the proposed distributed input and state estimation architecture, they can be used as design metrics such that the design parameters can be judiciously selected to make (8.42) and (8.43) small. However, unlike the stability of our framework that is guaranteed once each node satisfies the local condition given by (8.9), such a performance characterization requires global information. However, one can further analyze the effect of each specific design parameter to these ultimate bounds and make conclusions without possibly requiring global information, which will be considered as a future research direction.*

The following remarks discuss how to choose our design parameters while Appendix C summarizes their effect for interested readers.

Remark 8.3.4. *Note that the terms "$-\gamma P_i^{-1}\hat{x}_i(t)$" and "$-(\sigma_i K_i+\gamma \mathrm{I}_p)\hat{w}_i(t)$" appearing respectively in (8.7) and (8.8) are often referred as leakage terms. If the gains "γP_i^{-1}" and "$\sigma_i K_i+\gamma \mathrm{I}_p$" respectively multiplying these terms are not small, then they may result in poor overall system performance (see, for example, [150, 151] and references therein), and hence, it is of common practice to choose these multiplier gains γ and σ_i to be small. However, as noted in Remark 8.3.1, σ_i may not be chosen as small unless all nodes are active and the condition $P_iB = C_i^{\mathrm{T}} J_i^{\mathrm{T}}$ holds. Therefore, we cast (8.9) as an optimization problem given by*

$$\text{minimize } \sigma_i, \tag{8.47}$$

$$\text{subject to } (8.9), \tag{8.48}$$

for all nodes $i = 1,\ldots,N$. In addition, it should be noted that since the matrix $\bar{K}$ appear in the numerator of the ultimate bound, σ_i and K_i should be chosen such that the norm $\|\sigma_i K_i\|_2$ is small.

Remark 8.3.5. *To elucidate the effect of design parameters to the ultimate bound given by (8.42), we consider, for example, a system with 4 sensors (1 and 3 are active nodes, and 2 and 4 are passive nodes)*

tracking a target with dynamics

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -1 & -0.25 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t), \tag{8.49}$$

where $w(t) = sin(0.25t)$. *Node 1 is subject to* $C_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$, *and node 3 is subject to* $C_3 = \begin{bmatrix} 0 & 1 \end{bmatrix}$. *We design* σ_i *by solving the linear matrix inequality (8.9). As a result, with* $J_i = K_i$ *and* $K_1 = K_2 = K_3 = K_4 = 50$, *we have* $\sigma_1 = 0.03$, $\sigma_2 = \sigma_4 = 0.05$, *and* $\sigma_3 = 0.03$ *with* $P_1 = \begin{bmatrix} 26.31 & -2.67 \\ -2.67 & 4.01 \end{bmatrix}$, $P_2 = P_4 = \begin{bmatrix} 1.54 & 0.08 \\ 0.08 & 1.62 \end{bmatrix}$, *and* $P_3 = \begin{bmatrix} 4.60 & 4.02 \\ 4.02 & 25.62 \end{bmatrix}$. *We then vary* α *and* γ *to see the effect of these parameters to the ultimate bound* ψ_1 *given by (8.42). Figure 8.1 shows the effect of the variation in* α *and* γ *to (8.42). From the figure, we can see that one can pick a small value for* γ *and a large value for* α *to reduce the ultimate bound.*

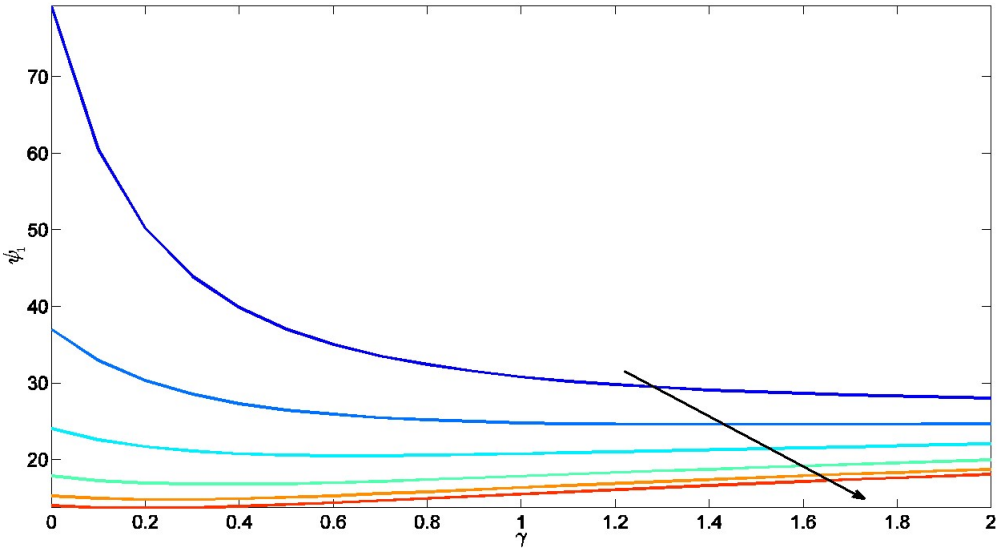


Figure 8.1: Effect of $\gamma \in (0,2]$ and $\alpha \in \{0.25, 1, 2.5, 5, 10, 50\}$ to the ultimate bound ψ_1 in (8.42), where the arrow indicate the direction α is increased.

8.3.3 Illustrative Numerical Example

We now present several numerical examples to illustrate the results given earlier in this section. For this purpose, consider a process composed of two decoupled systems with the dynamics given by (8.5),

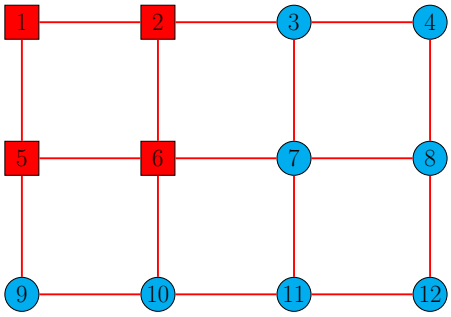


Figure 8.2: Communication graph of the sensor network in Example 1 with 4 active nodes and 8 passive nodes (lines denote communication links, squares denote active nodes, and circles denote passive nodes).

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_{n1}^2 & -2\omega_{n1}\xi_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_{n2}^2 & -2\omega_{n2}\xi_2 \end{bmatrix}, \tag{8.50}$$

$$B = \begin{bmatrix} 0 & 0 \\ \omega_{n1}^2 & 0 \\ 0 & 0 \\ 0 & \omega_{n2}^2 \end{bmatrix}, \tag{8.51}$$

$\omega_{n1} = 1.2$, $\xi_1 = 0.9$, $\omega_{n2} = 0.5$, and $\xi_2 = 0.6$. This process, for example, can represent a linearized vehicle model with the first and third states corresponding to the positions in the x and y directions, respectively, while the second and fourth states corresponding to the velocities in the x and y directions, respectively. The initial conditions are set to $x_0^{\mathrm{T}} = [-3,\ 0.5,\ 2.5,\ 0.25]$. In addition, we consider the input is given by

$$w(t) = \begin{bmatrix} 2.5\sin(t) \\ 4\cos(1.2t) \end{bmatrix}. \tag{8.52}$$

To maintain the readability of the paper, the values of L_i, σ_i, P_i in the following examples are put in Appendix B.

8.3.3.1 Example 1

For the first example, we consider a sensor network with 12 nodes exchanging information over an undirected and connected graph topology, where there are 4 active nodes and 8 passive nodes as shown in Figure 8.2. Each node's sensing capability is represented by (8.6) with the output matrices

$$C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{8.53}$$

for the odd index nodes and

$$C_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8.54}$$

for the even index nodes. In addition, all nodes are subject to zero initial conditions and we set $J_i = K_i = \text{diag}([100;100])$, $\alpha = 50$, and $\gamma = 0.1$. For the observer gain L_i, the odd index nodes are subject to (D.1) while the even index nodes are subject to (D.2).

By solving the linear matrix inequality (8.9) for each node, σ_i and $P_i > 0$ are obtained as $\sigma_1 = \sigma_5$, $\sigma_2 = \sigma_6$, $\sigma_3 = \sigma_4 = \sigma_7 = \sigma_8 = \sigma_9 = \sigma_{10} = \sigma_{11} = \sigma_{12}$ where σ_1, σ_2 and σ_3 are subject to (D.3), (D.4) and (D.5), respectively. In addition, P_1, P_2 and P_{12} are subject to (D.6), (D.7), and (D.8), respectively. Note that $P_1 = P_5$, $P_2 = P_6$ and $P_3 = P_4 = P_7 = P_8 = P_9 = P_{10} = P_{11} = P_{12}$. Under the proposed distributed estimation architecture (8.7) and (8.8), nodes are able to closely estimate the process states and inputs as shown in Figure 8.3 and 8.4, respectively. ▲

8.3.3.2 Example 2

In this example, we increase the number of active nodes in the sensor network to 8 as depicted in Figure 8.5. The sensing capability of each agent is the same as in Example 1. Note that, because of the change in the number of active nodes, the design parameters are adjusted accordingly as $\sigma_1 = \sigma_3 = \sigma_5 = \sigma_7$, $\sigma_2 = \sigma_4 = \sigma_6 = \sigma_8$, $\sigma_9 = \sigma_{10} = \sigma_{11} = \sigma_{12}$ where σ_1, σ_2 and σ_9 are subjected to (D.9), (D.10) and (D.11), respectively. In addition, $P_1 = P_3 = P_5 = P_7$, $P_2 = P_4 = P_6 = P_8$, $P_9 = P_{10} = P_{11} = P_{12}$, where P_1, P_2 and P_{12} are the same as (D.6), (D.7), and (D.8), respectively. Other parameters and gains are also kept the same. Figures 8.6 and 8.7 show the performance of the sensor network for the proposed distributed estimation

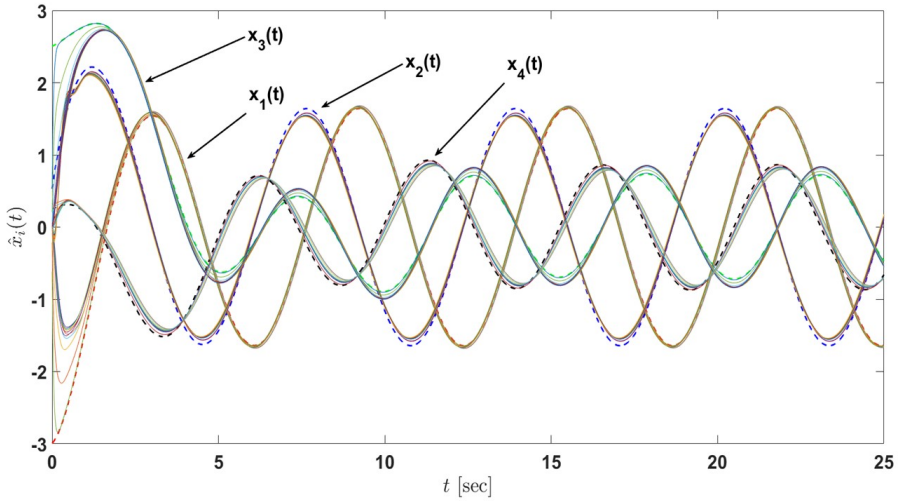


Figure 8.3: State estimates of the sensor network in Example 1 with 4 active nodes and 8 passive nodes under the proposed architecture (8.7) and (8.8) (the dash lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

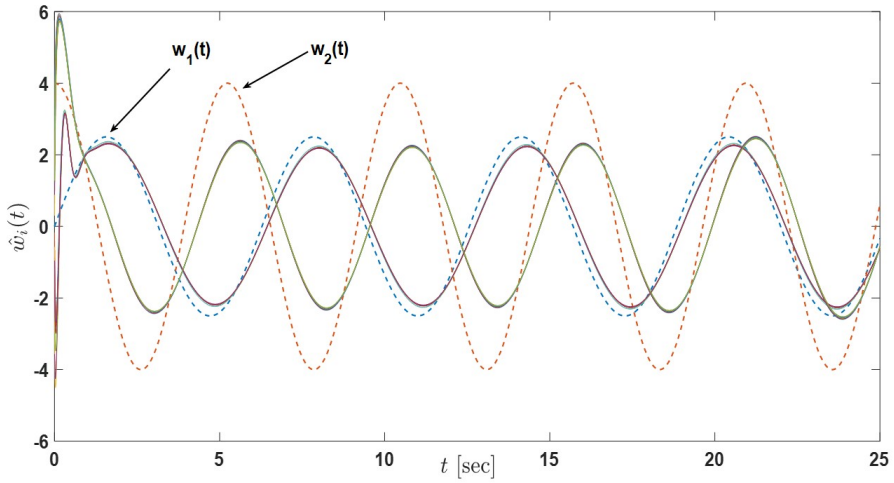


Figure 8.4: Input estimates of the sensor network in Example 1 with 4 active nodes and 8 passive nodes under the proposed architecture (8.7) and (8.8) (the dash lines denote the inputs of the actual process and the solid lines denote the input estimates of nodes).

architecture. In addition, in order to compare the performance of Example 1 and Example 2, the state and input error norms of both examples are plotted in Figures 8.8 and 8.9, respectively. The transient responses are captured in the figures approximately during the first two or three seconds, and it can be seen that Example 2 converges faster than Example 1 in state estimation, yet it encounters overshoot in input estimation. In addition, we can roughly approximate the average of both state and input error norms are

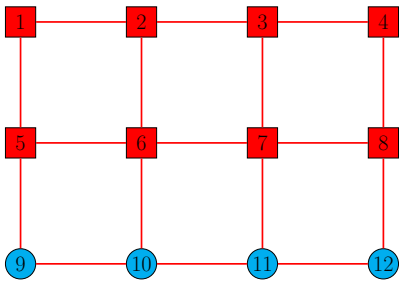


Figure 8.5: Communication graph of the sensor network in Example 2 and 3 with 8 active nodes and 4 passive nodes (lines denote communication links, squares denote active nodes, and circles denote passive nodes).

reduced by a factor of 2 in Example 2 compared to Example 1. In general, Examples 1 and 2 show that the steady state performance is improved by increasing the number of active nodes in the sensor network. ▲

8.3.3.3 Example 3

In this example, we consider a sensor network with 8 active nodes and 4 passive nodes as in Example 2 (Figure 8.5), but change the system output matrices for each node as follows

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{8.55}$$

$$C_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{8.56}$$

$$C_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{8.57}$$

where $C_1 = C_5 = C_9$, $C_2 = C_4 = C_6 = C_8 = C_{10} = C_{12}$ and $C_3 = C_7 = C_{11}$. Note that for the odd index nodes, the pair (A, C_i) is not observable. We also choose $J_i = K_i = \text{diag}([100; 100])$, $\alpha = 50$, and $\gamma = 0.1$.

Here, the observer gain L_i is chosen such that $L_1 = L_5 = L_9$, $L_2 = L_4 = L_6 = L_8 = L_{10} = L_{12}$ and $L_3 = L_7 = L_{11}$ where L_1, L_2 and L_3 are subject to (D.12), (D.13) and (D.14), respectively. By solving the linear matrix inequality (8.9) for each node, σ_i and $P_i > 0$ are obtained as $\sigma_1 = \sigma_5$, $\sigma_2 = \sigma_4 = \sigma_6 = \sigma_8$, $\sigma_3 = \sigma_7$, $\sigma_9 = \sigma_{10} = \sigma_{11} = \sigma_{12}$ where σ_1, σ_2, σ_3 and σ_9 are subject to (D.15), (D.16), (D.17) and (D.18), respectively. In addition, $P_1 = P_5$, $P_2 = P_4 = P_6 = P_8$, $P_3 = P_7$, and $P_9 = P_{10} = P_{11} = P_{12}$ where P_1, P_2, P_3 and P_{12} are subject to (D.19), (D.7), (D.20) and (D.8), respectively. Figure 8.10 and 8.11 show that under

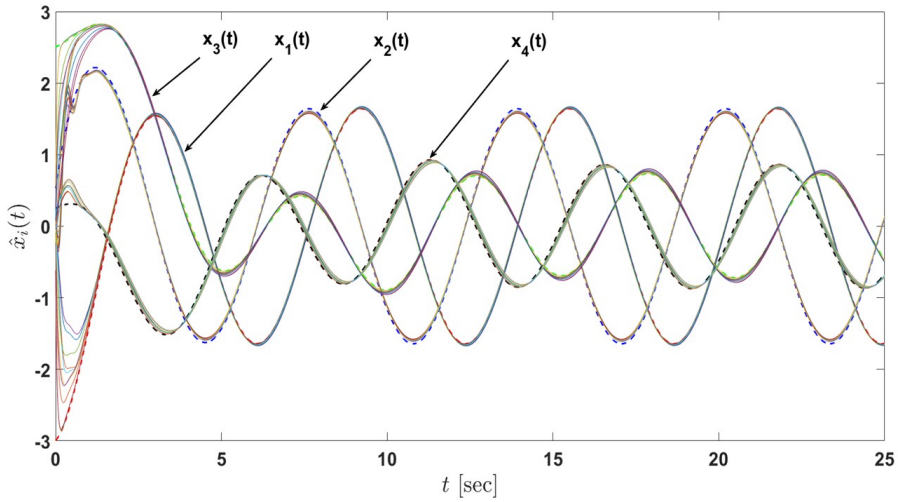


Figure 8.6: State estimates of the sensor network in Example 2 with 8 active nodes and 4 passive nodes under the proposed architecture (8.7) and (8.8) (the dash lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

the proposed distributed estimation architecture, nodes are able to closely estimate the process states and inputs, although some active nodes are not able to fully observe the process. ▲

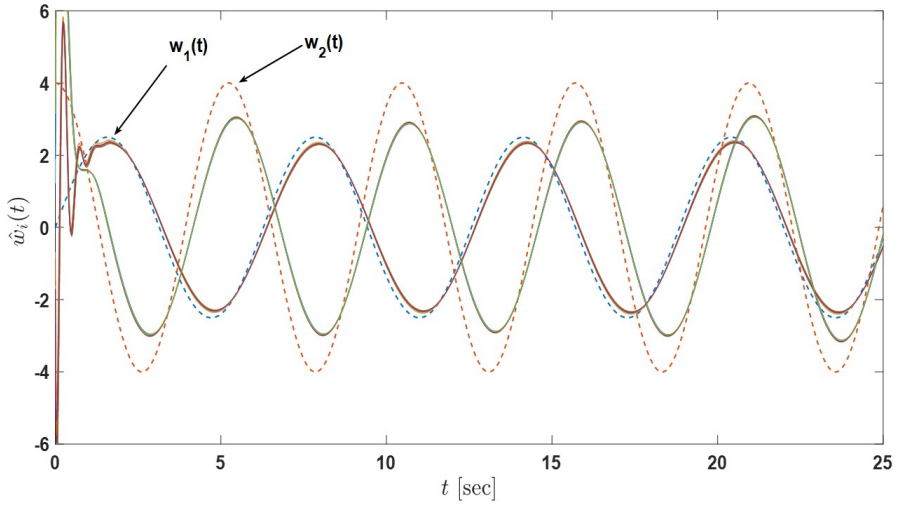


Figure 8.7: Input estimates of the sensor network in Example 2 with 8 active nodes and 4 passive nodes under the proposed architecture (8.7) and (8.8) (the dash lines denote the inputs of the actual process and the solid lines denote the input estimates of nodes).

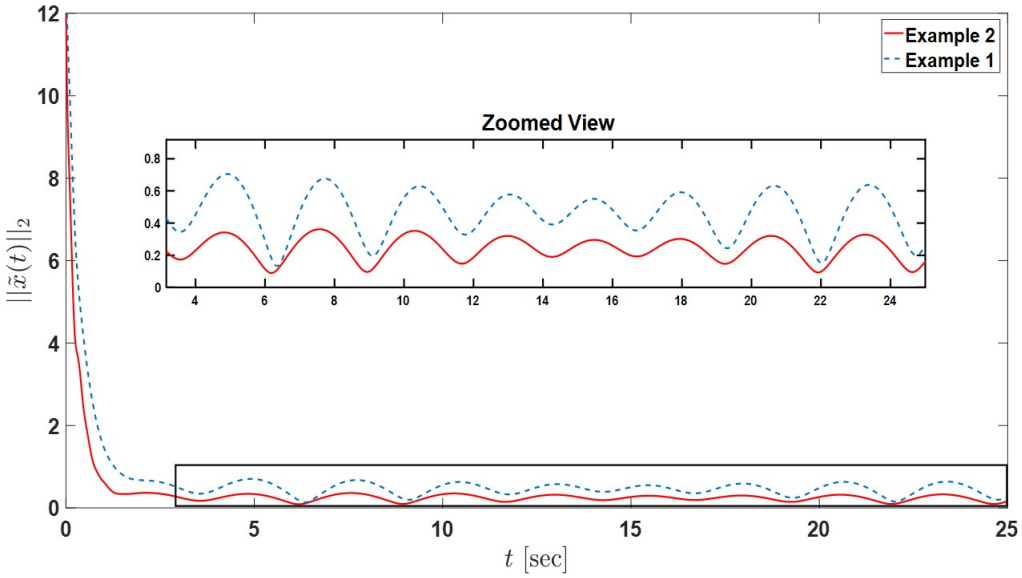


Figure 8.8: State error norms of the sensor networks in Example 1 and Example 2.

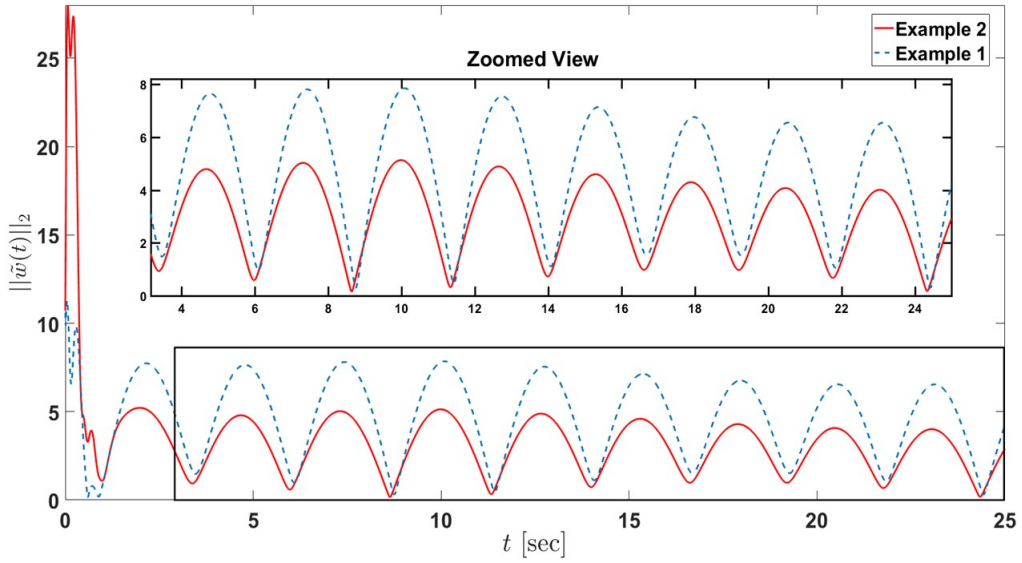


Figure 8.9: Input error norms of the sensor networks in Example 1 and Example 2.

8.4 Distributed Input and State Estimation for Active-Passive Sensor Networks with Varying Node Roles

We now generalize the results of the previous section to the case when the active and passive role of each sensor node is varying over time. For this purpose, once again, we consider a process given by (8.5). In addition, if a node in the sensor network is active for some time instant, then it is subject to the observations of the process given by (8.6) on that time instant, otherwise it is a passive node and has no observation. Note

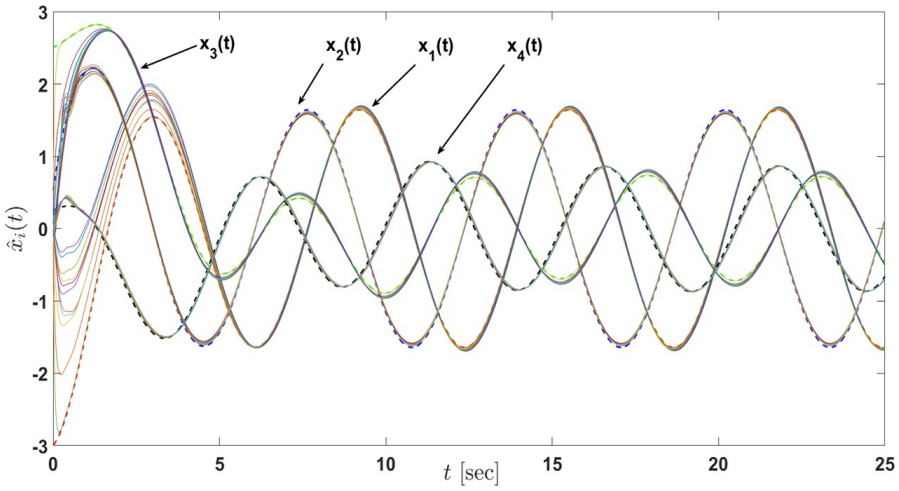


Figure 8.10: State estimates of the sensor network in Example 3 with 12 active nodes under the proposed architecture (8.7) and (8.8) (the dash lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

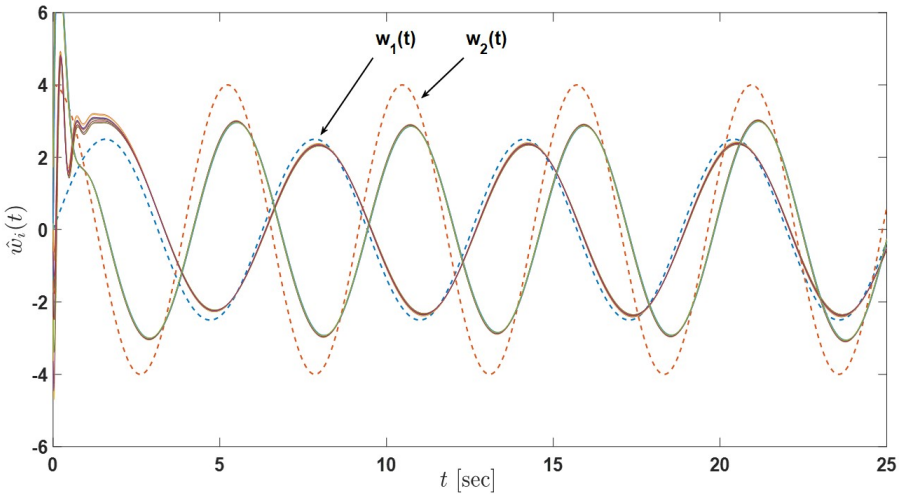


Figure 8.11: Input estimates of the sensor network in Example 3 with 12 active nodes under the proposed architecture (8.7) and (8.8) (the dash lines denote the inputs of the actual process and the solid lines denote the input estimates of nodes).

that a node is assumed to be smoothly changed back and forth between active and passive mode (i.e. $g_i(t)$ is a smooth function on the interval $[0,1]$). The proposed algorithm is discussed in Section 8.4.1, followed by the stability analysis (Section 8.4.2), and a numerical example is presented to illustrate the efficacy of the methods (Section 8.4.3).

8.4.1 Proposed Distributed Estimation Architecture

For node i, $i = 1,\ldots,N$, consider the distributed estimation algorithm given by

$$\dot{\hat{x}}_i(t) = (A-\gamma P_i^{-1})\hat{x}_i(t) + B\hat{w}_i(t) + g_i(t)L_i\big(y_i(t) - C_i\hat{x}_i(t)\big) - \alpha P_i^{-1}\sum_{i\sim j}(\hat{x}_i(t) - \hat{x}_j(t)),$$
$$\hat{x}_i(0) = \hat{x}_{i0}, \quad (8.58)$$

$$\dot{\hat{w}}_i(t) = g_i(t)J_i(y_i(t) - C_i\hat{x}_i(t)) - (\sigma_i K_i + \gamma \mathrm{I}_p)\hat{w}_i(t) - \alpha\sum_{i\sim j}(\hat{w}_i(t) - \hat{w}_j(t)), \quad \hat{w}_i(0) = \hat{w}_{i0}, \quad (8.59)$$

where $\hat{x}_i(t) \in \mathbb{R}^n$ is a local state estimate of $x(t)$ for node i, $\hat{w}_i \in \mathbb{R}^p$ is a local input estimate of $w(t)$ for node i, $L_i \in \mathbb{R}^{n\times m}$, $J_i \in \mathbb{R}^{p\times m}$ and $K_i \in \mathbb{S}_+^{p\times p}$ are design matrices of node i, and α, γ, and $\sigma_i \in \mathbb{R}$ are positive design coefficients for node i. Note that the parameter $g_i(t)$ in this section is time-varying and $g_i(t) \in [0,1]$. In addition, $P_i > 0$ is the consensus gain satisfying the two linear matrix inequalities given by

$$R_{i1} \triangleq \begin{bmatrix} A^{\mathrm{T}}P_i + P_iA & -P_iB \\ -B^{\mathrm{T}}P_i & -2\sigma_iK_i \end{bmatrix} \leq 0, \quad (8.60)$$

$$R_{i2} \triangleq \begin{bmatrix} (A-L_iC_i)^{\mathrm{T}}P_i + P_i(A-L_iC_i) & -P_iB + C_i^{\mathrm{T}}J_i^{\mathrm{T}} \\ -B^{\mathrm{T}}P_i + J_iC_i & -2\sigma_iK_i \end{bmatrix} \leq 0, \quad i = 1,\ldots,N. \quad (8.61)$$

8.4.2 Stability Analysis

Let $\tilde{x}_i(t) \triangleq x(t) - \hat{x}_i(t)$ and $\tilde{w}_i(t) \triangleq \hat{w}_i(t) - w(t)$. Then, similar to (8.11) and (8.12), one can write

$$\dot{\tilde{x}}_i(t) = \bar{A}_i(t)\tilde{x}_i(t) - B\tilde{w}_i(t) - \alpha P_i^{-1}\sum_{i\sim j}(\tilde{x}_i(t) - \tilde{x}_j(t)) - \gamma P_i^{-1}\big(\tilde{x}_i(t) - x(t)\big), \quad \tilde{x}_i(0) = \tilde{x}_{i0}, \quad (8.62)$$

$$\dot{\tilde{w}}_i(t) = g_i(t)J_iC_i\tilde{x}_i(t) - \sigma_iK_i(\tilde{w}_i(t) + w(t)) - \alpha\sum_{i\sim j}(\tilde{w}_i(t) - \tilde{w}_j(t)) - \gamma(\tilde{w}_i(t) + w(t)) - \dot{w}(t),$$
$$\tilde{w}_i(0) = \tilde{w}_{i0}, \quad (8.63)$$

where

$$\bar{A}_i(t) \triangleq A - g_i(t)L_iC_i. \quad (8.64)$$

Therefore, similar to Section 8.3.2, the compact form of the error dynamics are given by

$$\dot{\tilde{x}}(t) = \bar{A}(t)\tilde{x}(t) - (\mathrm{I}_N \otimes B)\tilde{w}(t) - P^{-1}(F\otimes \mathrm{I}_n)\tilde{x}(t) + \gamma P^{-1}(\mathbf{1}_N \otimes \mathrm{I}_n)x(t), \quad (8.65)$$

$$\dot{\tilde{w}}(t) = M(t)\tilde{x}(t) - \bar{K}\big(\tilde{w}(t) + (\mathbf{1}_N\otimes \mathrm{I}_p)w(t)\big) - (F\otimes \mathrm{I}_p)\tilde{w}(t) - \gamma(\mathbf{1}_N\otimes \mathrm{I}_p)w(t) - (\mathbf{1}_N\otimes \mathrm{I}_p)\dot{w}(t), \quad (8.66)$$

where

$$\bar{A}(t) \triangleq \operatorname{diag}([\bar{A}_1(t), \bar{A}_2(t), \ldots, \bar{A}_N(t)]), \tag{8.67}$$

$$M(t) \triangleq \operatorname{diag}([g_1(t)J_1C_1, g_2(t)J_2C_2, \ldots, g_N(t)J_NC_N]), \tag{8.68}$$

and $\bar{K}$, F and P are the same as (8.21), (8.22), and (8.23), respectively.

Theorem 8.4.1. *Consider the process given by (8.5) and the distributed input and state estimation architecture given by (8.58) and (8.59). Assume (8.60) and (8.61) hold and nodes exchange information using local measurements subject to an undirected and connected graph $\mathcal{G}$. Then, the error dynamics given by (8.65) and (8.66) are uniformly ultimately bounded.*

Proof. Consider the Lyapunov function candidate given by (8.24). Following the steps from the proof of Theorem 8.3.1, differentiating (8.24) along the trajectories of (8.65) and (8.66) yields

$$\dot{V}(\cdot) = z^{\mathrm{T}}(t)R_A(t)z(t) + z^{\mathrm{T}}(t)R_Bz(t) + 2z^{\mathrm{T}}(t)\phi, \tag{8.69}$$

where $z(t)$, R_B, and ϕ are defined in (8.26), (8.28), and (8.30), respectively. In addition,

$$R_A(t) \triangleq \begin{bmatrix} \bar{A}(t)^{\mathrm{T}}P + P\bar{A}(t) & -P(\mathrm{I}_N \otimes B) + M^{\mathrm{T}}(t) \\ -(\mathrm{I}_N \otimes B^{\mathrm{T}})P + M(t) & -2\bar{K} \end{bmatrix}. \tag{8.70}$$

Note that for this varying case of active and passive node roles, R_i in (8.9) becomes

$$\begin{aligned} R_i(t) &= \begin{bmatrix} \big(A - g_i(t)L_iC_i\big)^{\mathrm{T}}P_i + P_i\big(A - g_i(t)L_iC_i\big) & -P_iB + g_i(t)C_i^{\mathrm{T}}J_i^{\mathrm{T}} \\ -B^{\mathrm{T}}P_i + g_i(t)J_iC_i & -2\sigma_iK_i \end{bmatrix} \\ &= \begin{bmatrix} A^{\mathrm{T}}P_i + P_iA & -P_iB \\ -B^{\mathrm{T}}P_i & -2\sigma_iK_i \end{bmatrix} + g_i(t)\begin{bmatrix} (-L_iC_i)^{\mathrm{T}}P_i + P_i(-L_iC_i) & C_i^{\mathrm{T}}J_i^{\mathrm{T}} \\ J_iC_i & 0 \end{bmatrix} \end{aligned} \tag{8.71}$$

Since $g_i(t) \in [0,1]$, R_{i1} in (8.60) and R_{i2} in (8.61) corresponds to $g_i(t) = 0$ and $g_i(t) = 1$ in (8.71), respectively. Therefore, R_{i1} and R_{i2} are the vertices of the polytope. By Lemma 8.2.3, when the linear matrix inequalities (8.60) and (8.61) hold, $R_i(t) \leq 0$ for all $g_i(t) \in [0,1]$. Consequently, using the same argument as

in the proof of Theorem 8.3.1, we have $R_A(t) \leq 0$. Hence, (8.69) becomes

$$\begin{aligned}\dot{V}(\cdot) &= z^{\mathrm{T}}(t)R_A(t)z(t) + z^{\mathrm{T}}(t)R_B z(t) + 2z^{\mathrm{T}}(t)\phi \\ &\leq \lambda_{\max}(R_B)\|z(t)\|_2^2 + 2\|z(t)\|_2\bar{\phi} \\ &\leq (1-\theta)\lambda_{\max}(R_B)\|z(t)\|_2^2 + \theta\lambda_{\max}(R_B)\|z(t)\|_2^2 + 2\|z(t)\|_2\bar{\phi}, \end{aligned} \tag{8.72}$$

with $\lambda_{\max}(R_B) < 0$ and $\theta \in (0,1)$. Letting $\mu_2 \triangleq \frac{-2\bar{\phi}}{\theta\lambda_{\max}(R_B)} > 0$ and $\Omega_2 \triangleq \{z(t) : \|z(t)\|_2 \leq \mu_2\}$, it follows that $\dot{V}(\cdot) \leq (1-\theta)\lambda_{\max}(R_B)\|z(t)\|_2^2 < 0$ outside the compact set Ω_2, and hence, the error dynamics given by (8.62) and (8.63) are uniformly ultimately bounded from Theorem 4.18 in [119]. ■

Corollary 8.4.1. *Consider the process given by (8.5) and the distributed input and state estimation architecture given by (8.58) and (8.59). Assume (8.60) and (8.61) hold and nodes exchange information using local measurements subject to an undirected and connected graph $\mathcal{G}$. Then, for all $z(0) \in \mathbb{R}^{N(n+p)}$, there exists $T = T\big(z(0), \mu_2\big) \geq 0$ such that*

$$\|\tilde{x}(t)\|_2 \leq \xi_2 \triangleq \sqrt{\frac{\lambda_{\max}(\bar{P})}{\lambda_{\min}(\bar{P})}}\max\left\{\|z(0)\|_2\, e^{\left((1-\theta)\lambda_{\max}(R_B)/2\lambda_{\max}(\bar{P})\right)t}, \mu_2\right\}, \quad \forall t \geq 0, \tag{8.73}$$

$$\|\tilde{w}(t)\|_2 \leq \xi_2, \quad \forall t \geq 0, \tag{8.74}$$

where

$$\bar{P} = \begin{bmatrix} P & 0 \\ 0 & \mathrm{I}_{Np} \end{bmatrix}, \tag{8.75}$$

and

$$\|\tilde{x}(t)\|_2 \leq \psi_2 \triangleq \sqrt{\frac{\lambda_{\max}(\bar{P})}{\lambda_{\min}(P)}}\mu_2, \quad t \geq T, \tag{8.76}$$

$$\|\tilde{w}(t)\|_2 \leq \zeta_2 \triangleq \sqrt{\lambda_{\max}(\bar{P})}\mu_2, \quad t \geq T. \tag{8.77}$$

Proof. Same theoretical steps follow from the proof of Corollary 8.3.1, and hence, the proof is omitted here. ■

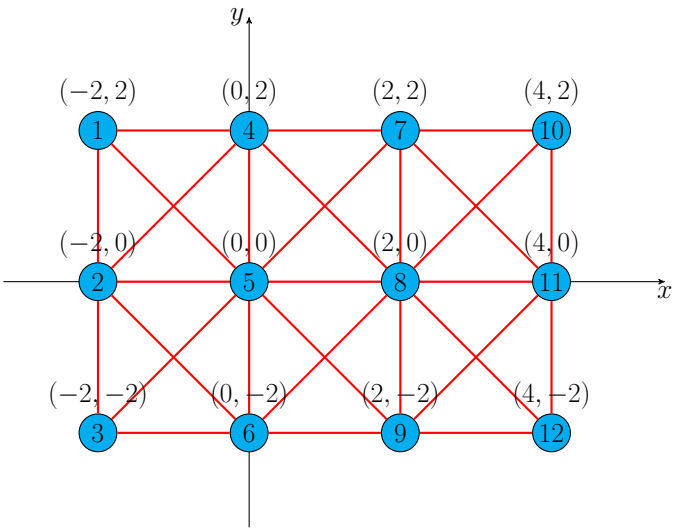


Figure 8.12: Communication graph of the active-passive sensor network in Example 4 with 12 nodes (lines denote communication links, circles denote nodes).

8.4.3 Illustrative Numerical Example

In this section, we present numerical examples to illustrate the results discussed in Section 8.4.1 and 8.4.2. For this purpose, we consider a process as the vehicle model depicted in Section 8.3.3 with the dynamics given by (8.5), where A and B are defined in (8.50) and (8.51), respectively. To maintain the readability of the paper, the values of L_i, σ_i, P_i in the following examples are put in Appendix B.

8.4.3.1 Example 4

For this example, the initial conditions of the process are set to $x_0^{\mathrm{T}} = [-3,\ 0.5,\ 2.5,\ 0.25]$. In addition, we consider the input is given by

$$w(t) = \begin{bmatrix} 2.5\sin(t) \\ 3.5\cos(1.2t) \end{bmatrix}. \tag{8.78}$$

We now consider an active-passive sensor network with 12 nodes exchanging information over an undirected and connected graph topology as presented in Figure 8.12, where the active and passive role of each node is varying overtime. Specifically, the sensors are distributed over an area, and each sensor position is shown in Figure 8.12. Suppose that each sensor sensing range is a circle with the radius $r = 3$. Recall that the first and third states of the process (or the vehicle) correspond to the positions in the x-axis

and y-axis directions, respectively. If the vehicle's position is within a sensor sensing range, then that sensor becomes smoothly active. On the other hand, if the vehicle's position is out of the sensor sensing range, then it becomes smoothly passive. Note that, for the transition of $g_i(t)$, we use the function $g_i(t) = e^{-\beta t}$ when node i is switching from 1 to 0, and $g_i(t) = 1 - e^{-\beta t}$ when node i is switching from 0 to 1, where β is a positive constant. We adapt this transition from Figure 2(d) of [152]. The network has two types of sensors, and each node's sensing capability is represented by (8.6) with the output matrices

$$C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{8.79}$$

for the odd index nodes and

$$C_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8.80}$$

for the even index nodes. Note that the pair (A, C_i) is observable for all $i = 1, \ldots, 12$ in this example, and therefore, the collective observability assumption is satisfied. All nodes are subjected to zero initial conditions and we set $J_i = K_i = \text{diag}([100; 100])$, $\alpha = 50$, and $\gamma = 0.1$. For the observer gain L_i, the odd index nodes are subject to (D.21) while the even index nodes are subject to (D.22).

By solving the linear matrix inequalities (8.60) and (8.61) simultaneously for each node, σ_i and $P_i > 0$ are obtained as $\sigma_1 = \sigma_3 = \sigma_5 = \sigma_7 = \sigma_9 = \sigma_{11}$, $\sigma_2 = \sigma_4 = \sigma_6 = \sigma_8 = \sigma_{10} = \sigma_{12}$ where σ_1 and σ_2 are subject to (D.23) and (D.24), respectively. In addition, P_1 and P_2 are subject to (D.25) and (D.26), respectively. Note that $P_1 = P_3 = P_5 = P_7 = P_9 = P_{11}$ and $P_2 = P_4 = P_6 = P_8 = P_{10} = P_{12}$.

Under the proposed distributed estimation architecture (8.58) and (8.59), nodes are able to closely estimate the process states as shown in Figure 8.13. Specifically, Figure 8.14 illustrates that the sensor network is able to estimate the trajectory of the vehicle (the first and third state of the process), while the input estimated in Figure 8.15 is not as good as the case for fixed node roles presented in Section 8.3.3, this can be explained by the conservatism of the solution of the linear matrix inequalities (8.60) and (8.61). That is, if we had the flexibility to make the σ_i values small such that $\sigma_1 = \sigma_3 = \sigma_5 = \sigma_7 = \sigma_9 = \sigma_{11} = 0.001$, $\sigma_2 = \sigma_4 = \sigma_6 = \sigma_8 = \sigma_{10} = \sigma_{12} = 0.001$, while keeping P_i and other parameters the same, the performance of the input and state estimate would become better as shown in Figure 8.16 and 8.17, respectively. However, with

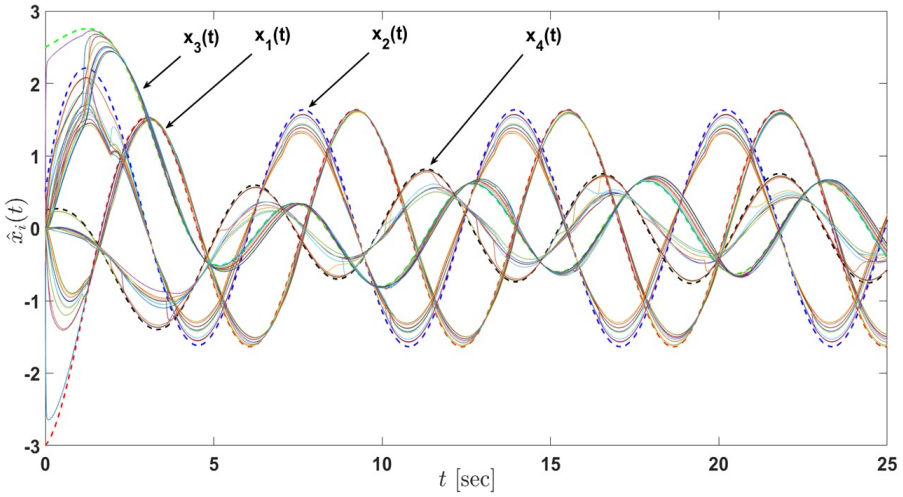


Figure 8.13: State estimates of the active-passive sensor network in Example 4 with 12 nodes under the proposed architecture (8.58) and (8.59) and satisfying the linear matrix inequalities (8.60) and (8.61) (the dash lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

these small values of σ_i, the linear matrix inequalities (8.60) and (8.61) are no longer satisfied. Numerical methods to reduce such conservatism in linear matrix inequality computations for (8.60) and (8.61) and/or relax the linear matrix inequality condition will be investigated as a future research.

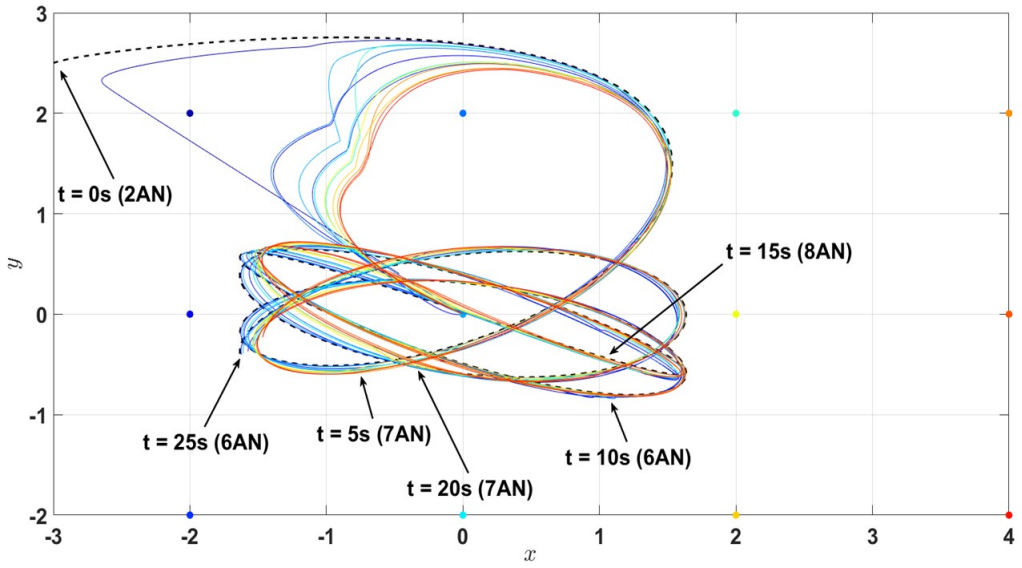


Figure 8.14: Position estimates (first and third states of the process) of the active-passive sensor network in Example 4 with 12 nodes under the proposed architecture (8.58) and (8.59) and satisfying the linear matrix inequalities (8.60) and (8.61) (the dash line denote the trajectory of the actual process (i.e. the combination of the first and third state) and the solid lines denote the state estimates of nodes). Here, AN stands for the the active nodes.

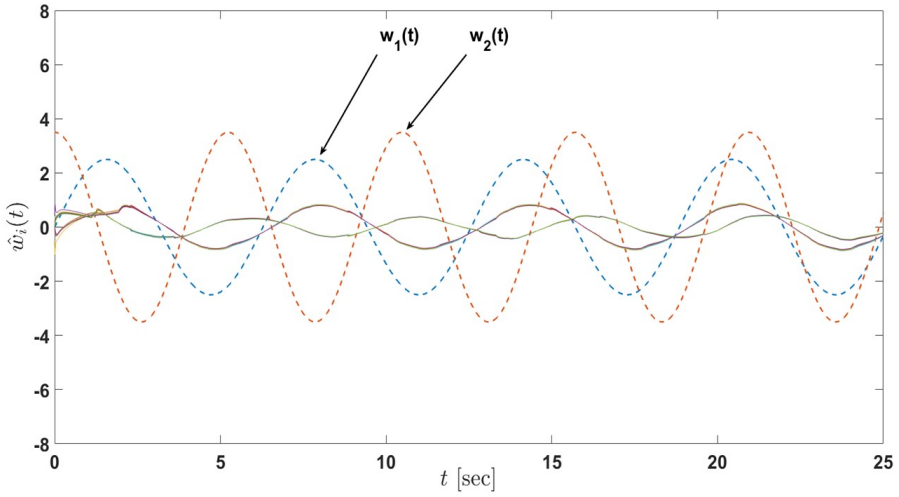


Figure 8.15: Input estimates of the active-passive sensor network in in Example 4 with 12 nodes under the proposed architecture (8.58) and (8.59) and satisfying the linear matrix inequalities (8.60) and (8.61) (the dash lines denote the inputs of the actual process and the solid lines denote the input estimates of nodes).

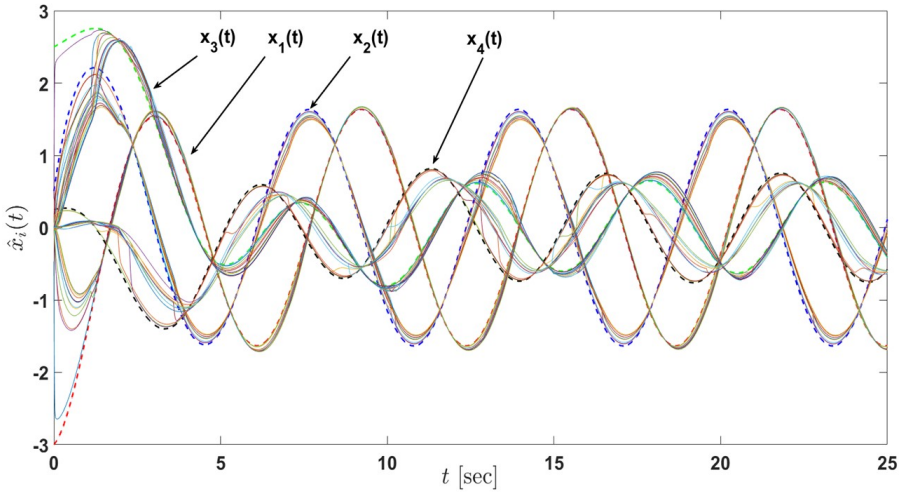


Figure 8.16: State estimates of the active-passive sensor network in Example 4 with 12 nodes under the proposed architecture (8.58) and (8.59) with the decrease in σ_i, $i = 1, \ldots, 12$ (the dash lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

8.4.3.2 Example 5

For this example, the initial conditions of the process are set to $x_0^{\mathrm{T}} = [-2,\ 0.5,\ 2.5,\ 0.25]$. In addition, we consider the input is given by

$$w(t) = \begin{bmatrix} 1.5\sin(0.5t) \\ 3\cos(0.6t) \end{bmatrix}. \tag{8.81}$$

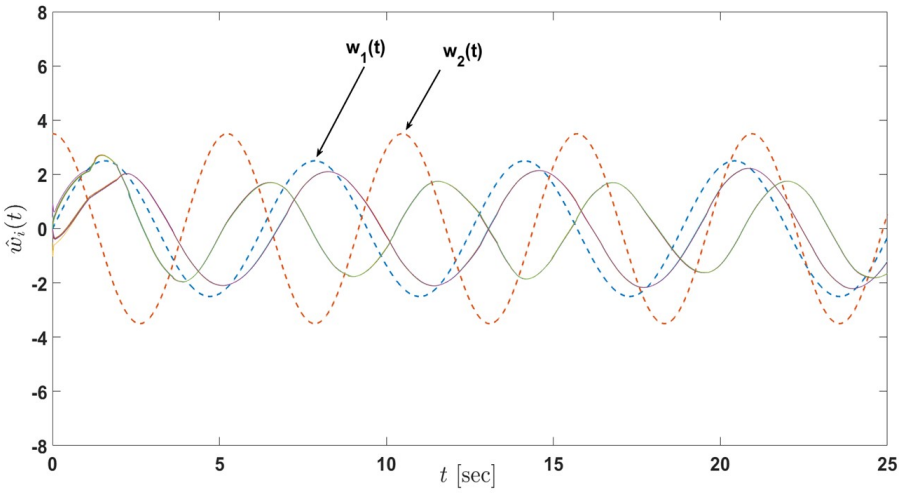


Figure 8.17: Input estimates of the active-passive sensor network in Example 4 with 12 nodes under the proposed architecture (8.58) and (8.59) with the decrease in σ_i, $i = 1, \ldots, 12$ (the dash lines denote the inputs of the actual process and the solid lines denote the input estimates of nodes).

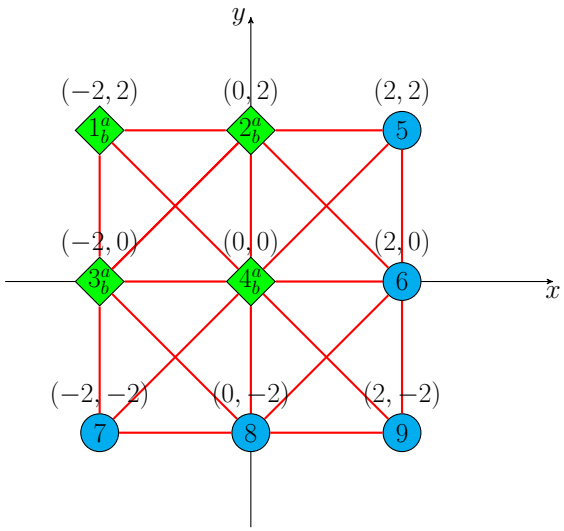


Figure 8.18: Communication graph of the active-passive sensor network in Example 5 with 13 nodes (lines denote communication links, circles denote normal nodes, and diamond denote overlapped nodes).

We now consider an active-passive sensor network with 13 nodes labeled respectively as 1a, 1b, 2a, 2b, 3a, 3b, 4a, 4b, 5, 6, 7, 8 and 9, exchanging information over an undirected and connected graph topology as presented in Figure 8.18, where the active and passive role of each node is varying overtime. Specifically, the sensors are distributed over an area with each pair of nodes Xa and Xb (where X = 1, 2, 3, 4 and denoted as diamond in Figure 8.18) is grouped at the same location such that when Xa is active (or passive), so is Xb and vice versa; Xa and Xb are neighbors of each other and have the same set of neighbors.

Suppose that each sensor sensing range is a circle with the radius $r = 3.5$. Note that, for the transition of $g_i(t)$, we use the same functions as Example 4. The network has six types of sensors, and each node's sensing capability is represented by (8.6) with the output matrices

$$C_{1a} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{8.82}$$

$$C_{1b} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{8.83}$$

$$C_{2b} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{8.84}$$

$$C_{3a} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{8.85}$$

$$C_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{8.86}$$

$$C_6 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8.87}$$

In addition, $C_{1a} = C_{2a}$, $C_{1b} = C_{3b}$, $C_{2b} = C_{4b}$, $C_{3a} = C_{4a}$, $C_5 = C_7 = C_9$ and $C_6 = C_8$. Note that the pairs (A, C_{Xa}) and (A, C_{Xb}) where X = 1, 2, 3, 4, are not observable, but with the setup of the problem (nodes Xa and Xb are either both active or both passive simultaneously), the collective observability condition is guaranteed. All nodes are subjected to zero initial conditions and we set $J_i = K_i = \text{diag}([25; 25])$, $\alpha = 75$, and $\gamma = 0.01$. The observer gains L_i are set to $L_{1a} = L_{2a}$, $L_{1b} = L_{3b}$, $L_{2b} = L_{4b}$, $L_{3a} = L_{4a}$, $L_5 = L_7 = L_9$ and $L_6 = L_8$ where the gains $L_{1a}, L_{1b}, L_{2b}, L_{3a}, L_5$ and L_6 are subject to (D.27), (D.28), (D.29), (D.30), (D.31), and (D.32), respectively.

By solving the linear matrix inequalities (8.60) and (8.61) simultaneously for each node, σ_i and $P_i > 0$ are obtained as $\sigma_{1a} = \sigma_{2a}$, $\sigma_{1b} = \sigma_{3b}$, $\sigma_{2b} = \sigma_{4b}$, $\sigma_{3a} = \sigma_{4a}$, $\sigma_5 = \sigma_7 = \sigma_9$ and $\sigma_6 = \sigma_8$ where $\sigma_{1a}, \sigma_{1b}, \sigma_{2b}, \sigma_{3a}, \sigma_5$ and σ_6 are subject to (D.33), (D.34), (D.35), (D.36), (D.37), and (D.38), respectively. In addition, $P_{1a}, P_{1b}, P_{2b}, P_{3a}, P_5$ and P_6 are subject to (D.39), (D.40), (D.41), (D.42), (D.43), and (D.44), respectively. Note that $P_{1a} = P_{2a}$, $P_{1b} = P_{3b}$, $P_{2b} = P_{4b}$, $P_{3a} = P_{4a}$, $P_5 = P_7 = P_9$ and $P_6 = P_8$.

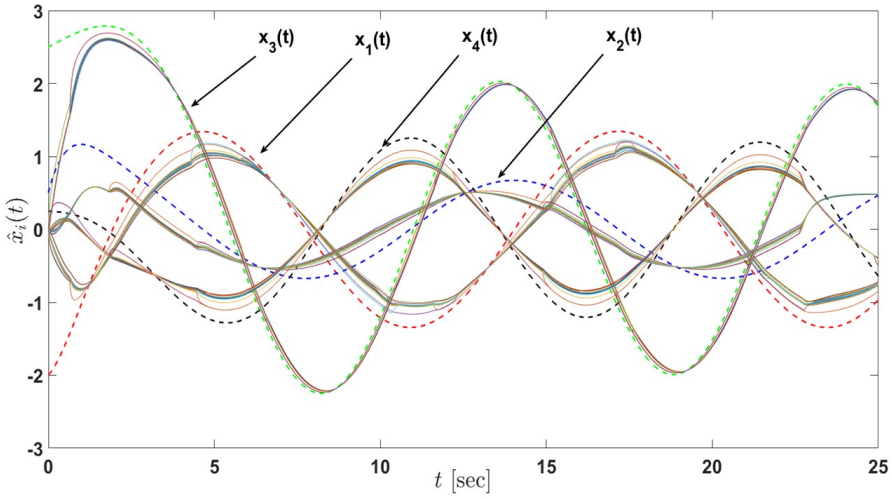


Figure 8.19: State estimates of the active-passive sensor network in Example 5 with 13 nodes under the proposed architecture (8.58) and (8.59) and satisfying the linear matrix inequalities (8.60) and (8.61) (the dash lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

Under the proposed distributed estimation architecture (8.58) and (8.59), nodes are able to closely estimate the process states as shown in Figure 8.19. Specifically, Figure 8.20 illustrates that the sensor network is able to estimate the trajectory of the vehicle (the first and third state of the process), while the input estimated in Figure 8.21 is still not as good as the case for fixed node roles presented in Section 8.3.3. Again, this can be explained by the conservatism of the solution of the linear matrix inequalities (8.60) and (8.61). If we had the flexibility to reduce σ_i to small values, for example, $\sigma_i = 0.001$, while keeping other parameters the same, the performance of the input and state estimate would become much better as shown in Figure 8.22 and 8.23, respectively. Note that with the choice of $\sigma_i = 0.001$ for this example, the linear matrix inequalities (8.60) and (8.61) are no longer satisfied. Once again, numerical methods to reduce such conservatism in linear matrix inequality computations for (8.60) and (8.61) and/or relax the linear matrix inequality condition will be investigated as a future research.

8.5 Conclusion

A distributed input and state estimation architecture was investigated for heterogeneous sensor networks having nodes with both fixed and varying active and passive information processing roles and nonidentical sensor modalities. It was shown that the proposed framework utilizes local information not only during the execution of the proposed estimation algorithm but also in its design; that is, global uniform ultimate boundedness of error dynamics is guaranteed once each node satisfies given local stability condi-

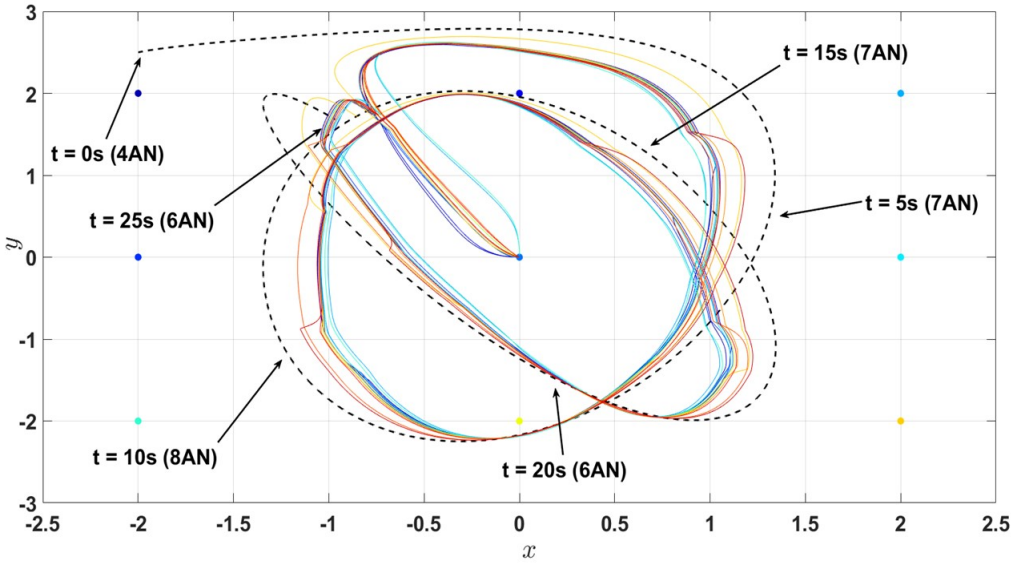


Figure 8.20: Position estimates (first and third states of the process) of the active-passive sensor network in Example 5 with 13 nodes under the proposed architecture (8.58) and (8.59) and satisfying the linear matrix inequalities (8.60) and (8.61) (the dash line denote the trajectory of the actual process (i.e. the combination of the first and third state) and the solid lines denote the state estimates of nodes). Here, AN stands for the the active nodes.

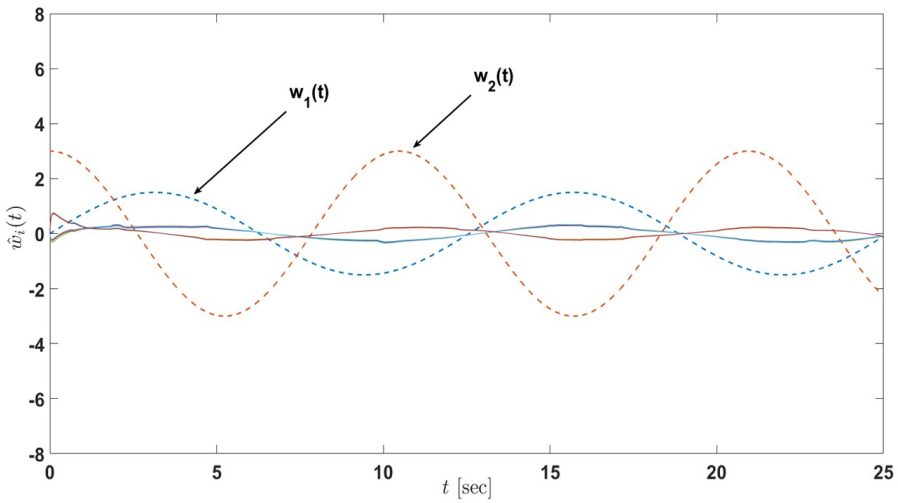


Figure 8.21: Input estimates of the active-passive sensor network in in Example 5 with 13 nodes under the proposed architecture (8.58) and (8.59) and satisfying the linear matrix inequalities (8.60) and (8.61) (the dash lines denote the inputs of the actual process and the solid lines denote the input estimates of nodes).

tions independent from the graph topology and neighboring information of these nodes. Several numerical examples illustrated the efficacy of the proposed architectures. Future research will include applications of the proposed framework to dynamic data-driven sensor network scenarios to guide and control autonomous vehicles and we will also consider extensions to time-varying graph topologies. It should be also mentioned

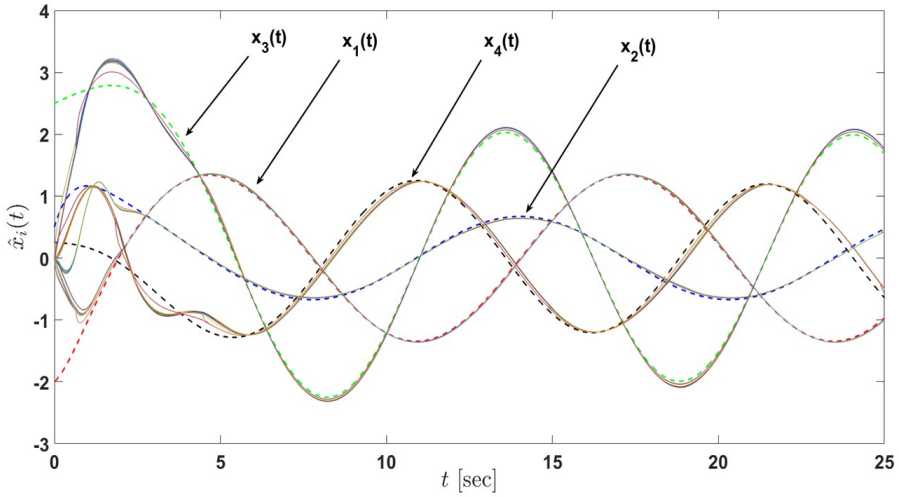


Figure 8.22: State estimates of the active-passive sensor network in Example 5 with 13 nodes under the proposed architecture (8.58) and (8.59) with the decrease in σ_i, $i = 1, \ldots, 12$ (the dash lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

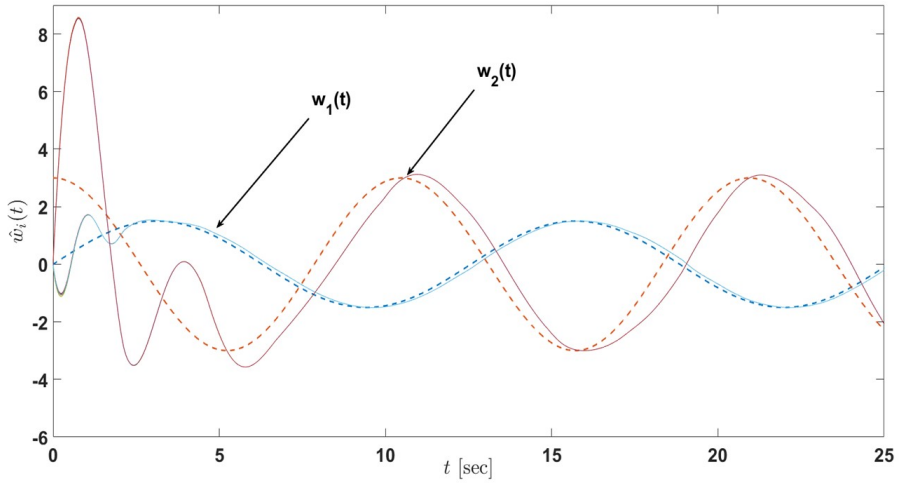


Figure 8.23: Input estimates of the active-passive sensor network in Example 5 with 13 nodes under the proposed architecture (8.58) and (8.59) with the decrease in σ_i, $i = 1, \ldots, 12$ (the dash lines denote the inputs of the actual process and the solid lines denote the input estimates of nodes).

especially for the results in Section 8.4 that structural sensor network construction to always guarantee collective observability is another interesting future research direction that will be considered by the authors.

8.6 Funding

This research has been supported by the Dynamic Data-Driven Applications Systems Program of the Air Force Office of Scientific Research.

8.7 Acknowledgments

The authors wish to thank the reviewers for their constructive suggestions.

Chapter 9: Distributed Coestimation in Heterogeneous Sensor Networks*

The contribution of this paper is a new system-theoretical dynamic information fusion framework for *heterogeneous sensor networks*, where a sensor network with both *nonidentical node information roles* and *nonidentical node modalities* is considered. Specifically, *nonidentical node information roles* allow nodes to be either *active* or *passive* in the sense that *active nodes* receive observations from a process of interest whereas *passive nodes* do not receive any information. In addition, active and passive roles of nodes can be *fixed* or *varying* with respect to time. Furthermore, *nonidentical node modalities* allow active nodes to receive different classes of measurements from the process. For this class of sensor networks, we propose a *distributed input and state coestimation* architecture, where the time evolution of input and state updates of each node *both* depend on the *local* input and state information exchanges. Using tools and methods from Lyapunov theory and linear matrix inequalities, we establish *stability and performance guarantees* of the overall heterogeneous sensor network executing the proposed distributed coestimation architecture under *local sufficient conditions* for each node. We also consider stochastic extensions that capture the practical aspect when the process and the node observations both include noise.

9.1 Introduction

9.1.1 Literature Review

Dynamic information fusion in sensor networks supports a wide array of scientific, civilian, and military data-driven applications, which range from reconnaissance and surveillance to command and control of vehicle swarms. Two common categories of dynamic information fusion are the Bayesian data fusion and the system-theoretical data fusion. While Bayesian methods regulate and fuse data according to probabilistic models (see, for example, [55–57] and references therein), system-theoretical methods (see, for example, [58–60] and references therein) focus on utilizing dynamic motions of given processes and

*This chapter has been submitted to the *International Journal of Control* for possible publication.

data exchange rules for controlling information fusion. This paper belongs to the latter category owing to attractive properties of system-theoretical data fusion in obtaining overall sensor network stability for real-world control, planning, and coordination applications.

Among several classes of heterogeneity in sensor networks, *nonidentical node information roles* and *nonidentical node modalities* are particularly important for numerous applications, which are considered in this paper. First, sensor networks are often associated with *nonidentical node information roles*. To elucidate this point, consider a representative application scenario shown in Figure 9.1. Here, the sensor network involves *active* and *passive nodes* in the sense that *active nodes* receive observations from the process whereas *passive nodes* do not receive any information. Note that the *active* and *passive node roles* also vary with respect to time in Figure 9.1; that is, active nodes can take passive roles during different time intervals and *vice versa*. Next, active nodes at any given time can practically have *nonidentical node modalities* in the sense that different measurements from the process can be observed.

There have been several papers in the literature that address heterogeneity in sensor networks, where the notable results can be listed as [1–3, 58–68, 70, 71, 73, 75, 153–156]. In particular, the authors of [58–65, 153] concentrate on dynamic consensus algorithms relevant to sensor networks; however, they assumed that all nodes are being active at all times. The authors of [66, 67] allow a subset of nodes being passive; however, these results are in the context of static consensus (that is, they are not suitable for dynamic data-driven applications). The authors of [68, 70, 71, 73, 75, 154, 155] focus on time-invariant (that is, fixed) and

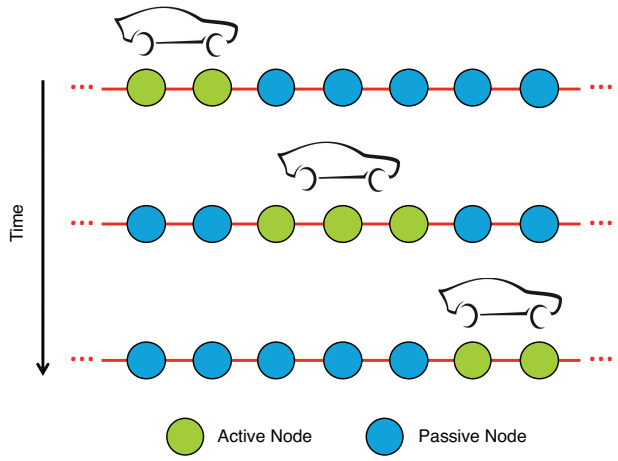


Figure 9.1: A dynamic information fusion scenario in a sensor network with time-varying active and passive node roles (lines and circles respectively denote communication links and nodes).

time-varying active and passive node roles; however, their results only consider nodes that are modeled as scalar integrator dynamics.

To address nonidentical node modalities in sensor networks, the authors of [1–3, 156] concentrate on nonidentical node modalities with their proposed distributed information fusion algorithms. However, it should be noted that [3] does not take into account the possibility of having passive nodes (that is, it requires all nodes are being active at all times). While fixed active and passive node roles are implicitly studied by the authors of [2], the contribution documented in this work requires a *global sufficient stability condition*, which may not be suitable for practical sensor networks composed of a (sufficiently) large set of nodes. More recently, nodes with nonidentical modalities are considered by the authors of [1] for both fixed and time-varying active and passive node roles under *local sufficient stability conditions* (we also refer to [157] for a preliminary version of the results in [1]). However, as discussed in Section 4.3 of [1], tuning the resulting distributed algorithm for satisfactory performance can be a challenge especially for the case when the active and passive node roles vary with respect to time. It should be also noted that the architecture proposed in [156] requires the measurements to be passed through local observers to extract more information before sending over the network for fusion.

9.1.2 Contribution

As discussed above, existing methods [1–3, 66–68, 70, 71, 73, 75, 154–157] do not provide a general system-theoretical dynamic information fusion architecture for addressing heterogeneity in sensor networks due to nonidentical node information roles and nonidentical node modalities. Motivated by this standpoint, this paper's contribution is a new, general system-theoretical dynamic information fusion framework for *heterogeneous sensor networks*, where a sensor network with both *nonidentical node information roles* and *nonidentical node modalities* is considered. For this class of sensor networks, we propose a *distributed input and state "coestimation"* architecture, where the time evolution of input and state updates of each node both depend on the *local* input and state information exchanges. Using tools and methods from Lyapunov theory and linear matrix inequalities, we establish *stability and performance guarantees* of the overall heterogeneous sensor network executing the proposed distributed coestimation architecture under *local sufficient conditions* for each node. We also consider stochastic extensions that capture the practical aspect when the process and the node observations both include noise. Finally, we present two numerical examples to demonstrate the efficacy of our theoretical contributions. As compared with the *distributed input*

and state "estimation" architecture in [1] discussed above, where the time evolution of input (respectively, state) update of each node *only* depends the local input (respectively, state) unlike the *distributed input and state "coestimation"* architecture of this paper, one of these examples further shows a substantially improved dynamic input and state fusion performance. Note that preliminary conference versions of this paper are appeared in [158] and [159]. The present paper *considerably* expands on [158] and [159] by providing the proofs of the results in [158] and [159]; additional theoretical results including a generalization in a stochastic setting; and additional informative discussions.

The remainder of this paper is organized as follows. The proposed distributed input and state coestimation architecture for fixed active and passive node roles subject to nonidentical active node modalities is presented in Section 9.2 with its system-theoretical stability analysis, where generalizations to the case of time-varying active and passive node roles is given in Section 9.3. For addressing practical situations when the process and the node observations both include noise, a stochastic extension is also presented in Section 9.4. Furthermore, the aforementioned illustrative numerical examples are given in Section 9.5 and concluding remarks are summarized in Section 9.6. Finally, for the *notation* used in this paper, we refer to the appendix.

9.2 Distributed Coestimation: Fixed Active and Passive Node Roles

9.2.1 Problem Setup and Proposed Algorithm

Consider a process[16] with the dynamics given by

$$\dot{x}(t) = Ax(t) + Bw(t), \quad x(0) = x_0, \tag{9.1}$$

where $x(t) \in \mathbb{R}^n$ is a process internal state vector and $w(t) \in \mathbb{R}^p$ is an input to this process. Here, we consider that $x(t)$ is not measurable. We also consider that $w(t)$ is an unknown but bounded signal with a bounded time rate of change. In addition, $A \in \mathbb{R}^{n\times n}$ is a Hurwitz system matrix and $B \in \mathbb{R}^{n\times p}$ is a system input matrix.

In this paper, a sensor network with N nodes is considered, where nodes exchange local measurements with their neighbors under an undirected and connected graph $\mathcal{G}$. Following the terminology from [70, 71, 73, 75], a node i, $i = 1,\ldots,N$, is called an *active node* when it is subject to the observation of the

[16] We follow here the problem setup introduced in [157].

process (9.1) given by

$$y_i(t) = C_i x(t). \tag{9.2}$$

Here, $y_i(t) \in \mathbb{R}^m$ and $C_i \in \mathbb{R}^{m\times n}$ respectively stand for a measurable process output and the system output matrix for an active node i, $i = 1,\ldots,N$. Moreover, a node i, $i = 1,\ldots,N$, is called a *passive node* when it has no observation from the process given by (9.1). For the results of this section, the active and passive roles of each node are considered to be fixed. Yet, recall from (9.2) that each active node can have nonidentical sensing modalities. We also practically consider for the well-posedness that each active node has complementary properties distributed over the sensor network to guarantee collective observability[17] while the pairs (A,C_i), $i = 1,\ldots,N$, may not be locally observable.

Based on the setup given above, we propose a distributed coestimation algorithm for sensor networks to estimate the unmeasurable state $x(t)$ and the unknown input $w(t)$ of the process (9.1). Here, we note that A is assumed to be Hurwitz to allow the employment of the passive nodes in the sensor network and this assumption does not result from the distributed coestimation approach proposed below.

Consider now a new input and state coestimation architecture for each node i, $i = 1,\ldots,N$, given by

$$\begin{aligned}
\dot{\hat{x}}_i(t) &= A\hat{x}_i(t) + B\hat{w}_i(t) + g_i L_i\big(y_i(t) - C_i\hat{x}_i(t)\big) - \alpha M_i \sum_{j=1}^{N} a_{ij}\big(\hat{x}_i(t) - \hat{x}_j(t)\big) \\
&\quad + \alpha S_i \sum_{j=1}^{N} a_{ij}\big(\hat{w}_i(t) - \hat{w}_j(t)\big), \quad \hat{x}_i(0) = \hat{x}_{i0}, \qquad (9.3) \\
\dot{\hat{w}}_i(t) &= g_i J_i\big(y_i(t) - C_i\hat{x}_i(t)\big) - \sigma_i K_i\hat{w}_i(t) + \alpha T_i \sum_{j=1}^{N} a_{ij}\big(\hat{x}_i(t) - \hat{x}_j(t)\big) \\
&\quad - \alpha N_i \sum_{j=1}^{N} a_{ij}\big(\hat{w}_i(t) - \hat{w}_j(t)\big), \quad \hat{w}_i(0) = \hat{w}_{i0}, \qquad (9.4)
\end{aligned}$$

where $\hat{x}_i(t) \in \mathbb{R}^n$ is the local estimate of $x(t)$ and $\hat{w}_i(t) \in \mathbb{R}^p$ is the local input estimate of $w(t)$. In addition, $L_i \in \mathbb{R}^{n\times m}$, $J_i \in \mathbb{R}^{p\times m}$, and $K_i \in \mathbb{R}^{p\times p}$ are the design gain matrices and $\alpha \in \mathbb{R}_+$ and $\sigma_i \in \mathbb{R}_+$ are the design coefficients. Finally, $M_i \in \mathbb{R}^{n\times n}$, $S_i \in \mathbb{R}^{n\times p}$, $T_i \in \mathbb{R}^{p\times n}$, and $N_i \in \mathbb{R}^{p\times p}$ are also the additional design gain matrices. Here, $g_i = 1$ when the node i is active and $g_i = 0$ when the node i is passive. Since the local state and input information exchange terms (i.e., the coupling terms "$\hat{x}_i(t) - \hat{x}_j(t)$" and "$\hat{w}_i(t) - \hat{w}_j(t)$") appear

[17] Collective observability is defined as the pair (A,C) is observable, where $C = [C_1^{\mathrm{T}}, C_2^{\mathrm{T}}, \ldots, C_N^{\mathrm{T}}]^{\mathrm{T}}$ (see, for example, [1–3]).

both in the state and input updates given by (9.3) and (9.4), the word "*coestimation*" is used to indicate the proposed distributed algorithm.

9.2.2 Analysis of Proposed Algorithm

For the main result of this section, first define

$$\tilde{x}_i(t) \triangleq x(t) - \hat{x}_i(t) \in \mathbb{R}^n, \tag{9.5}$$

$$\tilde{w}_i(t) \triangleq \hat{w}_i(t) - w(t) \in \mathbb{R}^p. \tag{9.6}$$

Now, the time derivative of (9.5) can be written as

$$\begin{aligned}\dot{\tilde{x}}_i(t) &= Ax(t) + Bw(t) - A\hat{x}_i(t) - B\hat{w}_i(t) - g_i L_i\big(y_i(t) - C_i\hat{x}_i(t)\big) \\ &\quad + \alpha M_i \sum_{j=1}^{N} a_{ij}\big(\hat{x}_i(t) - \hat{x}_j(t)\big) - \alpha S_i \sum_{j=1}^{N} a_{ij}\big(\hat{w}_i(t) - \hat{w}_j(t)\big) \\ &= (A - g_i L_i C_i)\tilde{x}_i(t) - B\tilde{w}_i(t) - \alpha M_i \sum_{j=1}^{N} \mathcal{L}_{ij}\tilde{x}_j(t) - \alpha S_i \sum_{j=1}^{N} \mathcal{L}_{ij}\tilde{w}_j(t).\end{aligned} \tag{9.7}$$

In (9.7), $\mathcal{L}_{ij}$ is the entry of the Laplacian matrix on the i-th row and j-th column. In addition, the time derivative of (9.6) can be written as

$$\begin{aligned}\dot{\tilde{w}}(t) &= g_i J_i C_i \tilde{x}_i(t) - \sigma_i K_i\big(\tilde{w}_i(t) + w(t)\big) - \dot{w}(t) - \alpha T_i \sum_{j=1}^{N} a_{ij}\big(\tilde{x}_i(t) - \tilde{x}_j(t)\big) \\ &\quad - \alpha N_i \sum_{j=1}^{N} a_{ij}\big(\tilde{w}_i(t) - \tilde{w}_j(t)\big) \\ &= g_i J_i C_i \tilde{x}_i(t) - \sigma_i K_i \tilde{w}_i(t) - \alpha T_i \sum_{j=1}^{N} \mathcal{L}_{ij}\tilde{x}_j(t) - \alpha N_i \sum_{j=1}^{N} \mathcal{L}_{ij}\tilde{w}_j(t) - \sigma_i K_i w(t) - \dot{w}(t).\end{aligned} \tag{9.8}$$

Next, let $z_i = [\tilde{x}_i^{\mathrm{T}}(t), \tilde{w}_i^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{n+p}$. Now, (9.7) and (9.8) can be compactly written as

$$\begin{aligned}\dot{z}_i(t) &= \underbrace{\begin{bmatrix} A - g_i L_i C_i & -B \\ g_i J_i C_i & -\sigma_i K_i \end{bmatrix}}_{\bar{A}_i} z_i(t) - \alpha \sum_{j=1}^{N} \mathcal{L}_{ij} \underbrace{\begin{bmatrix} M_i & S_i \\ T_i & N_i \end{bmatrix}}_{H_i} z_j(t) + \underbrace{\begin{bmatrix} 0 \\ -\sigma_i K_i w(t) - \dot{w}(t) \end{bmatrix}}_{\phi_i(t)} \\ &= \bar{A}_i z_i(t) - \alpha \sum_{j=1}^{N} \mathcal{L}_{ij} H_i z_j(t) + \phi_i(t).\end{aligned} \tag{9.9}$$

Here, we note that the local design terms L_i, J_i, K_i, and σ_i can be always chosen to ensure $\bar{A}_i$ being Hurwitz[18]. Therefore, $\bar{A}_i$ is implicitly considered to be Hurwitz for the following analysis (this is the *local sufficient stability* condition). We also note that for any given positive-definite matrix $Q_i \in \mathbb{R}^{(n+p)\times(n+p)}$, there exists a unique positive-definite matrix $P_i \in \mathbb{R}^{(n+p)\times(n+p)}$ satisfying

$$\bar{A}_i^{\mathrm{T}} P_i + P_i \bar{A}_i + Q_i = 0. \tag{9.10}$$

Now, let the aggregated vector be given by $z(t) \triangleq [z_1^{\mathrm{T}}(t), z_2^{\mathrm{T}}(t), \ldots, z_N^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{(n+p)N}$. To this end, (9.9) can be further written as

$$\begin{aligned}
\dot{z}(t) &= \underbrace{\begin{bmatrix} \bar{A}_1 & & 0 \\ & \ddots & \\ 0 & & \bar{A}_N \end{bmatrix}}_{\bar{A}} z(t) - \alpha \underbrace{\begin{bmatrix} H_1 & & 0 \\ & \ddots & \\ 0 & & H_N \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} \mathcal{L}_{11}\mathrm{I}_{n+p} & \mathcal{L}_{12}\mathrm{I}_{n+p} & \ldots & \mathcal{L}_{1N}\mathrm{I}_{n+p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}_{N1}\mathrm{I}_{n+p} & \mathcal{L}_{N2}\mathrm{I}_{n+p} & \ldots & \mathcal{L}_{NN}\mathrm{I}_{n+p} \end{bmatrix}}_{(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_{n+p})} z(t) + \underbrace{\begin{bmatrix} \phi_1(t) \\ \vdots \\ \phi_N(t) \end{bmatrix}}_{\phi(t)} \\
&= \bar{A}z(t) - \alpha H(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_{n+p}) z(t) + \phi(t).
\end{aligned} \tag{9.11}$$

In (9.11), $\mathcal{L}(\mathcal{G})$ is the Laplacian matrix. The following theorem presents the main result of this section.

Theorem 9.2.1. *Consider the process given by (9.1) and the distributed input and state estimation architecture given by (9.3) and (9.4). If the matrix H_i is selected as $H_i = P_i^{-1}$ and nodes exchange information using local measurements subject to an undirected and connected graph $\mathcal{G}$, then the error dynamics given by (9.11) is uniformly bounded.*

Proof. Consider the Lyapunov-like function candidate given by

$$V(z) = z^{\mathrm{T}} P z, \tag{9.12}$$

where $P = \mathrm{diag}([P_1, P_2, \ldots, P_N])$ is a positive-definite matrix. Note that $V(0) = 0$, and $V(z) > 0$ for all $z \neq 0$. Taking time-derivative of $V(z)$ along the trajectory of (9.11) yields

$$\begin{aligned}
\dot{V}\big(z(t)\big) &= z^{\mathrm{T}}(t)(\bar{A}^{\mathrm{T}}P + P\bar{A})z(t) - 2\alpha z^{\mathrm{T}}(t) PH(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_{n+p}) z(t) + 2z^{\mathrm{T}}(t) P\phi(t) \\
&= -z^{\mathrm{T}}(t) Q z(t) - 2\alpha z^{\mathrm{T}}(t) PH(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_{n+p}) z(t) + 2z^{\mathrm{T}}(t) P\phi(t),
\end{aligned} \tag{9.13}$$

[18] As mentioned earlier, A is considered to be Hurwitz because of the existence of passive nodes in the sensor network. Hence, this argument follows from the upper diagonal structure of $\bar{A}_i$ when, for example, $L_i = 0_{n\times m}$, $J_i = 0_{p\times m}$, and K_i is any positive definite matrix with $\sigma_i > 0$. For a desirable performance, however, different values for L_i, J_i and K_i should be judiciously selected such that $\bar{A}_i$ is Hurwitz (see Remark 9.2.2).

where $Q = \text{diag}([Q_1, Q_2, \ldots, Q_N])$ is also a positive-definite matrix. Since we choose M_i, S_i, T_i and N_i such that $H_i = \begin{bmatrix} M_i & S_i \\ T_i & N_i \end{bmatrix} = P_i^{-1}$ holds, we have $H = P^{-1}$ (i.e., $PH = \mathrm{I}_N \otimes \mathrm{I}_{n+p}$). Hence, (9.13) can be rewritten as

$$\begin{aligned} \dot{V}\big(z(t)\big) &= -z^{\mathrm{T}}(t)\big(Q + 2\alpha(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_{n+p})\big)z(t) + 2z^{\mathrm{T}}(t)P\phi(t) \\ &= -z^{\mathrm{T}}(t)\bar{Q}z(t) + 2z^{\mathrm{T}}(t)P\phi(t), \end{aligned} \tag{9.14}$$

where $\bar{Q} \triangleq Q + 2\alpha(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_{n+p})$. Since Q is a positive-definite matrix and the Laplacian matrix $\mathcal{L}(\mathcal{G})$ is a positive-semidefinite matrix, then $\bar{Q}$ is a positive-definite matrix (Proposition 8.1.2, [117]). In addition, since $\|w(t)\|_2 \leq \bar{w}$ and $\|\dot{w}(t)\|_2 \leq \bar{\dot{w}}$, then $\phi_i(t)$ is bounded, i.e., $||\phi_i(t)||_2 \leq \bar{\phi}_i$ with

$$\bar{\phi}_i \triangleq \sigma_i \bar{w} \|K_i\|_2 + \bar{\dot{w}}. \tag{9.15}$$

As a result, $\|\phi(t)\|_2 \leq \bar{\phi}$ holds with

$$\bar{\phi} \triangleq \sqrt{\phi_1^2 + \phi_2^2 + \ldots + \phi_N^2}. \tag{9.16}$$

From (9.14), we can now write

$$\begin{aligned} \dot{V}\big(z(t)\big) &\leq -\lambda_{\min}(\bar{Q})\|z(t)\|_2^2 + 2\|z(t)\|_2\|P\|_2\bar{\phi} \\ &= \|z(t)\|_2\big(2\|P\|_2\bar{\phi} - \lambda_{\min}(\bar{Q})\|z(t)\|_2\big). \end{aligned} \tag{9.17}$$

Finally, by letting $\mu \triangleq \frac{2\|P\|_2\bar{\phi}}{\lambda_{\min(\mathcal{Q})}}$ and $\Omega \triangleq \{z(t) : \|z(t)\|_2 \leq \mu\}$, it follows that $\dot{V}\big(z(t)\big) < 0$ outside the compact set Ω. Therefore, the error dynamics given by (9.11) is uniformly bounded [115]. ■

The following corollary is now immediate with regard to the performance of the proposed estimation approach.

Corollary 9.2.1. *Consider the process given by (9.1) and the distributed input and state estimation architecture given by (9.3) and (9.4). If the matrix H_i is selected as $H_i = P_i^{-1}$ and nodes exchange information*

using local measurement subject to an undirected and connected graph $\mathcal{G}$, then the bound

$$\|z(t)\|_2 \leq \sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}\mu, \tag{9.18}$$

holds for $t \geq T$.

Proof. In the proof of Theorem 1, we show that $V\big(z(t)\big)$ cannot grow outside the compact set Ω. Thus, from $\lambda_{\min}(P))\|z(t)\|_2^2 \leq V\big(z(t)\big) \leq \lambda_{\max}(P))\|z(t)\|_2^2$, we have $\lambda_{\min}(P))\|z(t)\|_2^2 \leq \lambda_{\max}(P))\mu^2$; hence the bound given by (9.18) is immediate. ■

Remark 9.2.1. *We now compare the new distributed input and state coestimation architecture given by (9.3) and (9.4) with its counterpart in [1]. For this purpose, recall the distributed input and state estimation law of [1]*

$$\dot{\hat{x}}_i(t) = (A-\gamma P_i^{-1})\hat{x}_i(t) + B\hat{w}_i(t) + g_i L_i\big(y_i(t) - C_i\hat{x}_i(t)\big) - \alpha P_i^{-1}\sum_{i\sim j}(\hat{x}_i(t)-\hat{x}_j(t)),\ \ \hat{x}_i(0) = \hat{x}_{i0}, \tag{9.19}$$

$$\dot{\hat{w}}_i(t) = g_i J_i(y_i(t) - C_i\hat{x}_i(t)) - (\sigma_i K_i + \gamma \mathrm{I}_p)\hat{w}_i(t) - \alpha\sum_{i\sim j}(\hat{w}_i(t)-\hat{w}_j(t)),\ \ \hat{w}_i(0) = \hat{w}_{i0}, \tag{9.20}$$

where $P_i \in \mathbb{R}^{(n+p)\times(n+p)}$ is a positive-definite gain matrix satisfying the linear matrix inequality

$$R_i = \begin{bmatrix} \bar{A}_i^{\mathrm{T}}P_i + P_i\bar{A}_i & -P_iB + g_iC_i^{\mathrm{T}}K_i^{\mathrm{T}} \\ -B^{\mathrm{T}}P_i + g_iJ_iC_i & -2\sigma_iK_i \end{bmatrix} \leq 0, \tag{9.21}$$

with $\bar{A}_i \triangleq A - g_iL_iC_i$. As discussed in [1], the terms "$-\gamma P_i^{-1}\hat{x}_i(t)$" and "$-(\sigma_iK_i+\gamma \mathrm{I}_p)\hat{w}_i(t)$" appearing respectively in (9.19) and (9.20) are referred as leakage terms. In particular, if "γP_i^{-1}" and "$\sigma_iK_i + \gamma \mathrm{I}_p$" in these terms are not small, they can lead to an unsatisfactory performance. Furthermore, "σ_iK_i" also appears in the linear matrix inequality given by (9.21). However, we may not be able to select this term as small while simultaneously satisfying (9.21), either due to the magnitude of the term "$-P_iB + g_iC_i^{\mathrm{T}}K_i$" being not small or a computational conservatism. To summarize, (9.19) and (9.20) of [1] may not always yield to an acceptable performance (see also Remark 9.3.1).

In contrast to (9.19) and (9.20) of [1] (Remark 9.2.1), the new input and state coestimation architecture given by (9.3) and (9.4) has only one leakage term "$-\sigma_iK_i\hat{w}_i(t)$" that appears in the input update (9.4). Additionally, as discussed in the next remark, "σ_iK_i" can be made judiciously small, and hence, the

proposed approach in this section has the potential to achieve a better overall estimation performance as compared with the results in [1].

Remark 9.2.2. *The ultimate bound given by (9.18) can be used as a design metric in the sense that design parameters can be chosen to make (9.18) small. For instance, small values for σ_i and K_i can be selected such that (9.15) and (9.16) are small, where they appear on the ultimate bound (9.18) through the term μ.*

9.3 Distributed Coestimation: Time-varying Active and Passive Node Roles

9.3.1 Problem Setup and Proposed Algorithm

In this section, we generalize the result of Section 9.2 to the case when the active and passive roles of each node vary in time. For this purpose, we again consider the process of interest given by (9.1). We also have a sensor network with N nodes exchanging information among each other using their local measurements according to an undirected and connected graph $\mathcal{G}$. In addition, a node is called active for some time instant when it is subject to the observation of the process given by (9.2) at that time instant. Likewise, a node is called passive when it has no observation at that time instant. Motivated by the case in (Figure 2d, [152]) and without loss of much practical generality, a node is considered to have ability to smoothly change back and forth between active and passive node roles, where the role change is captured by the smooth function $g_i(t) \in [0,1]$. Once again, we consider collective observability for well-posedness. In what follows, we first present the proposed distributed input and state coestimation algorithm below and then its analysis in Section 9.3.2.

In particular, consider now the proposed input and state coestimation architecture for each node i, $i = 1,\ldots,N$, given by

$$\begin{aligned}
\dot{\hat{x}}_i(t) &= A\hat{x}_i(t) + B\hat{w}_i(t) + g_i(t)L_i\big(y_i(t) - C_i\hat{x}_i(t)\big) - \alpha M_i \sum_{j=1}^{N} a_{ij}\big(\hat{x}_i(t) - \hat{x}_j(t)\big) \\
&\quad + \alpha S_i \sum_{j=1}^{N} a_{ij}\big(\hat{w}_i(t) - \hat{w}_j(t)\big), \quad \hat{x}_i(0) = \hat{x}_{i0}, \qquad (9.22) \\
\dot{\hat{w}}_i(t) &= g_i(t)J_i\big(y_i(t) - C_i\hat{x}_i(t)\big) - \sigma_i K_i \hat{w}_i(t) + \alpha T_i \sum_{j=1}^{N} a_{ij}\big(\hat{x}_i(t) - \hat{x}_j(t)\big) \\
&\quad - \alpha N_i \sum_{j=1}^{N} a_{ij}\big(\hat{w}_i(t) - \hat{w}_j(t)\big), \quad \hat{w}_i(0) = \hat{w}_{i0}, \qquad (9.23)
\end{aligned}$$

where $\hat{x}_i(t) \in \mathbb{R}^n$ is the local estimate of $x(t)$, and $\hat{w}_i(t) \in \mathbb{R}^p$ is the local input estimate of $w(t)$. In addition, $L_i \in \mathbb{R}^{n\times p}$, $J_i \in \mathbb{R}^{p\times m}$ and $K_i \in \mathbb{R}^{p\times p}$ are design gain matrices and $\alpha \in \mathbb{R}_+$ and $\sigma_i \in \mathbb{R}_+$ are the design coefficients. Finally, $M_i \in \mathbb{R}^{n\times n}$, $S_i \in \mathbb{R}^{n\times p}$, $T_i \in \mathbb{R}^{p\times n}$, and $N_i \in \mathbb{R}^{p\times p}$ are also the additional design gain matrices[19]. As discussed above, the smooth function $g_i(t) \in [0,1]$, $i = 1,\ldots,N$ indicates whether a node is active or passive at time t.

9.3.2 Analysis of Proposed Algorithm

For the main result of this section, first define (9.5) and (9.6). Now, the time derivative of (9.5) can be written as

$$\begin{aligned}\dot{\tilde{x}}_i(t) &= Ax(t)+Bw(t)-A\hat{x}_i(t)-B\hat{w}_i(t)-g_i(t)L_i\big(y_i(t)-C_i\hat{x}_i(t)\big)+\alpha M_i\sum_{j=1}^{N}a_{ij}\big(\hat{x}_i(t)-\hat{x}_j(t)\big)\\ &\quad -\alpha S_i\sum_{j=1}^{N}a_{ij}\big(\hat{w}_i(t)-\hat{w}_j(t)\big)\\ &= (A-g_i(t)L_iC_i)\tilde{x}_i(t)-B\tilde{w}_i(t)-\alpha M_i\sum_{j=1}^{N}\mathcal{L}_{ij}\tilde{x}_j(t)-\alpha S_i\sum_{j=1}^{N}\mathcal{L}_{ij}\tilde{w}_j(t). \end{aligned} \tag{9.24}$$

In (9.24), $\mathcal{L}_{ij}$ is the entry of the Laplacian matrix on the i-th row and j-th column. In addition, the time derivative of (9.6) can be written as

$$\begin{aligned}\dot{\tilde{w}}(t) &= g_i(t)J_iC_i\big(x_i(t)-\hat{x}_i(t)\big)-\sigma_iK_i\big(\tilde{w}_i(t)+w(t)\big)+\alpha T_i\sum_{j=1}^{N}a_{ij}\big(x(t)-\tilde{x}_i(t)-x(t)+\tilde{x}_j(t)\big)\\ &\quad -\alpha N_i\sum_{j=1}^{N}a_{ij}\big(\tilde{w}_i(t)+w(t)-\tilde{w}_j(t)-w(t)\big)-\dot{w}(t)\\ &= g_i(t)J_iC_i\tilde{x}_i(t)-\sigma_iK_i\tilde{w}_i(t)-\alpha T_i\sum_{j=1}^{N}\mathcal{L}_{ij}\tilde{x}_j(t)-\alpha N_i\sum_{j=1}^{N}\mathcal{L}_{ij}\tilde{w}_j(t)-\sigma_iK_iw(t)-\dot{w}(t). \end{aligned} \tag{9.25}$$

Next, let $z_i(t) \triangleq [\tilde{x}_i^{\mathrm{T}}(t), \tilde{w}_i^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{n+p}$. Now, (9.24) and (9.25) can be compactly written as

$$\begin{aligned}\dot{z}_i(t) &= \underbrace{\begin{bmatrix} A-g_i(t)L_iC_i & -B \\ g_i(t)J_iC_i & -\sigma_iK_i \end{bmatrix}}_{\bar{A}_i\left(g_i(t)\right)} z_i(t)-\alpha\sum_{j=1}^{N}\mathcal{L}_{ij}\underbrace{\begin{bmatrix} M_i & S_i \\ T_i & N_i \end{bmatrix}}_{H_i} z_j(t)+\underbrace{\begin{bmatrix} 0 \\ -\sigma_iK_iw(t)-\dot{w}(t) \end{bmatrix}}_{\phi_i(t)}\\ &= \bar{A}_i\big(g_i(t)\big)z_i(t)-\alpha\sum_{j=1}^{N}\mathcal{L}_{ij}H_iz_j(t)+\phi_i(t). \end{aligned} \tag{9.26}$$

[19] We refer to the steps *i)* and *ii)* given later in Section 9.3.2 on the selection of design gain matrices.

Here, the matrix $\bar{A}_i\big(g_i(t)\big)$ can be rewritten in the form

$$\begin{aligned}\bar{A}_i\big(g_i(t)\big) &= \underbrace{\begin{bmatrix} A & -B \\ 0 & -\sigma_i K_i \end{bmatrix}}_{\bar{A}_{i,0}} + g_i(t) \underbrace{\begin{bmatrix} -L_i C_i & 0 \\ J_i C_i & 0 \end{bmatrix}}_{\tilde{A}_i}, \\ &= \bar{A}_{i,0} + g_i(t)\tilde{A}_i, \end{aligned} \tag{9.27}$$

where $g_i(t) \in [0,1]$. Note that $\bar{A}_{i,0}$ and $\bar{A}_{i,1}$ are the matrices corresponding to $\bar{A}_i\big(g_i(t)\big)$ at $g_i(t) = 0$ and $g_i(t) = 1$, respectively. Hence, $\tilde{A}_i = \bar{A}_{i,1} - \bar{A}_{i,0}$. The following lemma is needed for the stability analysis (Theorem 9.3.1) of the proposed distributed input and state coestimation algorithm.

Lemma 9.3.1. *If there exists a common positive-definite matrix $P_i \in \mathbb{R}^{(n+p)\times(n+p)}$ for node i, $i = 1,\ldots,N$, satisfying*

$$\bar{A}_{i,0}^{\mathrm{T}} P_i + P_i \bar{A}_{i,0} \leq -\varepsilon \mathrm{I}_{n+p}, \tag{9.28}$$

$$\bar{A}_{i,1}^{\mathrm{T}} P_i + P_i \bar{A}_{i,1} \leq -\varepsilon \mathrm{I}_{n+p}, \tag{9.29}$$

then the inequality given by

$$\bar{A}_i\big(g_i(t)\big)^{\mathrm{T}} P_i + P_i \bar{A}_i\big(g_i(t)\big) \leq -\varepsilon \mathrm{I}_{n+p}, \tag{9.30}$$

holds for all $g_i(t) \in [0,1]$, where $\varepsilon \in \mathbb{R}_+$.

Proof. First, note that the inequality given by (9.30) implies

$$\mathcal{S} \triangleq \xi^{\mathrm{T}} \big[\bar{A}_i\big(g_i(t)\big)^{\mathrm{T}} P_i + P_i \bar{A}_i\big(g_i(t)\big) + \varepsilon \mathrm{I}_{n+p}\big]\xi \leq 0, \tag{9.31}$$

for any arbitrary nonzero vector ξ. Next, using (9.28) and (9.29), one can write

$$\begin{aligned} \mathcal{S} &= \xi^{\mathrm{T}} \big[\big(\bar{A}_{i,0} + g_i(t)\tilde{A}_i\big)^{\mathrm{T}} P_i + P_i\big(\bar{A}_{i,0} + g_i(t)\tilde{A}_i\big) + \varepsilon \mathrm{I}_{n+p}\big]\xi \\ &= \xi^{\mathrm{T}} \Big[\Big(\bar{A}_{i,0} + g_i(t)(\bar{A}_{i,1} - \bar{A}_{i,0})\Big)^{\mathrm{T}} P_i + P_i\Big(\bar{A}_{i,0} + g_i(t)(\bar{A}_{i,1} - \bar{A}_{i,0})\Big) + \varepsilon \mathrm{I}_{n+p}\Big]\xi \\ &= \xi^{\mathrm{T}} \Big[\Big(\big(1 - g_i(t)\big)\bar{A}_{i,0} + g_i(t)\bar{A}_{i,1}\Big)^{\mathrm{T}} P_i + P_i\Big(\big(1 - g_i(t)\big)\bar{A}_{i,0} + g_i(t)\bar{A}_{i,1}\Big) + \varepsilon \mathrm{I}_{n+p}\Big]\xi \\ &= \xi^{\mathrm{T}} \Big[\big(1 - g_i(t)\big)\big[\bar{A}_{i,0}^{\mathrm{T}} P_i + P_i \bar{A}_{i,0}\big] + g_i(t)\big[\bar{A}_{i,1}^{\mathrm{T}} P_i + P_i \bar{A}_{i,1}\big] + \varepsilon \mathrm{I}_{n+p}\Big]\xi \\ &\leq \xi^{\mathrm{T}} \Big[\big(1 - g_i(t)\big)\big(-\varepsilon \mathrm{I}_{n+p}\big) + g_i(t)\big(-\varepsilon \mathrm{I}_{n+p}\big) + \varepsilon \mathrm{I}_{n+p}\Big]\xi \\ &= 0. \end{aligned} \tag{9.32}$$

Thus, the proof is now complete. ■

Now, let the aggregated vector be given by $z(t) \triangleq [z_1^{\mathrm{T}}(t), z_2^{\mathrm{T}}(t), \ldots, z_N^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{(n+p)N}$. To this end, (9.26) can be written as

$$\begin{aligned}\dot{z}(t) &= \underbrace{\begin{bmatrix} \bar{A}_1(g_1(t)) & & 0 \\ & \ddots & \\ 0 & & \bar{A}_N(g_N(t)) \end{bmatrix}}_{\bar{A}(g(t))} z(t) - \alpha \underbrace{\begin{bmatrix} H_1 & & 0 \\ & \ddots & \\ 0 & & H_N \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} \mathcal{L}_{11}\mathrm{I}_{n+p} & \mathcal{L}_{12}\mathrm{I}_{n+p} & \ldots & \mathcal{L}_{1N}\mathrm{I}_{n+p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}_{N1}\mathrm{I}_{n+p} & \mathcal{L}_{N2}\mathrm{I}_{n+p} & \ldots & \mathcal{L}_{NN}\mathrm{I}_{n+p} \end{bmatrix}}_{(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_{n+p})} z(t) + \underbrace{\begin{bmatrix} \phi_1(t) \\ \vdots \\ \phi_N(t) \end{bmatrix}}_{\phi(t)} \\ &= \bar{A}\big(g(t)\big)z(t) - \alpha H(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_{n+p})z(t) + \phi(t). \end{aligned} \tag{9.33}$$

In (9.33), where $\mathcal{L}(\mathcal{G})$ is the Laplacian matrix and $g(t) = [g_1(t), \ldots, g_N(t)]^{\mathrm{T}}$. In what follows, for each node i, $i = 1, \ldots, N$, we:

i) Solve the linear matrix inequalities given by (9.28) and (9.29) for a common positive-definite matrix P_i (these are the *local sufficient stability* conditions).

ii) Obtain the design gain matrices M_i, S_i, T_i, and N_i from the matrix equality given by

$$H_i = \begin{bmatrix} M_i & S_i \\ T_i & N_i \end{bmatrix} = P_i^{-1}. \tag{9.34}$$

Notice from (9.34) that $H = P^{-1}$ (i.e., $PH = \mathrm{I}_N \otimes \mathrm{I}_{n+p} = \mathrm{I}_{N(n+p)}$), where $P \triangleq \mathrm{diag}([P_1, P_2, \ldots, P_N])$. We are now ready to state the main result of this section.

Theorem 9.3.1. *Consider the process given by (9.1) and the distributed input and state coestimation architecture given by (9.22) and (9.23). If there exists a common positive-definite matrix P_i for each node i, $i = 1, \ldots, N$, satisfying (9.28) and (9.29), one selects H_i according to (9.34), and nodes exchange information according to an undirected and connected graph $\mathcal{G}$, then the error dynamics given by (9.33) is uniformly bounded.*

Proof. Consider the Lyapunov-like function candidate given by (9.12), where $P = \mathrm{diag}([P_1, P_2, \ldots, P_N])$ is a positive-definite matrix with each P_i obtained through solving the linear matrix inequalities given by (9.28) and (9.29). Note that $V(0) = 0$, and $V(z) > 0$ for all $z \neq 0$. Taking time-derivative

of $V(z)$ along the trajectory of (9.33) and using Lemma 9.3.1 yields

$$\begin{aligned}\dot{V}\big(z(t)\big) &= z^{\mathrm{T}}(t)\big(\bar{A}^{\mathrm{T}}\big(g(t)\big)P+P\bar{A}\big(g(t)\big)\big)z(t)-2\alpha z^{\mathrm{T}}(t)PH(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_{n+p})z(t)+2z^{\mathrm{T}}(t)P\phi(t)\\ &\leq -\varepsilon z^{\mathrm{T}}(t)z(t)-2\alpha z^{\mathrm{T}}(t)(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_{n+p})z(t)+2z^{\mathrm{T}}(t)P\phi(t)\\ &= -z^{\mathrm{T}}(t)\bar{Q}z(t)+2z^{\mathrm{T}}(t)P\phi(t), \qquad (9.35)\end{aligned}$$

where $\bar{Q}=\varepsilon \mathrm{I}_{N(n+p)}+2\alpha(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_{n+p})$. Since $\varepsilon \mathrm{I}_{N(n+p)}$ is a positive-definite matrix and the Laplacian matrix $\mathcal{L}(\mathcal{G})$ is a positive-semidefinite matrix, $\bar{Q}$ is a positive-definite matrix. In addition, $\|\phi\|_2\leq\bar{\phi}$ with $\bar{\phi}$ defined by (9.16). Therefore, an upper bound to (9.35) can be found as $\dot{V}\big(z(t)\big)\leq -\lambda_{\min}(\bar{Q})\|z(t)\|_2^2+2\|z(t)\|_2\|P\|_2\bar{\phi}=\|z(t)\|_2\big(2\|P\|_2\bar{\phi}-\lambda_{\min}(\bar{Q})\|z(t)\|_2\big)$. Letting $\mu\triangleq\frac{2\|P\|_2\bar{\phi}}{\lambda_{\min}(\bar{Q})}$ and $\Omega\triangleq\{z(t):\|z(t)\|_2\leq\mu\}$, it follows that $\dot{V}\big(z(t)\big)<0$ outside the compact set Ω, and therefore, the error dynamics given by (9.33) is uniformly bounded [115]. ■

Note that Theorem 9.3.1 establishes the stability of the proposed distributed input and state coestimation architecture given by (9.22) and (9.23) in terms of uniform boundedness under *local sufficient stability conditions* (9.28) and (9.29) for each node. The following corollary is now immediate on the performance of the proposed architecture.

Corollary 9.3.1. *Consider the process given by (9.1) and the distributed input and state coestimation architecture given by (9.22) and (9.23). If there exists a common positive-definite matrix P_i for each node i, $i=1,\ldots,N$, satisfying (9.28) and (9.29), one selects H_i according to (9.34), and nodes exchange information according to an undirected and connected graph $\mathcal{G}$, then the bound (9.18) holds for $t\geq T$.*

Proof. The result follows from the proof of Corollary 9.2.1. ■

Note that the discussion given in Remark 9.2.2 also holds for the results of this section. That is, since the ultimate bound in Corollary 9.3.1 depends on the design parameters of (9.22) and (9.23), this bound can be used as design metric in the sense that the design parameters can be judiciously selected to make (9.18) small.

Remark 9.3.1. *Once again, we compare the proposed distributed input and state coestimation architecture of this section given by (9.22) and (9.23) with its counterpart in [1]. In particular, the related distributed*

estimation law of [1] that allows time-varying active and passive roles have the form

$$\dot{\hat{x}}_i(t) = (A-\gamma P_i^{-1})\hat{x}_i(t)+B\hat{w}_i(t)+g_i(t)L_i\big(y_i(t)-C_i\hat{x}_i(t)\big)-\alpha P_i^{-1}\sum_{i\sim j}(\hat{x}_i(t)-\hat{x}_j(t)),\quad \hat{x}_i(0)=\hat{x}_{i0}, \tag{9.36}$$

$$\dot{\hat{w}}_i(t) = g_i(t)J_i(y_i(t)-C_i\hat{x}_i(t))-(\sigma_i K_i+\gamma \mathrm{I}_p)\hat{w}_i(t)-\alpha\sum_{i\sim j}(\hat{w}_i(t)-\hat{w}_j(t)),\quad \hat{w}_i(0)=\hat{w}_{i0}, \tag{9.37}$$

with $P_i \in \mathbb{R}^{(n+p)\times(n+p)}$ is a positive-definite gain matrix satisfying

$$R_{i1} \triangleq \begin{bmatrix} A^{\mathrm{T}}P_i+P_iA & -P_iB \\ -B^{\mathrm{T}}P_i & -2\sigma_iK_i \end{bmatrix} \leq 0, \tag{9.38}$$

$$R_{i2} \triangleq \begin{bmatrix} (A-L_iC_i)^{\mathrm{T}}P_i+P_i(A-L_iC_i) & -P_iB+C_i^{\mathrm{T}}J_i^{\mathrm{T}} \\ -B^{\mathrm{T}}P_i+J_iC_i & -2\sigma_iK_i \end{bmatrix} \leq 0. \tag{9.39}$$

Similar to discussion in Remark 9.2.1, (9.36) and (9.37) respectively contain the leakage terms "$-(\sigma_iK_i+\gamma \mathrm{I}_p)\hat{w}_i(t)$" in input and state updates. In particular, if the gains "γP_i^{-1}" and "$\sigma_iK_i+\gamma \mathrm{I}_p$" are not small, then they can result in poor performance. In contrast, the distributed coestimation architecture with time-varying active and passive node roles proposed this section has only one leakage term "$-\sigma_iK_i\hat{w}_i(t)$" appearing in the input update (9.23). Moreover, the proposed architecture of this section adds the coupling terms "$\hat{x}_i(t)-\hat{x}_j(t)$" and "$\hat{w}_i(t)-\hat{w}_j(t)$" to both input and state updates. Finally, the structure of the linear matrix inequalities given by (9.28) and (9.29) is simpler than the ones in (9.38) and (9.39). For these reasons, the proposed coestimation architecture here can be easily (i.e., better) tuned for an overall performance as opposed to the approach in [1] (see also (Section 4.3, [1]) for further details on the tuning challenges with regard to (9.36) and (9.37)).

Remark 9.3.2. *The purpose of solving the linear matrix inequalities given by (9.28) and (9.29) is to find a common positive-definite solution P_i in order to select $H_i = P_i^{-1}$ according to (9.34). Note that the existence of such a common solution depends on many factors such as the characteristics of the system matrix A as well as the design gain matrices K_i, L_i, and J_i. To this end, the results in [160–164] (also see references therein) can be utilized on the existence of a common positive-definite solution to linear matrix inequalities. Following the results in [161], we can consider the matrix $\tilde{A}_{i,1}$ in (9.27) as a perturbation matrix and search for stability region, then design gain matrices accordingly and use convex programming tools to test the*

feasibility. In some special cases, where we can design $\bar{A}_{i,0}$ and $\bar{A}_{i,1}$ as commuting matrices or in companion form, the results of [162] and [163], respectively, can be useful.

9.4 Distributed Coestimation in a Stochastic Setting

9.4.1 Problem Setup and Proposed Algorithm

In this section, we consider a stochastic case and generalize the results in Section 9.2. Note that this is without loss of generality as the theoretical content of this section can be similarly applied to the results in Section 9.3. Specifically, consider a process of interest with the dynamics given by

$$\mathrm{d}x = \big(Ax(t) + Bw(t)\big)\mathrm{d}t + v(t)\mathrm{d}\nu(t), \quad x(0) = x_0, \tag{9.40}$$

where $x(t) \in \mathbb{R}^n$ is a process internal state vector and $w(t) \in \mathbb{R}^p$ is an input to this process. Here, we consider that $x(t)$ is not measurable. We also consider that $w(t)$ is unknown but a bounded signal with a bounded time rate of change. In addition, $v(t) \in \mathbb{R}^n$ is a bounded external noise intensity function (i.e., $\|v(t)\|_2 \leq v^*$) and $\nu(t)$ is a one dimensional Brownian motion defined on the probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with expectation $\mathbb{E}\{\mathrm{d}\nu(t)\} = 0$ and variance $\mathbb{D}\{\mathrm{d}\nu(t)\} = 1$. Furthermore, $A \in \mathbb{R}^{n\times n}$ is a Hurwitz system matrix and $B \in \mathbb{R}^{n\times p}$ is the system input matrix.

Consider a sensor network with N nodes exchanging information among each other using their local measurements through an undirected and connected graph $\mathcal{G}$. Following the terminology from previous sections, a node i, $i = 1,\ldots,N$, is called an *active node* when it is subject to the observation of the process (9.40) given by

$$y_i(t) = C_i x(t) + h_i(t)\frac{\mathrm{d}s_i(t)}{\mathrm{d}t}. \tag{9.41}$$

Here, $y_i(t) \in \mathbb{R}^m$ and $C_i \in \mathbb{R}^{m\times n}$ respectively stand for a measurable process output and the system output matrix for an active node i, $i = 1,\ldots,N$. In addition, $h_i(t) \in \mathbb{R}^m$ is a bounded external noise intensity function (i.e., $\|h_i(t)\|_2 \leq h^*$), $s_i(t)$ is a one dimensional Brownian motion (independent of the process noise $\nu(t)$ and its neighbors $s_j(t)$) defined on the probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with expectation $\mathbb{E}\{\mathrm{d}s_i(t)\} = 0$ and variance $\mathbb{D}\{\mathrm{d}s_i(t)\} = 1$. Moreover, a node i is called a *passive node* when it has no observation from the process (9.40). Recall from (9.41) that each node can have nonidentical sensing modalities. We, once again,

consider collective observability for well-posedness. Finally, we utilize here the distributed input and state architecture given by (9.3) and (9.4).

9.4.2 Analysis of Proposed Algorithm

To present our main results, we first define (9.5) and (9.6). Now, the stochastic differential of (9.5) is given by

$$\begin{aligned}
\mathrm{d}\tilde{x}_i(t) &= \Big(Ax(t)+Bw(t)-A\hat{x}_i(t)-B\hat{w}_i(t)-g_iL_i\big(y_i(t)-C_i\hat{x}_i(t)\big)+\alpha M_i\sum_{j=1}^{N}a_{ij}\big(\hat{x}_i(t)-\hat{x}_j(t)\big)\\
&\quad -\alpha S_i\sum_{j=1}^{N}a_{ij}\big(\hat{w}_i(t)-\hat{w}_j(t)\big)\Big)\mathrm{d}t+v(t)\mathrm{d}\nu(t)\\
&= \Big((A-g_iL_iC_i)\tilde{x}_i(t)-B\tilde{w}_i(t)-\alpha M_i\sum_{j=1}^{N}\mathcal{L}_{ij}\tilde{x}_j(t)-\alpha S_i\sum_{j=1}^{N}\mathcal{L}_{ij}\tilde{w}_j(t)\Big)\mathrm{d}t\\
&\quad +v(t)\mathrm{d}\nu(t)-g_iL_ih_i(t)\mathrm{d}s_i(t). \qquad (9.42)
\end{aligned}$$

In addition, the stochastic differential of (9.6) is given by

$$\begin{aligned}
\mathrm{d}\tilde{w}(t) &= \Big(g_iJ_iC_i\tilde{x}_i(t)-\sigma_iK_i\tilde{w}_i(t)-\alpha T_i\sum_{j=1}^{N}\mathcal{L}_{ij}\tilde{x}_j(t)-\alpha N_i\sum_{j=1}^{N}\mathcal{L}_{ij}\tilde{w}_j(t)-\sigma_iK_iw(t)-\dot{w}(t)\Big)\mathrm{d}t\\
&\quad +g_iJ_ih_i(t)\mathrm{d}s_i(t). \qquad (9.43)
\end{aligned}$$

Next, let $z_i(t)\triangleq[\tilde{x}_i^{\mathrm{T}}(t),\tilde{w}_i^{\mathrm{T}}(t)]^{\mathrm{T}}\in\mathbb{R}^{n+p}$. Now, (9.42) and (9.43) can be written in a compact form as

$$\begin{aligned}
\mathrm{d}z_i(t) &= \Bigg(\underbrace{\begin{bmatrix}A-g_iL_iC_i & -B\\ g_iJ_iC_i & -\sigma_iK_i\end{bmatrix}}_{\bar{A}_i}z_i(t)-\alpha\textstyle\sum_{j=1}^{N}\mathcal{L}_{ij}\underbrace{\begin{bmatrix}M_i & S_i\\ T_i & N_i\end{bmatrix}}_{H_i}z_j(t)+\underbrace{\begin{bmatrix}0\\ -\sigma_iK_iw(t)-\dot{w}(t)\end{bmatrix}}_{\phi_i(t)}\Bigg)\mathrm{d}t\\
&\quad +\underbrace{\begin{bmatrix}v(t)\\ 0\end{bmatrix}}_{\bar{v}}\mathrm{d}\nu(t)+\underbrace{\begin{bmatrix}-g_iL_ih_i(t)\\ g_iJ_ih_i(t)\end{bmatrix}}_{r_i(t)}\mathrm{d}s_i(t)\\
&= \Big(\bar{A}_iz_i(t)-\alpha\sum_{j=1}^{N}\mathcal{L}_{ij}H_iz_j(t)+\phi_i(t)\Big)\mathrm{d}t+\bar{v}(t)\mathrm{d}\nu(t)+r_i(t)\mathrm{d}s_i(t). \qquad (9.44)
\end{aligned}$$

Similar to the discussion in Section 9.2.2, one can always choose the local design terms L_i, J_i, K_i, and σ_i such that $\bar{A}_i$ is Hurwitz for each agent, and hence, there exists a unique positive-definite matrix $P_i\in\mathbb{R}^{(n+p)\times(n+p)}$ such that (9.10) holds for a given positive-definite matrix $Q_i\in\mathbb{R}^{(n+p)\times(n+p)}$.

Now, let the aggregated vector be given by $z(t) \triangleq [z_1^{\mathrm{T}}(t), z_2^{\mathrm{T}}(t), \ldots, z_N^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{(n+p)N}$. To this end, (9.44) can be further written as

$$\begin{aligned}
\mathrm{d}z(t) &= \left(\underbrace{\begin{bmatrix} \bar{A}_1 & & 0 \\ & \ddots & \\ 0 & & \bar{A}_N \end{bmatrix}}_{\bar{A}} z(t) - \alpha \underbrace{\begin{bmatrix} H_1 & & 0 \\ & \ddots & \\ 0 & & H_N \end{bmatrix}}_{H} \cdot \underbrace{\begin{bmatrix} \mathcal{L}_{11}\mathrm{I}_{n+p} & \mathcal{L}_{12}\mathrm{I}_{n+p} & \ldots & \mathcal{L}_{1N}\mathrm{I}_{n+p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}_{N1}\mathrm{I}_{n+p} & \mathcal{L}_{N2}\mathrm{I}_{n+p} & \ldots & \mathcal{L}_{NN}\mathrm{I}_{n+p} \end{bmatrix}}_{(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_{n+p})} z(t) + \phi(t) \right) \mathrm{d}t \\
&\quad + \big(\mathbf{1}_N \otimes \bar{v}(t)\big)\mathrm{d}v(t) + R(t)(\mathrm{d}s(t) \otimes \mathbf{1}_{n+p}) \\
&= \big(\bar{A}z(t) - \alpha H(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_{n+p})z(t) + \phi(t)\big)\mathrm{d}t + \big(\mathbf{1}_N \otimes \bar{v}(t)\big)\mathrm{d}v(t) + R(t)(\mathrm{d}s(t) \otimes \mathbf{1}_{n+p}), \qquad (9.45)
\end{aligned}$$

where $\phi(t) \triangleq [\phi_1^{\mathrm{T}}(t), \phi_2^{\mathrm{T}}(t), \ldots, \phi_N^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{(n+p)N}$, $R(t) \triangleq \mathrm{diag}([r_1^{\mathrm{T}}(t) \ \ldots \ r_N^{\mathrm{T}}(t)]^{\mathrm{T}}) = \mathrm{diag}(r(t))$, and $\mathrm{d}s(t) \triangleq \begin{bmatrix} \mathrm{d}s_1(t) & \ldots & \mathrm{d}s_N(t) \end{bmatrix}^{\mathrm{T}}$. The following proposition presents the main result of this section.

Proposition 9.4.1. *Consider the process given by (9.40) and the distributed input and state estimation architecture given by (9.3) and (9.4). If the matrix H_i is selected as $H_i = P_i^{-1}$ and nodes exchange information using local measurements subject to an undirected and connected graph $\mathcal{G}$, then $z(t)$ evolving according to the dynamics given by (9.45) satisfies the bound*

$$\mathbb{E}\{z^{\mathrm{T}}Pz\} \leq e^{-\kappa t} z^{\mathrm{T}}(0)Pz(0) + \kappa^{-1}\eta, \ \ \forall\, t \geq 0, \qquad (9.46)$$

where $\kappa \triangleq \frac{\lambda_{\min}(\bar{Q})}{2\lambda_{\max}(P)}$ and $\eta \triangleq \frac{2\bar{\phi}^2}{\lambda_{\min}(\bar{Q})}\|P\|_2^2 + \lambda_{\max}(P)\bar{\psi}^2$.

Proof. Applying Ito formula (see, for example, [165]) to the Lyapunov-like function candidate $V(z) = \sum_{i=1}^N z_i^{\mathrm{T}} P_i z_i$, one can write

$$\mathrm{d}V = LV\,\mathrm{d}t + 2\sum_{i=1}^N z_i^{\mathrm{T}} P_i(\bar{v}\mathrm{d}v(t) + r_i \mathrm{d}s_i(t)), \qquad (9.47)$$

where L is a linear differential operator associated with (9.44)

$$\begin{aligned}
LV &= \sum_{i=1}^N \left(z_i^{\mathrm{T}}(P_i\bar{A}_i + P_i\bar{A}_i^{\mathrm{T}})z_i - 2\alpha z_i^{\mathrm{T}} P_i \sum_{j=1}^N \mathcal{L}_{ij} H_i z_j + 2z_i^{\mathrm{T}} P_i \phi_i + \tfrac{1}{2}\,\mathrm{tr}\left(2P_i\bar{v}\bar{v}^{\mathrm{T}} + 2P_i r_i r_i^{\mathrm{T}}\right) \right) \\
&= \sum_{i=1}^N \left(-z_i^{\mathrm{T}} Q_i z_i - 2\alpha z_i^{\mathrm{T}} \sum_{j=1}^N \mathcal{L}_{ij} z_j + 2z_i^{\mathrm{T}} P_i \phi_i \ + \bar{v}^{\mathrm{T}} P_i \bar{v} + r_i^{\mathrm{T}} P_i r_i \right) \\
&= -\sum_{i=1}^N z_i^{\mathrm{T}} Q_i z_i - 2\alpha \sum_{i=1}^N \sum_{j=1}^N \mathcal{L}_{ij} z_i^{\mathrm{T}} z_j + 2\sum_{i=1}^N z_i^{\mathrm{T}} P_i \phi_i + \sum_{i=1}^N \psi_i^{\mathrm{T}} \big(\mathrm{I}_2 \otimes P_i\big)\psi_i, \qquad (9.48)
\end{aligned}$$

where $\psi_i \triangleq [\bar{v}^{\mathrm{T}}, r_i^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2(n+p)}$. In addition, (9.48) can be written in a compact form as

$$\begin{aligned} LV &= -z^{\mathrm{T}}Qz - 2\alpha z^{\mathrm{T}}(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_{n+p})z + 2z^{\mathrm{T}}P\phi + \psi^{\mathrm{T}}\bar{P}\psi \\ &= -z^{\mathrm{T}}\bar{Q}z + 2z^{\mathrm{T}}P\phi + \psi^{\mathrm{T}}\bar{P}\psi, \end{aligned} \tag{9.49}$$

where $\psi \triangleq [\psi_1^{\mathrm{T}}, \psi_2^{\mathrm{T}}, \ldots, \psi_N^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2(n+p)N}$; $P \triangleq \mathrm{diag}([P_1, P_2, \ldots, P_N]) \in \mathbb{R}^{N(n+p)\times N(n+p)}$, $\bar{P} \triangleq \mathrm{diag}([\mathrm{I}_2 \otimes P_1, \mathrm{I}_2 \otimes P_2, \ldots, \mathrm{I}_2 \otimes P_N]) \in \mathbb{R}^{2N(n+p)\times 2N(n+p)}$, and $Q \triangleq \mathrm{diag}([Q_1, Q_2, \ldots, Q_N]) \in \mathbb{R}^{N(n+p)\times N(n+p)}$ are positive-definite matrices; and $\bar{Q} \triangleq Q + 2\alpha\big(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_{n+p}\big)$. Since Q is a positive-definite matrix and the Laplacian matrix $\mathcal{L}(\mathcal{G})$ is a positive-semidefinite matrix, $\bar{Q}$ is a positive-definite matrix (Proposition 8.1.2, [117]). In addition, $\|\phi\|_2 \leq \bar{\phi}$ with $\bar{\phi}$ defined in (9.16). Furthermore, since $\|v\|_2 \leq v^*$ and $\|h_i\|_2 \leq h^*$, $\bar{v}$ and r_i are bounded (i.e., $\|\bar{v}_i\|_2 \leq \bar{v}^*$ and $\|r_i\|_2 \leq r^*$). As a result, ψ_i is bounded (i.e., $||\psi_i||_2 \leq \bar{\psi}_i$) with $\bar{\psi}_i \triangleq \sqrt{(\bar{v}^*)^2 + (r^*)^2}$, $i = 1, \ldots, N$. Therefore, we have $\|\psi\|_2 \leq \bar{\psi}$ with $\bar{\psi} \triangleq \sqrt{\psi_1^2 + \psi_2^2 + \ldots + \psi_N^2}$. By Young's inequality, one can now write $2\|z^{\mathrm{T}}\|_2\|P\phi\|_2 \leq \mu\|z\|_2^2 + \frac{1}{\mu}\|P\phi\|_2^2 \leq \mu\|z\|_2^2 + \frac{\bar{\phi}^2}{\mu}\|P\|_2^2$. Now, (9.49) becomes

$$LV \leq -(\lambda_{\min}(\bar{Q}) - \mu)\|z\|_2^2 + \frac{\bar{\phi}^2}{\mu}\|P\|_2^2 + \lambda_{\max}(P)\bar{\psi}^2. \tag{9.50}$$

From $V(z) = \sum_{i=1}^{N} z_i^{\mathrm{T}} P_i z_i = z^{\mathrm{T}} P z$, we also note that $\lambda_{\min}(P)\|z\|_2^2 \leq V(z) \leq \lambda_{\max}(P)\|z\|_2^2$. Thus, $\frac{V(z)}{\lambda_{\max}(P)} \leq \|z\|_2^2$. Using this fact and choosing $\mu = \frac{1}{2}(\lambda_{\min}(\bar{Q})$, we now have

$$\begin{aligned} LV &\leq \underbrace{-\frac{\lambda_{\min}(\bar{Q})}{2\lambda_{\max}(P)}}_{\kappa} V + \underbrace{\frac{2\bar{\phi}^2}{\lambda_{\min}(\bar{Q})}\|P\|_2^2 + \lambda_{\max}(P)\bar{\psi}^2}_{\eta} \\ &= -\kappa V + \eta. \end{aligned} \tag{9.51}$$

From Dynkin's formula (see, for example, [166, 167]), (9.46) is now immediate. ■

Remark 9.4.1. *Proposition 9.4.1 also holds for the distributed input and state coestimation architecture given by (9.22) and (9.23), where the active and passive node roles are varying over time and the corresponding parameters are chosen as outlined in Section 9.3.*

Remark 9.4.2. *Considering (9.46), one can write* $\lim_{t\to\infty} \mathbb{E}\{V(z(t))\} = \kappa^{-1}\eta$. *Now, using the definitions of* κ *and* η *given in (9.51) along with the fact* $\lambda_{\min}(P)\mathbb{E}\{\|z\|_2^2\} \leq \mathbb{E}\{V(z(t))\}$, *this expression implies*

$$\lim_{t\to\infty} \mathbb{E}\Big\{\|z\|_2\Big\} \leq \sqrt{\tfrac{\lambda_{\max}(P)}{\lambda_{\min}(P)}\left(\mu^2 + \tfrac{2\lambda_{\max}(P)\bar{\psi}^2}{\lambda_{\min}(\bar{Q})}\right)},\ \mu \triangleq \tfrac{2\|P\|_2\bar{\phi}}{\lambda_{\min}(\bar{Q})}. \tag{9.52}$$

When we compare (9.52) with the deterministic (worst-case) bound given by (9.18), it can be seen that the only additional term "$2\lambda_{\max}(P)\bar{\psi}^2/\big(\lambda_{\min}(P)\lambda_{\min}(\bar{Q})\big)$" in (9.52) results from the bound of the external noise intensity functions of the process and sensors' measurements $\bar{\psi}$.

Remark 9.4.3. *The error expression given by (9.45) can be rewritten as*

$$\mathrm{d}z(t) \;=\; \big(\mathcal{S}z(t)+\phi(t)\big)\mathrm{d}t+\big(\mathbf{1}_N\otimes\bar{v}(t)\big)\mathrm{d}\nu(t)+R(t)(\mathrm{d}s(t)\otimes\mathbf{1}_{n+p}), \tag{9.53}$$

where $\mathcal{S}\triangleq\bar{A}-\alpha H(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_{n+p})$. Referring now to the steps taken in the proof of Theorem 9.2.1, one can write

$$\begin{aligned}\mathcal{S}^{\mathrm{T}}P+P\mathcal{S} &= \big(\bar{A}-\alpha H(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_{n+p})\big)^{\mathrm{T}}P+P\big(\bar{A}-\alpha H(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_{n+p})\big)\\ &= \underbrace{\bar{A}^{\mathrm{T}}P+P\bar{A}}_{-Q}-2\alpha(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_{n+p})\\ &= -\big(Q+2\alpha(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_{n+p})\big)\triangleq-\bar{Q}.\end{aligned} \tag{9.54}$$

Hence, $\mathcal{S}$ is Hurwitz since both $\bar{Q}$ and P are positive-definite (Corollary 11.9.1, [117]). We note that a white noise process is the derivative of a Wiener process and its derivative is a generalized function (see, for example, (Observation 1.2.11, [165]) and [168]). Recall that $\nu(t)$ and $s_i(t)$ are Brownian motion processes or normalized Wiener processes; thus, we can define the zero mean white noise processes $\zeta(t)\triangleq\frac{\mathrm{d}\nu(t)}{\mathrm{d}t}\in\mathbb{R}$ and $\xi(t)\triangleq\frac{\mathrm{d}s(t)\otimes\mathbf{1}_{n+p}}{\mathrm{d}t}\in\mathbb{R}^{N(n+p)}$. As a result, (9.53) can be rewritten as

$$\dot{z}(t) \;=\; \mathcal{S}z(t)+\phi(t)+\big(\mathbf{1}_N\otimes\bar{v}(t)\big)\zeta(t)+R(t)\xi(t), \tag{9.55}$$

and its solution is given by

$$z(t) \;=\; e^{\mathcal{S}t}z(0)+\int_0^t e^{\mathcal{S}(t-s)}\phi(s)\mathrm{d}s+\int_0^t e^{\mathcal{S}(t-s)}\big(\mathbf{1}_N\otimes\bar{v}(s)\big)\zeta(s)\mathrm{d}s+\int_0^t e^{\mathcal{S}(t-s)}R(s)\xi(s)\mathrm{d}s. \tag{9.56}$$

Note that the error covariance of the system is given by $J(z(t))\triangleq\mathbb{E}\{z(t)z^{\mathrm{T}}(t)\}$ and its differential equation is

$$\dot{J}(t)=\mathbb{E}\left\{\frac{\mathrm{d}z(t)}{\mathrm{d}t}z^{\mathrm{T}}(t)\right\}+\mathbb{E}\left\{z(t)\frac{\mathrm{d}z^{\mathrm{T}}(t)}{\mathrm{d}t}\right\}. \tag{9.57}$$

The first term of (9.57) can be distributed as

$$\begin{aligned}\mathbb{E}\left\{\frac{\mathrm{d}z(t)}{\mathrm{d}t}z^{\mathrm{T}}(t)\right\} &= \mathbb{E}\left\{\left(\mathcal{S}z(t)+\phi(t)\ +\left(\mathbf{1}_N\otimes\bar{v}(t)\right)\zeta(t)+R(t)\xi(t)\right)z^{\mathrm{T}}(t)\right\}\\ &= \mathcal{S}\mathbb{E}\{z(t)z^{\mathrm{T}}(t)\}+\mathbb{E}\{\phi(t)z^{\mathrm{T}}(t)\}+\mathbb{E}\{\left(\mathbf{1}_N\otimes\bar{v}(t)\right)\zeta(t)z^{\mathrm{T}}(t)\}+\mathbb{E}\{R(t)\xi(t)z^{\mathrm{T}}(t)\}. \quad (9.58)\end{aligned}$$

We note here that $\left(\mathbf{1}_N\otimes\bar{v}(t)\right)\zeta(t)$ *represents the process noise,* $R(t)\xi(t)$ *represents the measurement noise,* $\phi(t)$ *represents the leakage term containing the process input and its time rate of change, and* $z(0)$ *is the initial error of state and input estimation. Since these terms are unrelated, one can assume that they are mutually orthogonal. Using the solution given by (9.56) with the above assumption, one can calculate the cross-correlation matrices*[20] $R_{\zeta z}(t,t)\triangleq\mathbb{E}\{\left(\mathbf{1}_N\otimes\bar{v}(t)\right)\zeta(t)z^{\mathrm{T}}(t)\}=\frac{1}{2}\left(\mathbf{1}_N\otimes\bar{v}(t)\right)\left(\mathbf{1}_N\otimes\bar{v}(t)\right)^{\mathrm{T}}$. *By substituting these matrices into (9.58), one can obtain*

$$\mathbb{E}\left\{\frac{\mathrm{d}z(t)}{\mathrm{d}t}z^{\mathrm{T}}(t)\right\} = \mathcal{S}J(t)+\phi(t)\mathbb{E}\{z^{\mathrm{T}}(t)\}+\tfrac{1}{2}\left(\mathbf{1}_N\otimes\bar{v}(t)\right)\left(\mathbf{1}_N\otimes\bar{v}(t)\right)^{\mathrm{T}}+\tfrac{1}{2}R(t)R^{\mathrm{T}}(t). \quad (9.59)$$

Notice that the second term of (9.57) is just the transpose of the first term; hence, from (9.59), (9.57) is equivalent to

$$\dot{J}(t) = \mathcal{S}J(t)+J(t)\mathcal{S}^{\mathrm{T}}+\left(\phi(t)\mathbb{E}\{z^{\mathrm{T}}(t)\}+\mathbb{E}\{z(t)\}\phi^{\mathrm{T}}(t)\right)+\left(\mathbf{1}_N\otimes\bar{v}(t)\right)\left(\mathbf{1}_N\otimes\bar{v}(t)\right)^{\mathrm{T}}+R(t)R^{\mathrm{T}}(t) \quad (9.60)$$

From a practical standpoint, the overall coestimation performance gets better as the error covariance $J(t)$ *gets smaller. Note that* $J(t)$ *can be obtained by solving (9.60) with the initial condition* $J(0)=J_0$, *where* J_0 *is the covariance of* $z(0)$ *or the uncertainty of the initial coestimate. However, solving (9.60) and determining the optimal design parameters would require global information. Here, we can observe from the last three terms on the right hand side of (9.60) that the error covariance* $J(t)$ *depends on the input of the process, its time rate of change,* σ_i *and* K_i, $i=1,\ldots,N$, *via* $\phi(t)$, *the intensity of the process noise via* $\bar{v}(t)$, *and the intensity of the sensors' measurement noises as well as* L_i *and* J_i, $i=1,\ldots,N$, *via* $R(t)$. *In addition, a necessary condition for* $\mathcal{S}$ *to be Hurwitz is that* $\bar{A}$ *or* $\bar{A}_i$, $i=1,\ldots,N$, *is Hurwitz. Therefore, we can influence the coestimation performance without utilizing global information by choosing* σ_i, K_i, L_i, J_i *such that* $\bar{A}_i$ *is Hurwitz for each agent as mentioned in Section 9.2.2 and the bound (9.52) is small, simultaneously.*

[20] We refer to (Chapter 9.4, [169]) for a similar analysis.

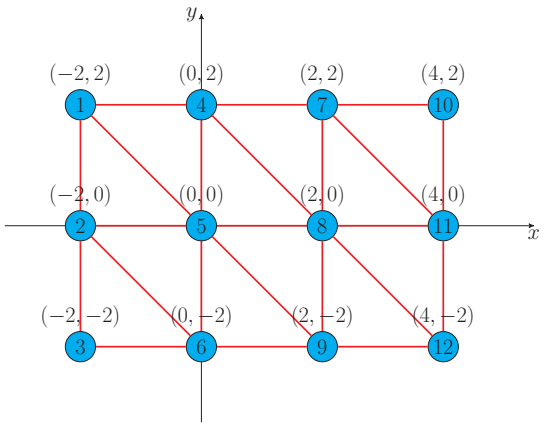


Figure 9.2: Communication graph of the time-varying heterogeneous sensor network with 12 nodes (lines denote communication links and circles denote nodes).

9.5 Illustrative Numerical Examples

We now present two numerical examples to illustrate the proposed distributed input and state coestimation methodology. Specifically, the first example shows the behavior of the sensor network in the absence of noise, while the second example shows the behavior of the sensor network when the dynamics of the process of interest is a stochastic process and sensors contain noise.

9.5.1 Example 1

From [1, 157, 158], a process representing a linear target motion satisfying the dynamics given by (9.1) is considered here with

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_{\mathrm{n}1}^2 & -2\omega_{\mathrm{n}1}\xi_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_{\mathrm{n}2}^2 & -2\omega_{\mathrm{n}2}\xi_2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ \omega_{\mathrm{n}1}^2 & 0 \\ 0 & 0 \\ 0 & \omega_{\mathrm{n}2}^2 \end{bmatrix}, \tag{9.61}$$

where $\omega_{n1} = 1.2, \xi_1 = 0.9, \omega_{n2} = 1.3$, and $\xi_2 = 0.5$. This process composes of two decoupled systems in which the first and third states represent the target's positions in x and y directions, while the second and fourth states represent the target's velocities in x and y directions. The input of the process and the initial conditions of the states are respectively set to $w(t) = [2.5\sin(0.3t) + 1.5,\ 1.5\cos(0.5t)]^{\mathrm{T}}$ and $x_0 = [-2.5,\ 0.5,\ 2.5,\ 0.25]^{\mathrm{T}}$.

A sensor network with 12 nodes exchanging information according to an undirected and connected graph is utilized and arranged spatially as shown in Figure 9.2 for this example. Here, we consider the active and passive node roles are varying over time. Particularly, a sensor's sensing range is defined as a circle with radius $r = 2.5$ centered at each node. When the target (position) enters a sensor's sensing range, the sensor (smoothly) becomes active. Conversely, when the target leaves a sensor's sensing range, it (smoothly) becomes passive. Once again, the transition for $g_i(t)$ is adopted from (Figure 2(d), [152]) with $g_i(t) = e^{-\beta t}$ when node i is switching from 1 to 0, and $g_i(t) = 1 - e^{-\beta t}$ when node i is switching from 0 to 1, where $\beta \in \mathbb{R}_+$.

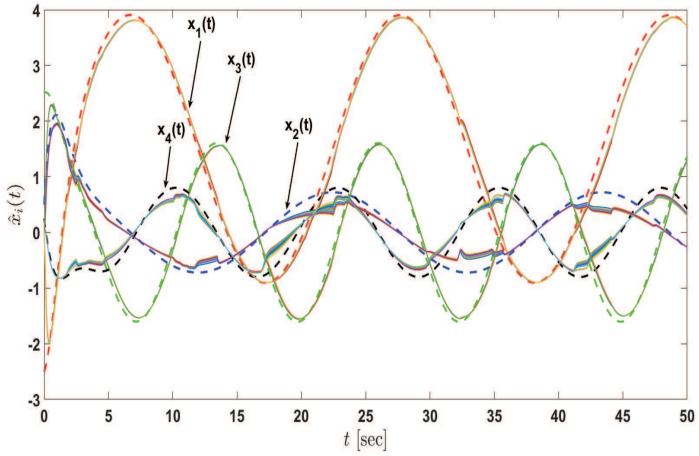


Figure 9.3: The time evolution of $\hat{x}_i(t),\ i = 1,\ldots,N$, of the considered time-varying heterogeneous sensor network under the proposed distributed "coestimation" architecture given by (9.22) and (9.23) (the dashed lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

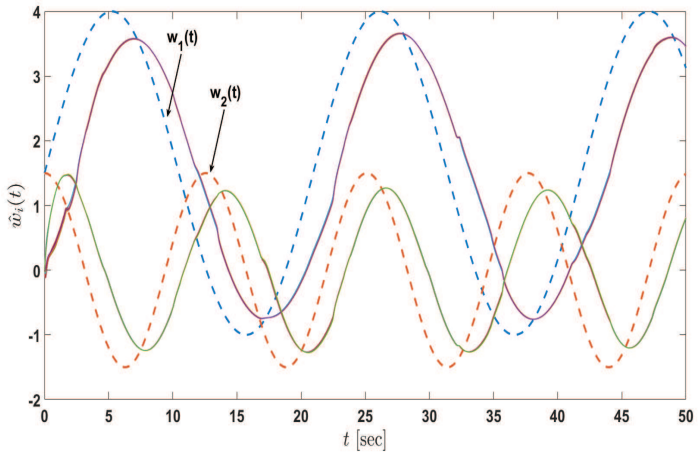


Figure 9.4: The time evolution of $\hat{w}_i(t),\ i = 1,\ldots,N$, of the considered time-varying heterogeneous sensor network under the proposed distributed "coestimation" architecture given by (9.22) and (9.23) (the dashed lines denote the inputs of the actual process and the solid lines denote the input estimates of nodes).

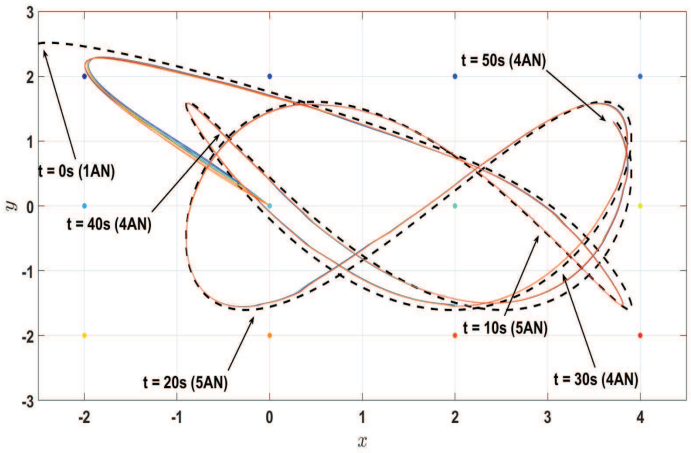


Figure 9.5: Position estimates (first and third states of the process) of the considered time-varying heterogeneous sensor network under the proposed distributed "coestimation" architecture given by (9.22) and (9.23) (the dashed line denotes the trajectory of the actual process (i.e. the combination of the first and third state) and the solid lines denote the state estimates of nodes). Here, AN stands for the the active nodes.

The sensing capability of each node is given by (9.2) with $C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ for the odd index nodes and $C_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ for the even index nodes. Similarly, σ_i is respectively set to 0.01 and 0.001 for odd and even index nodes. The pair (A, C_i) is observable for all $i = 1, \ldots, 12$ in this example; hence, collective observability holds. Furthermore, all nodes' estimations are set to zero initial conditions and gain matrices are chosen such that $J_i = \text{diag}([20; 20])$, $K_i = \text{diag}([10; 10])$ for $i = 1, \ldots, N$. The odd index nodes are subject to $L_i = \begin{bmatrix} 20.13 & 1.33 & 0.00 & 0.00 \\ 0.00 & 0.00 & 20.32 & 3.19 \end{bmatrix}^{\mathrm{T}}$, while the even index nodes are subject to $L_i = \begin{bmatrix} -40.17 & 57.54 & 4.53 & -6.45 \\ 4.14 & -5.88 & -40.20 & 60.42 \end{bmatrix}^{\mathrm{T}}$. For all nodes, we set $\alpha = 25$. In addition, we obtain the common P_i by solving the linear matrix inequalities (9.28) and (9.29) with $\varepsilon = 0.000001$ for the odd nodes that results in

$$P_1 = \begin{bmatrix} 0.937 & 0.211 & 0.000 & 0.000 & 0.907 & 0.000 \\ 0.211 & 0.333 & 0.000 & 0.000 & 0.191 & 0.000 \\ 0.000 & 0.000 & 0.928 & 0.184 & 0.000 & 0.905 \\ 0.000 & 0.000 & 0.184 & 0.361 & 0.000 & 0.184 \\ 0.907 & 0.191 & 0.000 & 0.000 & 0.986 & 0.000 \\ 0.000 & 0.000 & 0.905 & 0.184 & 0.000 & 1.010 \end{bmatrix}, \tag{9.62}$$

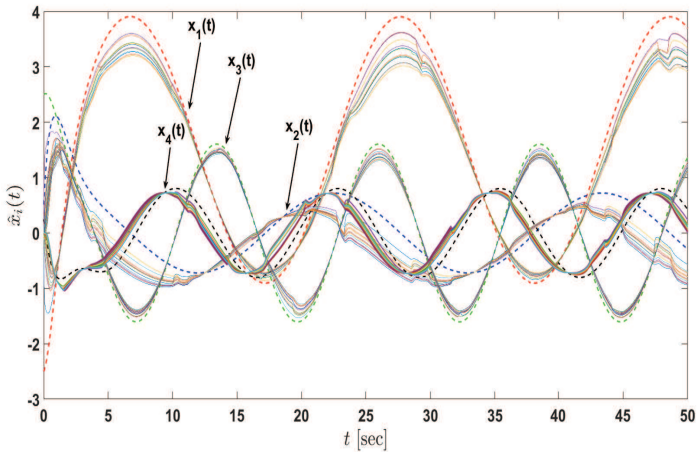


Figure 9.6: The time evolution of $\hat{x}_i(t)$, $i = 1, \ldots, N$, of the considered time-varying heterogeneous sensor network under the recent distributed "estimation" architecture in [1] given by (9.36) and (9.37) (the dashed lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

and $\varepsilon = 0.0001$ for the even nodes that results in

$$P_2 = \begin{bmatrix} 1.907 & 1.744 & -0.034 & 0.011 & 1.895 & -0.035 \\ 1.744 & 2.118 & -0.031 & 0.044 & 1.773 & -0.033 \\ -0.034 & -0.031 & 0.862 & 0.649 & -0.036 & 0.856 \\ 0.011 & 0.044 & 0.649 & 1.106 & 0.009 & 0.680 \\ 1.895 & 1.773 & -0.036 & 0.009 & 2.018 & -0.052 \\ -0.035 & -0.033 & 0.856 & 0.680 & -0.052 & 0.980 \end{bmatrix}. \tag{9.63}$$

That is, $P_1 = P_3 = P_5 = P_7 = P_9 = P_{11}$ and $P_2 = P_4 = P_6 = P_8 = P_{10} = P_{12}$. Based on the matrix P_i, $i = 1, 2, \ldots, 12$, we obtain H_i from (9.34) and the matrices M_i, S_i, T_i, and N_i are selected accordingly.

For the proposed distributed input and state "coestimation" architecture given by (9.22) and (9.23), the process states and inputs are closely estimated as shown in Figures 9.3 and 9.4. Specifically, it is illustrated in Figure 9.5 that the sensor network is able to estimate the trajectory of the target (the first and third states of the process). We note that the lag of the input estimate in Figure 9.4 can be reduced by increasing J_i. However, J_i may not be trivially selected as large while satisfying the linear matrix inequalities (9.28) and (9.29) simultaneously. In addition, the numerical results in Figures 9.6, 9.7 and 9.8 utilizing the architecture in [1] (i.e, utilizing the distributed input and state "estimation" law given by (9.36) and (9.37)) are included here for comparison purposes. These results are generated under the same scenario outlined above including the dynamics of the process, communication graph of nodes, and

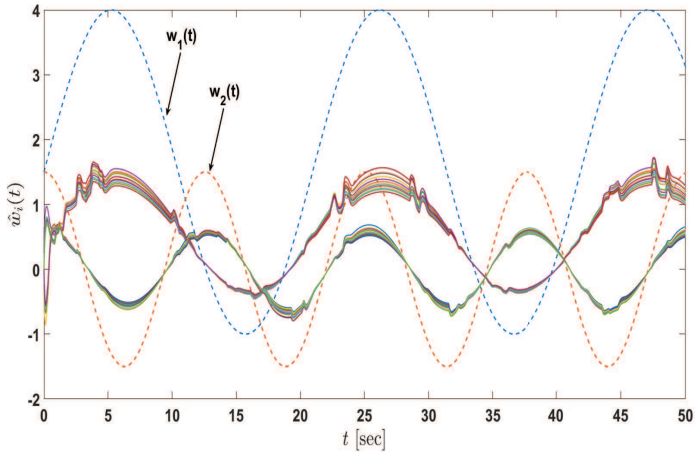


Figure 9.7: The time evolution of $\hat{w}_i(t),\ i = 1,\ldots,N$, of the considered time-varying heterogeneous sensor network under the recent distributed "estimation" architecture in [1] given by (9.36) and (9.37) (the dashed lines denote the inputs of the actual process and the solid lines denote the input estimates of nodes).

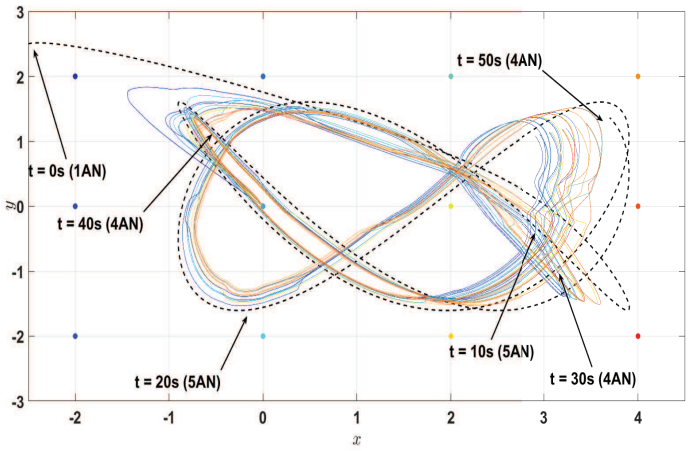


Figure 9.8: Position estimates (first and third states of the process) of the considered time-varying heterogeneous sensor network under the recent distributed "estimation" architecture in [1] given by (9.36) and (9.37) (the dashed line denotes the trajectory of the actual process (i.e. the combination of the first and third state) and the solid lines denote the state estimates of nodes). Here, AN stands for the the active nodes.

sensors' modalities. Figures 9.3, 9.4 and 9.5 clearly highlight the substantially improved dynamic input and state fusion performance of the proposed distributed "coestimation" architecture of this paper over the the distributed "estimation" approach in [1], which is depicted by Figures 9.6, 9.7 and 9.8. ▲

9.5.2 Example 2

In this example, we consider the same setup presented in Example 1 with the process subject to the stochastic dynamics given by (9.40), where the noise intensity function $v(t)$ is a constant vector with elements assigned randomly within the range $(0, 0.2)$. In addition, the observation of each sensor is given

by (9.41) and the noise intensity function $h_i(t)$ is considered as a constant vector with elements assigned randomly within the range $(0, 0.5)$. All other design parameters are chosen as in Example 1.

The simulations are run repeatedly five times to investigate the effect of the noises on the process and the performance of the sensor network under the proposed algorithm. For the proposed distributed input and state "coestimation" architecture given by (9.22) and (9.23) (see Remark 9.4.1), sensor network nodes are able to closely estimate the process states and inputs as shown in Figures 9.9, 9.10 and 9.11, respectively. Specifically, Figure 9.9 shows the estimates of the first and third states of the process, while Figure 9.10 shows the estimates of the second and fourth ones with the shaded areas being the boundaries of the process when the simulation repeated five times. Since estimates of the second and fourth states directly influenced by the input estimates, the noise intensities in these signals are greater comparing to estimates of the first and third states. ▲

9.6 Conclusion

Considering an important practical class of *heterogeneous sensor networks* with both *nonidentical node information roles* and *nonidentical node modalities*, a new dynamic information fusion framework was documented in this paper. The proposed framework involved a *distributed input and state coestimation* algorithm for each node such that the time evolution of input and state updates *both* depend on the *local* input and

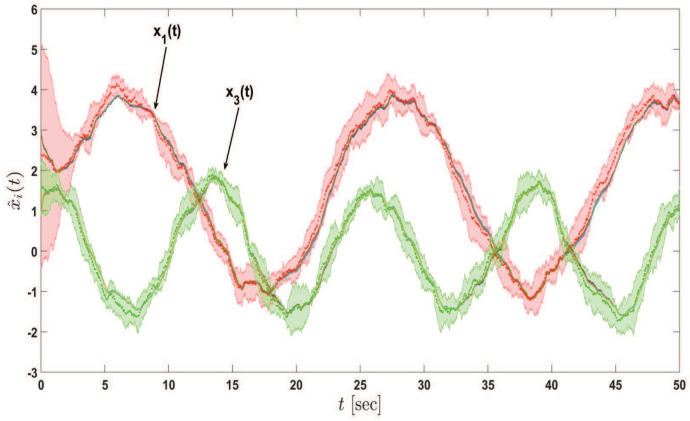


Figure 9.9: The time evolution of the first and third states of $\hat{x}_i(t)$, $i = 1, \ldots, N$, of the considered time-varying heterogeneous sensor network under the proposed distributed "coestimation" architecture given by (9.22) and (9.23) (the dashed lines denote the states of the actual process, the shaded areas denote the boundary of the process' states when repeatedly run five times, and the solid lines denote the state estimates of nodes).

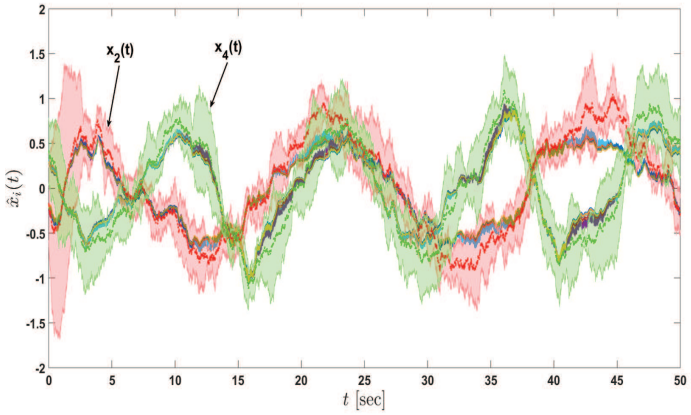


Figure 9.10: The time evolution of the second and the fourth states of $\hat{x}_i(t),\ i = 1,\ldots,N$, of the considered time-varying heterogeneous sensor network under the proposed distributed "coestimation" architecture given by (9.22) and (9.23) (the dashed lines denote the states of the actual process, the shaded areas denote the boundary of the process' states when repeatedly run five times, and of the actual process and the solid lines denote the state estimates of nodes).

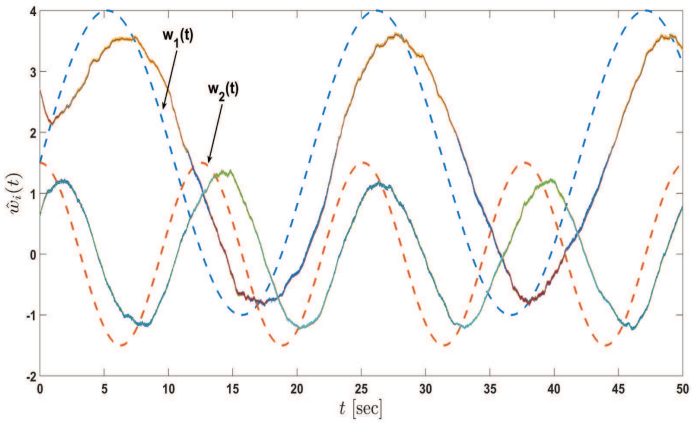


Figure 9.11: The time evolution of $\hat{w}_i(t),\ i = 1,\ldots,N$, of the considered time-varying heterogeneous sensor network under the proposed distributed "coestimation" architecture given by (9.22) and (9.23) (the dashed lines denote the inputs of the actual process and the solid lines denote the input estimates of nodes).

state information exchanges. We first considered *fixed* active and passive node roles subject to nonidentical active node modalities (Section 9.2) and then provided generalizations to the case of *time-varying* active and passive node roles (Section 9.3) as well as to the stochastic case involving noise in the process and the node observations (Section 9.4). Furthermore, we analytically proved all the presented results using tools and methods from Lyapunov theory and linear matrix inequalities, where *local sufficient conditions* were also given for each node. Finally, illustrative numerical examples demonstrated that nodes executing the

proposed coestimation architecture can closely estimate both states and inputs of the considered process of interest.

The results presented in this paper can be useful for several future research directions for dynamic data-driven applications including but not limited to: *a)* From a practical standpoint, the system matrix and the input matrix for the considered process in this paper can be extended to the case when they are both subject to system uncertainties. *b)* From another practical standpoint, one can consider processes involving nonlinear functions for further generalizing the presented results of this paper. *c)* Tools and methods from recent event-triggered control theory developments can be utilized in order to reduce the communication cost between sensor nodes. *d)* For capturing cases when the process of interest leaves the sensing field of the network for certain time periods, tools and methods from filtering and estimation theories can be utilized with the presented results of this paper. *e)* The presented setup can be extended to involve mobile sensor nodes. *f)* The presented results can be applied to real-world sensor networks through experiments in order to bridge the gap between theory and practice.

Chapter 10: Dynamic Information Fusion with the Integration of Local Observers, Value of Information, and Active-Passive Consensus Filters*

This paper proposes a dynamic information fusion framework for sensor networks with the integration of local observers, value of information, and active-passive consensus filters as well as a layer to monitor the validity of information. Specifically, we consider a process of interest consisting of multiple subprocesses (for example, multiple targets to be monitored). The heterogeneity in the sensor networks is considered and handled in many aspects such as nodes are allowed to have different sensing capabilities, different information node roles (active and/or passive; that is, a node can be subject to observations of the process or to no observation), and different weights on information (value of information). In addition, the information validity monitor layer allows operators to evaluate the reliability of the fused information based on the local feedbacks received from the sensor network. Several illustrative numerical examples are also presented to illustrate the efficacy and discuss the practical aspects of the proposed dynamic information fusion framework.

10.1 Introduction

With the remarkable technological developments in the past two decades, advanced devices such as autonomous mobile robots, and sensors have become affordable to deploy in a large quantities. This also leads to a need in the development of advanced algorithms to gather and integrate information as well as to control such multiagent systems. In particular, dynamic information fusion in sensor networks plays an important roles in a wide array of applications for both scientific, civilian and military purposes. One of the main challenges in dynamic information fusion is the heterogeneity of sensor networks. The sources of this heterogeneity include the differences in sensor modalities, the quality of sensing information (value

*This chapter is previously published in [156]. Permission is included in Appendix I.

of information), and the information roles of nodes (active and passive; that is, a node can be subject to observations of the process or to no observation), to name but a few examples.

While information roles of nodes and active-passive consensus filter are recently investigated in [68, 73, 154, 170, 171] and references therein, these results often consider scalar integrator dynamics and/or lack a complete structure to process the local information before the fusion such as utilizing local observers to extract more information and assigning weights on information. Although the distributed algorithms in [3] and [2] consider the differences in sensor modalities, the information roles of nodes in these results are not explicitly discussed. Several works such as [67, 71, 172] consider the value of information, yet information roles of nodes and/or heterogeneous modalities are not considered. Nonetheless, the aforementioned works do not have a direct architecture to quantify and evaluate the quality of fused information of sensor networks in real-time.

The contribution of this paper is to propose a dynamic information fusion framework for sensor networks with the integration of local observers, value of information, and active-passive consensus filters as well as a layer to monitor the validity of information; see Figure 10.1. Specifically, we consider a process of interest consists of multiple subprocesses (e.g, multiple targets to be monitored). The heterogeneity in the sensor networks is considered and handled in many aspects such as nodes are allowed to have different sensing capabilities, different information node roles (active and passive), and different weights on information (value of information). In addition, the information validity monitor layer allows operators to evaluate the reliability of the fused information based on the local feedbacks received from the sensor network. Several illustrative numerical examples are also presented to illustrate the efficacy and discuss the practical aspects of the proposed dynamic information fusion framework.

The organization of this paper is as follows. In Section 10.2, we present the setup of the process, the sensor network, the structure of the local observers, and the value of information. In Section 10.3, the active-passive consensus filter with fixed information node roles is introduced and analyzed, as an intermediate result. The main result of this paper is then presented in Section 10.4, which is a practical extension of the result of Section 10.3 to the time-varying case. The information validity monitor layer is presented in Section 10.5 and illustrative numerical examples together with some discussions are provided in Section 10.6. Finally, concluding remarks are summarized in Section 10.7. For readers, we refer to Appendix G for the notations, mathematical preliminaries, and necessary lemmas for the results of this paper.

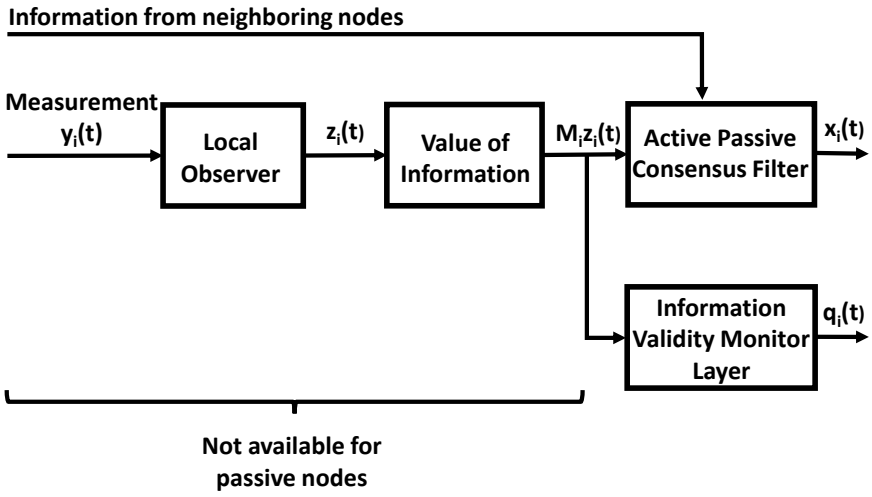


Figure 10.1: The dynamic information fusion framework of an individual node with the integration of a local observer, value of information, active-passive consensus filter, and information validity monitor layer.

10.2 Problem Setup

10.2.1 Considered Process and the Sensor Network

Consider a large-scale process of interest with the dynamics given by

$$\dot{z} = Az(t),\ z(0) = z_0, \tag{10.1}$$

where $z(t) \in \mathbb{R}^n$ denotes the unmeasurable process state vector,

$$A = \underset{i=1,2}{\text{block–diag}}(A_1, A_2, \ldots, A_m) \in \mathbb{R}^{n\times n}, \tag{10.2}$$

is the system matrix with $A_h \in \mathbb{R}^{n_h\times n_h}$ for $h = 1,2,\ldots,m$ and $n_h \leq n, \sum_{h=1}^{m} n_h = n$. While (10.1) adopts a simple structure in order to allow us to directly focus on the overarching contribution of this paper (i.e., a new dynamic information fusion framework), it can represent linear or linearized, controlled or uncontrolled process dynamics. Note that the contribution of this paper can be readily extended to the cases, where (10.1) include measurement noise and/or uncertainties resulting from modeling efforts. Furthermore, the block-diagonal structure of the system matrix A indicates that the process can be decomposed into m subprocesses. An example of such a process is independent multiple targets that need to be monitored on an observation field, which presents the overarching application focus of this paper.

Next, consider a sensor network with N nodes exchanging information among each other using their local measurements through a connected, undirected graph $\mathcal{G}$. If a node i, $i = 1,\ldots,N$ has no observations, then we say that it is a "passive node". On the other hand, if a node i, $i = 1,\ldots,N$ is subject to observations of process (10.1) given by

$$y_i(t) = {}^{g}C_i z(t), \tag{10.3}$$

where $y_i(t) \in \mathbb{R}^p$ denotes the measurable process output for node i, $i = 1,\ldots,N$, and ${}^{g}C_i \in \mathbb{R}^{p\times n}$ denotes the system output matrix with g is the sensor's category defined below, then we say that node i is an "active node".

Remark 10.2.1. *Here, we consider that the sensors can have different sensing capabilities and can be categorized into multiple categories. We define a Category I sensor as* ${}^{I}C_i = \begin{bmatrix} 0 & \ldots & {}^{j}\bar{C}_i & \ldots & 0 \end{bmatrix}$ *such that* ${}^{j}\bar{C}_i \in \mathbb{R}^{p\times n_j}$ *and the pair* $(A_j, {}^{j}\bar{C}_i)$ *is detectable, where* $j \in \mathbb{Z}_+$ *denotes the corresponding subprocess that node* i *can sense, and* $j \in [1,m]$ *(e.g., a Category I sensor can observe a specific subprocess). Next, we define a Category II sensor as a combination of two or more Category I sensors, for example,* ${}^{II}C_i = \begin{bmatrix} 0 & \ldots & {}^{j}\bar{C}_i & \ldots & {}^{k}\bar{C}_i & \ldots & 0 \end{bmatrix}$ *such that* ${}^{j}\bar{C}_i \in \mathbb{R}^{p\times n_j}$, ${}^{k}\bar{C}_i \in \mathbb{R}^{p\times n_k}$ *and the pairs* $(A_j, {}^{j}\bar{C}_i)$, $(A_k, {}^{k}\bar{C}_i)$ *are detectable, where* $j,k \in \mathbb{Z}_+$ *and* $j,k \in [1,m]$ *(e.g., a Category II sensor can simultaneously observe several subprocesses). We also define a Category III sensor as a generalized sensor that does not possess a detectable pair (or an observable pair), for example,* ${}^{III}C_i = \begin{bmatrix} 0 & \ldots & {}^{j}\bar{c}_i & \ldots & {}^{k}\bar{c}_i & \ldots & 0 \end{bmatrix}$, *where* ${}^{j}\bar{c}_i, {}^{k}\bar{c}_i \in \mathbb{R}$, $j,k \in \mathbb{Z}_+$ *and* $j,k \in [1,n]$ *(e.g, a Category III sensor can observe a state of a subprocess or several states of several subprocesses). A sensor's category is not necessarily limited to the ones that we introduce above but can be extended to the mixtures of those categories as well.*

10.2.2 Local Observers and the Value of Information Matrix

We first introduce here the construction of local observers based on the local measurements $y_i(t)$. Depending on the sensor's category, the value of information matrix is then constructed. Because of the diagonal structure of the system matrix A, we can construct the local observer vector $z_i(t) \in \mathbb{R}^n$ for the process based on the type and capability of each active node i. Specifically, if the sensor is in either Category

I or II, a subprocess state can be estimated by the local (Luenberger) observer given by

$$\dot{s}_i(t) = A_h s_i(t) + {}^{h}L_i(y_i(t) - {}^{h}\bar{C}_i s_i(t)),\ s_i(0) = s_{i0}, \tag{10.4}$$

where h is an index of a corresponding subprocess of A that node i can observe, $s_i \in \mathbb{R}^{n_h}$ is the local state estimate of a subprocess A_h, ${}^{h}\bar{C}_i$ is the output matrix corresponding to states observed by node i on the subprocess A_h, and ${}^{h}L_i \in \mathbb{R}^{n_h \times p}$ is the corresponding local observer gain for node i. From this point, the local observer of node i, $z_i(t) \in \mathbb{R}^n$ can be constructed, for example $z_i(t) \triangleq \begin{bmatrix} 0 & \dots & s_i^{\mathrm{T}}(t) & \dots & 0 \end{bmatrix}^{\mathrm{T}}$, where the position of $s_i(t)$ is corresponding to the state of the subprocess A_h in $z(t)$. If the sensor is in Category III, there is no theoretical need for a local observer. Therefore, $z_i(t)$ can be constructed directly from $y_i(t)$ as $z_i(t) \triangleq \begin{bmatrix} 0 & \dots & y_i(t)^{\mathrm{T}} & \dots & 0 \end{bmatrix}^{\mathrm{T}}$, where the position of $y_i(t)$ elements are corresponding to the substate of $z(t)$ that node i can sense.

Based on the sensor's type and capability, in addition, the value of information matrix has a natural diagonal structure in the form given by

$$M_i \triangleq \mathrm{diag}(m_i) \in \mathbb{R}^{n \times n}, \tag{10.5}$$

where $m_i \triangleq \begin{bmatrix} m_{i1} & m_{i2} & \dots & m_{in} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^n$ for $i = 1, 2, \dots, N$ and m_{ir} are nonnegative scalar weights with $r = 1, 2, \dots, n$. A substate of $z_i(t)$ is called "valid" when its weight is positive. Conversely, a substate of $z_i(t)$ is called "invalid" when its weight is 0. Under certain circumstances (e.g., see Ref. [71] and references therein), a sensor can be subject to some observations, yet the information may not be reliable (and so its substates are set to be invalid). We also assume that $z_i(t)$ and $\dot{z}_i(t)$ are bounded. Considering the multivehicle application focus of this paper as the process to be monitored, this assumption generally holds, because the vehicles' properties such as positions and velocities are bounded, on the observation field. After $z_i(t)$ and the value of information matrix M_i are constructed, they are then passed to the active-passive consensus filter for information fusion and to the information validity monitor layer for evaluation of the quality of fused information as illustrated in Figure 10.1.

For the purpose of establishing an intermediate result, Section 10.3 next presents the active-passive consensus filter for the case where the active-passive roles of nodes are fixed for each node and assume the local estimation $z_i(t)$ is constant. We then introduce the main result of this paper by extending the result

of Section 10.3 to the actual practical case, where both the active-passive role of each node and $z_i(t)$ are time-varying in Section 10.4.

10.3 Active-Passive Consensus Filters with Fixed Information Node Roles

10.3.1 Proposed Architecture

The active-passive consensus filter aims to drive substates of each node to the average of all valid active corresponding substates (i.e, an agent needs to be active and its corresponding substates need to have positive weights) of the vectors of the local observers $z_i(t)$, $i = 1,2,\ldots,N$. Specifically, in this section, we assume that the active-passive role of each node is fixed and the vectors of the local observers $z_i(t) \equiv z_i$, $i = 1,2,\ldots,N$, are constants for the sake of establishing an intermediate result for the following sections of this paper. Mathematically speaking, we consider the proposed active-passive consensus filter given by

$$\dot{x}_i(t) = -\alpha \sum_{i\sim j}(x_i(t)-x_j(t)) + \alpha \sum_{i\sim j}(\xi_i(t)-\xi_j(t)) - \alpha k_i M_i(x_i(t)-z_i),\ x_i(0)=x_{i0}, \tag{10.6}$$

$$\dot{\xi}_i(t) = -\gamma \sum_{i\sim j}(x_i(t)-x_j(t)),\ \xi_i(0)=\xi_{i0}, \tag{10.7}$$

where $x_i(t) \in \mathbb{R}^n$, $\xi_i(t) \in \mathbb{R}^n$, $z_i \in \mathbb{R}^n$ denote the state, the integral action, and the local observer vector of node i, $i = 1,\ldots,N$, respectively. Here, M_i is the value of information matrix defined in (10.5) and $\alpha,\gamma \in \mathbb{R}_+$ are constant gains. Under the assumption that the information node roles are fixed, $k_i = 1$ for active nodes and $k_i = 0$ for passive nodes.

Remark 10.3.1. *We introduce a similar active-passive consensus filter in Ref. [71]; however, our previous result documented in that paper only considered scalar integrator dynamics. We next present the stability properties of (10.6) and (10.7) having $x_i(t) \in \mathbb{R}^n$ and $\xi_i(t) \in \mathbb{R}^n$. In addition, we note that the stability results documented in Ref. [71] can also be applied to each scalar element of (10.6) and (10.7) in parallel. Yet, the presented stability properties of the next subsection is compact in the sense that we do not focus on scalar elements of (10.6) and (10.7) but to their compact form; hence, from our standpoint, it is worth to include the following content to this paper for completeness.*

10.3.2 Stability Analysis

Let $x(t) \triangleq \begin{bmatrix} x_1^{\mathrm{T}}(t) & x_2^{\mathrm{T}}(t) & \ldots & x_N^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}$, $\xi(t) \triangleq \begin{bmatrix} \xi_1^{\mathrm{T}}(t) & \xi_2^{\mathrm{T}}(t) & \ldots & \xi_N^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}$, and $\zeta \triangleq \begin{bmatrix} z_1^{\mathrm{T}} & z_2^{\mathrm{T}} & \ldots & z_N^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. The proposed algorithm (10.6) and (10.7) can be rewritten in the compact form given by

$$\begin{aligned} \dot{x}(t) &= -\alpha(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)x(t) + \gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\xi(t) - \alpha\mathcal{M}x(t) + \alpha\mathcal{M}\zeta \\ &= -\alpha F x(t) + \gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\xi(t) + \alpha\mathcal{M}\zeta, \;\; x(0) = x_0, \qquad (10.8) \\ \dot{\xi}(t) &= -\gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)x(t), \;\; \xi(0) = \xi_0, \qquad (10.9) \end{aligned}$$

where

$$\mathcal{M} \triangleq \underset{i=1,2}{\text{block}\text{–}\text{diag}}(k_1 M_1, k_2 M_2, \ldots, k_N M_N) \in \mathbb{R}^{Nn \times Nn}, \qquad (10.10)$$

and $F \triangleq \left(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n + \mathcal{M}\right) \in \mathbb{R}^{Nn \times Nn}$. Since M_i for all $i = 1, 2, \ldots, N$ are diagonal matrices, $\mathcal{M}$ is then a diagonal matrix. Furthermore, since $(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)$ and $\mathcal{M}$ are nonnegative definite, F is either nonnegative definite or positive definite.

Note that a substate of z_i is said to be valid if its weight in M_i is positive. In addition, z_i is active when $k_i = 1$. Since we are interested in driving substates of $x_i(t)$ of each node to the average of all valid active corresponding substates of local observer vectors z_i , $i = 1, 2, \ldots, N$ in the network, define

$$\begin{aligned} S &\triangleq (\mathbf{1}_N^{\mathrm{T}} \otimes \mathrm{I}_n)\mathcal{M}(\mathbf{1}_N \otimes \mathrm{I}_n) \in \mathbb{R}^{n \times n} \\ &= k_1 M_1 + k_2 M_2 + \ldots + k_N M_N, \qquad (10.11) \end{aligned}$$

as the diagonal matrix with the total weight of valid active substates of z_i on the diagonal. We now let

$$\varepsilon \triangleq S^{+}(\mathbf{1}_N^{T} \otimes \mathrm{I}_n)\mathcal{M}\zeta \in \mathbb{R}^n \qquad (10.12)$$

be the average of all valid active substates of local observer vectors z_i, $i = 1, 2, \ldots, N$ in the network. Note that since z_i, $i = 1, 2, \ldots, N$ are constant, ζ and ε are also constant in this case.

Remark 10.3.2. *In this section, we do not assume collective observability (see, for example, [1, 2]). Collective observability can imply that at any moment there exists at least one node that is active and has valid information for each substate of z_i; hence, the matrix S has full rank and is invertible. Recall that S is a diagonal matrix with each element on the diagonal is the total weight of the corresponding valid active substate of local observer vectors z_i. In addition, a substate of $x_i(t)$ and $z_i(t)$ are said to be completely passive if there is no node in the network can observe or has a valid observation on that corresponding substate. In other words, if the substate r is completely passive (that is, $k_i m_{ir} = 0$ for all nodes $i = 1,\ldots,N$ where m_{ir} is the r-th element on the diagonal of the matrix M_i), then S has a row of zeros (the r-th row). A nice property of the pseudo-inverse of a diagonal matrix is that each positive diagonal element is inversed, except zero diagonal elements are still zeros. For example, if $S = \text{diag}\Big(\begin{bmatrix} a & b & 0 \end{bmatrix}\Big)$, then $S^+ = \text{diag}\Big(\begin{bmatrix} a^{-1} & b^{-1} & 0 \end{bmatrix}\Big)$ for $a,b \in \mathbb{R}_+$. In the case if a substate of z_i is completely passive, the corresponding substate of $x_i(t)$ converges to average consensus and this happens only when an element on the diagonal of S is 0. This is shown and discussed later in Remark 10.3.4 of Section 10.3.3.*

We now define the error $\delta(t) \triangleq \big(x(t) - (\mathbf{1}_N \otimes \varepsilon)\big) \in \mathbb{R}^{Nn}$ and taking its time derivative to obtain

$$\begin{aligned}
\dot{\delta}(t) &= -\alpha F\big(\delta(t) + (\mathbf{1}_N \otimes \varepsilon)\big) + \gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\xi(t) + \alpha\mathcal{M}\zeta \\
&= -\alpha F\delta(t) - \alpha\mathcal{M}(\mathbf{1}_N \otimes \varepsilon) + \gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\xi(t) + \alpha\mathcal{M}\zeta \\
&= -\alpha F\delta(t) + \gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\xi(t) - \alpha\mathcal{M}\omega,\ \delta(0) = \delta_0, \qquad (10.13)
\end{aligned}$$

where the third equality comes from the facts that $F = \big(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n + \mathcal{M}\big)$ and $\mathcal{L}(\mathcal{G})\mathbf{1}_N = 0_N$, and $\omega \triangleq \big((\mathbf{1}_N \otimes \varepsilon) - \zeta\big) \in \mathbb{R}^{Nn}$. Next, define $e(t) \triangleq \big(\xi(t) - \frac{\alpha}{\gamma}(\mathcal{L}^+(\mathcal{G}) \otimes \mathrm{I}_n)\mathcal{M}\omega\big) \in \mathbb{R}^{Nn}$ and take its time derivative as

$$\begin{aligned}
\dot{e}(t) &= -\gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\big(\delta(t) + (\mathbf{1}_N \otimes \varepsilon)\big) \\
&= -\gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\delta(t),\ e(0) = e_0. \qquad (10.14)
\end{aligned}$$

From (10.13), Lemma G.0.1, and the definition of $e(t)$, one can write

$$\begin{aligned}
\dot{\delta}(t) &= -\alpha F\delta(t) + \gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\Big(e(t) + \frac{\alpha}{\gamma}(\mathcal{L}^+(\mathcal{G}) \otimes \mathrm{I}_n)\mathcal{M}\omega\Big) - \alpha\mathcal{M}\omega \\
&= -\alpha F\delta(t) + \gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)e(t) + \alpha\Big(\big(\mathcal{L}(\mathcal{G})\mathcal{L}^+(\mathcal{G}) \otimes \mathrm{I}_n\big) - \mathrm{I}_{Nn}\Big)\mathcal{M}\omega \\
&= -\alpha F\delta(t) + \gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)e(t) + \alpha\Big(\big((\mathrm{I}_N - \tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^{\mathrm{T}}) \otimes \mathrm{I}_n\big) - \mathrm{I}_{Nn}\Big)\mathcal{M}\omega
\end{aligned}$$

$$
\begin{aligned}
&= -\alpha F\delta(t) + \gamma(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n)e(t) + \alpha\Big(\big(\mathrm{I}_{Nn} - (\tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^{\mathrm{T}})\otimes \mathrm{I}_n\big) - \mathrm{I}_{Nn}\Big)\mathcal{M}\omega \\
&= -\alpha F\delta(t) + \gamma(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n)e(t) - \frac{\alpha}{N}(\mathbf{1}_N\otimes \mathrm{I}_n)(\mathbf{1}_N^{\mathrm{T}}\otimes \mathrm{I}_n)\mathcal{M}\omega,\ \delta(0) = \delta_0. \qquad (10.15)
\end{aligned}
$$

To further write (10.15) in a simpler form, we now introduce the following lemma.

Lemma 10.3.1. *Let ε be defined by (10.12), $\mathcal{M}$ be defined by (10.10), and $\omega \triangleq \big((\mathbf{1}_N\otimes\varepsilon) - \zeta\big)$. Then,*

$$
(\mathbf{1}_N^{\mathrm{T}}\otimes \mathrm{I}_n)\mathcal{M}\omega = 0. \qquad (10.16)
$$

Proof. See the Appendix H.

Remark 10.3.3. *Although, owing to the assumptions of this subsection, k_i, ζ, and ε are constant, the result of Lemma 10.3.1 is still valid for the case when $k_i(t)$, $\zeta(t)$, and $\varepsilon(t)$ are time-varying.*

Under the result of Lemma 10.3.1, (10.15) can now be simplified as

$$
\dot{\delta}(t) = -\alpha F\delta(t) + \gamma(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n)e(t),\ \delta(0) = \delta_0. \qquad (10.17)
$$

The closed-loop error dynamics of the system given by (10.6) and (10.7) are

$$
\begin{aligned}
\dot{\delta}(t) &= -\alpha F\delta(t) + \gamma(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n)e(t),\ \delta(0) = \delta_0. \qquad (10.18)\\
\dot{e}(t) &= -\gamma(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n)\delta(t),\ e(0) = e_0. \qquad (10.19)
\end{aligned}
$$

We are now ready to state the following result.

Theorem 10.3.1. *Consider a sensor network with N nodes given by (10.6) and (10.7), where nodes exchange information using local measurements under a connected, undirected graph $\mathcal{G}$. Then, the closed loop error dynamics (10.18) and (10.19) are Lyapunov stable and $\delta(t)$ converges to the null space of F.*

Proof. Consider the Lyapunov function candidate given by

$$
V(\delta, e) = \frac{1}{2}\delta^{\mathrm{T}}\delta + \frac{1}{2}e^{\mathrm{T}}e. \qquad (10.20)
$$

Note that $V(0,0)=0$ and $V(\delta,e)>0$ for all $(\delta,e)\neq 0$. The time derivative of (10.20) along the trajectories of (10.18) and (10.19) is given by

$$\begin{aligned}\dot{V}(\delta(t),e(t)) &= -\alpha\delta^{\mathrm{T}}(t)F\delta(t)+\gamma\delta^{\mathrm{T}}(t)(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n)e(t)-\gamma e^{\mathrm{T}}(t)(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n)\delta(t)\\ &= -\alpha\delta^{\mathrm{T}}(t)F\delta(t)\leq 0 \end{aligned} \tag{10.21}$$

Therefore, the closed-loop error dynamics (10.18) and (10.19) are Lyapunov stable. Because $\ddot{V}(\delta(t),e(t))$ is also bounded for all $t\geq 0$, it follows from Barbalat's lemma (see Ref. [115]) that $\lim_{t\to\infty}\dot{V}(\delta(t),e(t))=\lim_{t\to\infty}\Big(-\alpha\delta^{\mathrm{T}}(t)F\delta(t)\Big)=0$. Therefore, as $t\to\infty$,

$$\delta^{\mathrm{T}}(t)F\delta(t)=\delta^{\mathrm{T}}(t)F^{1/2}F^{1/2}\delta(t)=\big(F^{1/2}\delta(t)\big)^{\mathrm{T}}\big(F^{1/2}\delta(t)\big)=\|\big(F^{1/2}\delta(t)\big)\|_2\to 0. \tag{10.22}$$

Note that (10.22) indicates that $\delta(t)$ converges to the null space of $F^{1/2}$. Since F is a symmetric matrix, it is always diagonalizable by an orthogonal matrix $U\in\mathbb{R}^{Nn\times Nn}$ such that $F=U\Lambda U^{\mathrm{T}}$, where $\Lambda\in\mathbb{R}^{Nn\times Nn}$ is the diagonal matrix with eigenvalues of F on the diagonal. As a result, $F^{1/2}=U\Lambda^{1/2}U^{\mathrm{T}}$ since $F^{1/2}F^{1/2}=U\Lambda^{1/2}U^{\mathrm{T}}U\Lambda^{1/2}U^{\mathrm{T}}=U\Lambda U^{\mathrm{T}}=F$. Therefore, $(F^{1/2})^{\mathrm{T}}=(U\Lambda^{1/2}U^{\mathrm{T}})^{\mathrm{T}}=(U^{\mathrm{T}})^{\mathrm{T}}\Lambda^{1/2}U^{\mathrm{T}}=U\Lambda^{1/2}U^{\mathrm{T}}=F^{1/2}$, thus $F^{1/2}$ is also a symmetric matrix. Utilize the result of Lemma G.0.4 for $F^{1/2}$, we have $\mathcal{N}(F^{1/2})=\mathcal{N}(F)$. Hence, $\delta(t)$ converges to the null space of F. ■

We now investigate the null space of $F=\big(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n+\mathcal{M}\big)$ in the next subsection owing to the fact that the above theorem shows that $\delta(t)$ converges to the null space of F.

10.3.3 Convergence Analysis

We first decompose the structure of F as

$$\begin{aligned}F &= \mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n+\mathcal{M}\\ &= \begin{bmatrix}\mathcal{L}_{11}\mathrm{I}_n & \mathcal{L}_{12}\mathrm{I}_n & \dots & \mathcal{L}_{1N}\mathrm{I}_n\\ \mathcal{L}_{21}\mathrm{I}_n & \mathcal{L}_{22}\mathrm{I}_n & \dots & \mathcal{L}_{2N}\mathrm{I}_n\\ \vdots & \vdots & \ddots & \vdots\\ \mathcal{L}_{N1}\mathrm{I}_n & \mathcal{L}_{N2}\mathrm{I}_n & \dots & \mathcal{L}_{NN}\mathrm{I}_n\end{bmatrix}+\begin{bmatrix}k_1M_1 & 0 & \cdots & 0\\ 0 & k_2M_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & k_NM_N\end{bmatrix}\\ &= \begin{bmatrix}\mathcal{L}_{11}\mathrm{I}_n+k_1M_1 & \mathcal{L}_{12}\mathrm{I}_n & \dots & \mathcal{L}_{1N}\mathrm{I}_n\\ \mathcal{L}_{21}\mathrm{I}_n & \mathcal{L}_{22}\mathrm{I}_n+k_2M_2 & \dots & \mathcal{L}_{2N}\mathrm{I}_n\\ \vdots & \vdots & \ddots & \vdots\\ \mathcal{L}_{N1}\mathrm{I}_n & \mathcal{L}_{N2}\mathrm{I}_n & \dots & \mathcal{L}_{NN}\mathrm{I}_n+k_NM_N\end{bmatrix},\end{aligned} \tag{10.23}$$

where $\mathcal{L}_{ij}$ denotes the corresponding element at i-th row and j-th column of the Laplacian matrix $\mathcal{L}(\mathcal{G})$. Recall that $M_i = \text{diag}(m_i)$ is the value of information matrix of node i for $i = 1,2,\ldots,N$, where $m_i \triangleq \begin{bmatrix} m_{i1} & m_{i2} & \ldots & m_{in} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^n$ and m_{ir} are nonnegative scalar weights with $r = 1,2,\ldots,n$. Therefore, $\mathcal{M}$ can be rewritten as

$$\mathcal{M} = \text{diag}\left(\begin{bmatrix} k_1 m_{11} & \ldots & k_1 m_{1n} & \ldots & k_n m_{N1} & \ldots & k_n m_{Nn} \end{bmatrix}^{\mathrm{T}} \right). \tag{10.24}$$

Next, there is a permutation matrix $J \in \mathbb{R}^{Nn \times Nn}$ that allows us to reorder $\delta(t)$ as

$$J\delta(t) = J \begin{bmatrix} \delta_{11}(t) \\ \vdots \\ \delta_{1n}(t) \\ \vdots \\ \delta_{N1}(t) \\ \vdots \\ \delta_{Nn}(t) \end{bmatrix} = \begin{bmatrix} \delta_{11}(t) \\ \vdots \\ \delta_{N1}(t) \\ \vdots \\ \delta_{1n}(t) \\ \vdots \\ \delta_{Nn}(t) \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{Substate 1} \\ \vdots \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{Substate n} \end{matrix} \tag{10.25}$$

where $\delta_{ir}(t)$ indicates the error of substate r, $r = 1,2,\ldots,n$ of node i, $i = 1,2,\ldots,N$. Therefore,

$$\begin{aligned} JFJ^{\mathrm{T}} &= J(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)J^{\mathrm{T}} + J\mathcal{M}J^{\mathrm{T}} \\ &= \begin{bmatrix} \mathcal{L}(\mathcal{G}) & & \\ & \ddots & \\ & & \mathcal{L}(\mathcal{G}) \end{bmatrix} + \begin{bmatrix} \begin{bmatrix} k_1 m_{11} & & \\ & \ddots & \\ & & k_N m_{N1} \end{bmatrix} & & & \\ & \ddots & & \\ & & \begin{bmatrix} k_1 m_{1n} & & \\ & \ddots & \\ & & k_N m_{Nn} \end{bmatrix} \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} \begin{bmatrix} \mathcal{L}_{11}+k_1m_{11} & \cdots & \mathcal{L}_{1N} \\ \vdots & \ddots & \vdots \\ \mathcal{L}_{N1} & \cdots & \mathcal{L}_{NN}+k_Nm_{N1} \end{bmatrix} & & \\ & \ddots & \\ & & \begin{bmatrix} \mathcal{L}_{11}+k_1m_{1n} & \cdots & \mathcal{L}_{1N} \\ \vdots & \ddots & \vdots \\ \mathcal{L}_{N1} & \cdots & \mathcal{L}_{NN}+k_Nm_{Nn} \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} F_1 & & \\ & \ddots & \\ & & F_n \end{bmatrix}, \tag{10.26}$$

where $F_r \triangleq \begin{bmatrix} \mathcal{L}_{11}+k_1m_{1r} & \cdots & \mathcal{L}_{1N} \\ \vdots & \ddots & \vdots \\ \mathcal{L}_{N1} & \cdots & \mathcal{L}_{NN}+k_Nm_{Nr} \end{bmatrix}$ for $r = 1,2,\ldots,n$. Note that, from, for example, Lemma 2 in Ref. [102] or Lemma 3.3 in Ref. [16], if there exists at least one node i for $i = 1,2,\ldots,N$ such that k_im_{ir} is positive, F_r is positive definite and $\mathcal{N}(F_r) = 0$. On the other hand, if a substate r is completely passive, then $k_im_{ir} = 0$ for all i as discussed in Remark 10.3.2. In this case, $F_r = \mathcal{L}(\mathcal{G})$, and hence, $\mathcal{N}(F_r) = \mathcal{N}(\mathcal{L}(\mathcal{G})) = \text{span}(\mathbf{1}_N)$.

Let $f(r)$ for $r = 1,2,\ldots,n$ be a function such that

$$f(r) \triangleq \begin{cases} 0 & \text{if there exists at least one node } i \text{ in the network} \\ & \text{such that } k_im_{ir} \text{ is positive,} \\ 1 & \text{otherwise.} \end{cases} \tag{10.27}$$

We now can write

$$\mathcal{N}(F_r) = \text{span}\big(f(r)\mathbf{1}_N\big). \tag{10.28}$$

Therefore,

$$\mathcal{N}(JFJ^{\mathrm{T}}) = \text{span}\left(\begin{bmatrix} f(1)\mathbf{1}_N \\ \vdots \\ f(n)\mathbf{1}_N \end{bmatrix}\right) = \text{span}(\nu) = a\nu, \tag{10.29}$$

where $a \in \mathbb{R}$. Note that if each specific substate r has at least one positive scalar weight m_{ir} for for some $i \in \mathcal{V}_{\mathcal{G}}$, then ν is a zero vector. In all other cases, $\nu \in \mathbb{R}^{Nn}$ is a non-zero vector.

Since J is invertible, $\text{rank}(JFJ^{\mathrm{T}}) = \text{rank}(JF) = \text{rank}(F)$. In addition, $\text{rank}(JFJ^{\mathrm{T}}) + \text{def}(JFJ^{\mathrm{T}}) = \text{rank}(JF) + \text{def}(JF) = \text{rank}(F) + \text{def}(F) = Nn$. Therefore, $\text{def}(JFJ^{\mathrm{T}}) = \text{def}(JF) = \text{def}(F)$. As a result, $(JF)J^{\mathrm{T}}(a\nu) = 0$ also indicates that $J^{\mathrm{T}}(a\nu)$ is the null space of JF. It should be also noted that the permutation matrix J satisfies $JJ^{\mathrm{T}} = J^{\mathrm{T}}J = \mathrm{I}_{Nn}$; hence, $J^{\mathrm{T}}JFJ^{\mathrm{T}}(a\nu) = (J^{\mathrm{T}}J)F(J^{\mathrm{T}}(a\nu)) = F(J^{\mathrm{T}}(a\nu)) = 0$. Consequently, $J^{\mathrm{T}}(a\nu)$ is the null space of F and we can rewrite $J^{\mathrm{T}}(a\nu)$ as

$$\eta \triangleq J^{\mathrm{T}}(a\nu) = \begin{bmatrix} f(1) \\ \vdots \\ f(n) \\ \vdots \\ f(1) \\ \vdots \\ f(n) \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{Node 1} \\ \vdots \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{Node N} \end{matrix} = a \left(\mathbf{1}_N \otimes \begin{bmatrix} f(1) \\ \vdots \\ f(n) \end{bmatrix} \right) \triangleq a(\mathbf{1}_N \otimes \bar{f}). \tag{10.30}$$

Note that since F is a constant matrix, $\bar{f} \in \mathbb{R}^n$ is also a constant vector in this case. In addition, Theorem 1 indicates that $\delta(t)$ converges to $a(\mathbf{1}_N \otimes \bar{f})$. Recall that by definition $\delta(t) \triangleq x(t) - (\mathbf{1}_N \otimes \varepsilon)$, and thus

$$\lim_{t\to\infty} \left(x(t) - (\mathbf{1}_N \otimes \varepsilon) - a(\mathbf{1}_N \otimes \bar{f}) \right) \quad = \quad 0, \tag{10.31}$$

or equivalently,

$$\lim_{t\to\infty} \left(x(t) - (\mathbf{1}_N \otimes (\varepsilon + a\bar{f}) \right) \quad = \quad 0. \tag{10.32}$$

In general, (10.32) shows that under the proposed active-passive consensus filter given by (10.6) and (10.7) in subsection 10.3.1, all nodes reach the consensus in substate-wise.

Remark 10.3.4. *If a substate r, $r = 1, 2, \ldots, n$ is completely passive, the corresponding substate of ε takes a zero value owing to a corresponding zero row in the matrix S as discussed in Remark 10.3.2, and $f(r) = 1$ by (10.27). As a result, (10.30) and (10.32) indicate that a completely passive substate r is under the average consensus and finally converge to a consensus value. On the other hand, if there is at least one node i in*

the network that is active and has valid information on a substate r, $r = 1,2,\ldots,n$ (i.e., there is at least one positive $k_i m_{ir}$ for $i = 1,2,\ldots,N$), then $f(r) = 0$ and (10.30) and (10.32) suggest the substate converge to the average of all valid active corresponding substates in the network.

10.4 Active-Passive Consensus Filters with Time-varying Information Node Roles

10.4.1 Proposed Architecture

In this section, we extend the intermediate result in Section 10.3 to the actual practical case, where both the active-passive role of each node and the local observer vectors $z_i(t)$, $i = 1,2,\ldots,N$ are time-varying. However, owing to the properties of the overall time-varying system, we now consider that at any time moment t, for each substate of the process, there is at least one node that is active and has a valid information on that substate (that is, there is no completely passive substate at any time moment) as in Ref. [71]. For this purpose, we consider the proposed active-passive consensus filter given by

$$\dot{x}_i(t) = -\alpha \sum_{i\sim j}(x_i(t) - x_j(t)) + \alpha \sum_{i\sim j}(\xi_i(t) - \xi_j(t)) - \alpha k_i(t) M_i(x_i(t) - z_i(t)),\ x_i(0) = x_{i0}, \quad (10.33)$$

$$\dot{\xi}_i(t) = -\gamma\Big(\sum_{i\sim j}(x_i(t) - x_j(t)) + \sigma\xi_i(t)\Big),\ \xi_i(0) = \xi_{i0}, \quad (10.34)$$

where $x_i(t) \in \mathbb{R}^n$, $\xi_i(t) \in \mathbb{R}^n$, $z_i(t) \in \mathbb{R}^n$ denote the state, the integral action and the local estimate of node i, $i = 1,\ldots,N$ respectively. M_i is the value of information matrix defined in (10.5). Moreover, $\alpha, \gamma \in \mathbb{R}_+$ are constant consensus gains. Note that $k_i(t)$ in this section is time-varying and $k_i(t) \in [0,1]$. We further assume that each node can smoothly change back and forth between active and passive role (i.e., $k_i(t)$ is a smooth function on the interval $[0,1]$). We also note again the discussion in Remark 10.3.1 here.

10.4.2 Stability Analysis

Let $x(t) \triangleq \begin{bmatrix} x_1^{\mathrm{T}}(t) & x_2^{\mathrm{T}}(t) & \ldots & x_N^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}$, $\xi(t) \triangleq \begin{bmatrix} \xi_1^{\mathrm{T}}(t) & \xi_2^{\mathrm{T}}(t) & \ldots & \xi_N^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}$, and $\zeta(t) \triangleq \begin{bmatrix} z_1^{\mathrm{T}}(t) & z_2^{\mathrm{T}}(t) & \ldots & z_N^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}$. Similar to (10.8) and (10.9), the proposed algorithm (10.33) and (10.34) can be rewritten in the compact form as

$$\dot{x}(t) = -\alpha F(t)x(t) + \gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\xi(t) + \alpha\mathcal{M}(t)\zeta(t),\ x(0) = x_0, \quad (10.35)$$

$$\dot{\xi}(t) = -\gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)x(t) - \gamma\sigma\xi(t),\ \xi(0) = \xi_0, \quad (10.36)$$

where $\mathcal{M}(t) \triangleq \text{block–diag}_{i=1,2}\left(k_1(t)M_1, k_2(t)M_2, \ldots, k_N(t)M_N\right) \in \mathbb{R}^{Nn\times Nn}$, and $F(t) \triangleq \left(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n + \mathcal{M}(t)\right) \in \mathbb{R}^{Nn\times Nn}$. Since M_i for all $i = 1,2,\ldots,N$ are diagonal matrices, $\mathcal{M}(t)$ is a diagonal matrix for any $t \geq 0$.

Similar to (10.11), we define

$$\begin{aligned} S(t) &\triangleq (\mathbf{1}_N^{\mathrm{T}} \otimes \mathrm{I}_n)\mathcal{M}(t)(\mathbf{1}_N \otimes \mathrm{I}_n) \in \mathbb{R}^{n\times n} \\ &= k_1(t)M_1 + k_2(t)M_2 + \ldots + k_N(t)M_N, \end{aligned} \tag{10.37}$$

as a diagonal matrix that contains the total weight of all valid active substates of local observer vectors $z_i(t)$ on its diagonal. Note that since we assume at any time moment t, for each substate of the process, there is at least one node that is active and has a valid information on that substate, $S(t)$ is full rank as discussed in Remark 10.3.2; hence, it is invertible. We now let

$$\varepsilon(t) \triangleq S^{-1}(t)(\mathbf{1}_N^T \otimes \mathrm{I}_n)\mathcal{M}(t)\zeta(t) \in \mathbb{R}^n \tag{10.38}$$

be the dynamic average of all valid active substates of local observer vectors $z_i(t)$, $i = 1,2,\ldots,N$ in the network.

Next, we define the error as

$$\delta(t) \triangleq x(t) - (\mathbf{1}_N \otimes \varepsilon(t)) \in \mathbb{R}^{Nn}, \tag{10.39}$$

$$e(t) \triangleq \xi(t) - \frac{\alpha}{\gamma}(\mathcal{L}^+(\mathcal{G})\otimes \mathrm{I}_n)\mathcal{M}(t)\omega(t) \in \mathbb{R}^{Nn}, \tag{10.40}$$

where $\omega(t) \triangleq \left((\mathbf{1}_N \otimes \varepsilon(t)) - \zeta(t)\right) \in \mathbb{R}^{Nn}$. Similar to (10.15) and (10.14), by taking the time derivative of (10.39) and (10.40), we obtain

$$\dot{\delta}(t) = -\alpha F(t)\delta(t) + \gamma(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n)e(t) - \frac{\alpha}{N}(\mathbf{1}_N \otimes \mathrm{I}_n)(\mathbf{1}_N^{\mathrm{T}} \otimes \mathrm{I}_n)\mathcal{M}(t)\omega(t) - (\mathbf{1}_N \otimes \dot{\varepsilon}(t)), \quad \delta(0) = \delta_0, \tag{10.41}$$

$$\begin{aligned} \dot{e}(t) = &-\gamma(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n)\delta(t) - \gamma\sigma e(t) - \sigma\alpha(\mathcal{L}(\mathcal{G})^+ \otimes \mathrm{I}_n)\mathcal{M}(t)\omega(t) \\ &-\frac{\alpha}{\gamma}(\mathcal{L}(\mathcal{G})^+ \otimes \mathrm{I}_n)\left(\dot{\mathcal{M}}(t)\omega(t) + \mathcal{M}(t)\dot{\omega}(t)\right), \quad e(0) = e_0. \end{aligned} \tag{10.42}$$

By Lemma 10.3.1 and Remark 10.3.3, we can further reduce (10.41) to

$$\dot{\delta}(t) = -\alpha F(t)\delta(t) + \gamma(\mathcal{L}(\mathcal{G})\otimes \mathrm{I}_n)e(t) - (\mathbf{1}_N \otimes \dot{\varepsilon}(t)),\ \delta(0) = \delta_0. \tag{10.43}$$

Next, define

$$q_1(t) \triangleq -(\mathbf{1}_N \otimes \dot{\varepsilon}(t)), \tag{10.44}$$

$$q_2(t) \triangleq -\sigma\alpha(\mathcal{L}(\mathcal{G})^+ \otimes \mathrm{I}_n)\mathcal{M}(t)\omega(t) - \frac{\alpha}{\gamma}(\mathcal{L}(\mathcal{G})^+ \otimes \mathrm{I}_n)\big(\dot{\mathcal{M}}(t)\omega(t) + \mathcal{M}(t)\dot{\omega}(t)\big). \tag{10.45}$$

Since $k_i(t)$ is a smooth function on the interval $[0,1]$, $\mathcal{M}(t)$, $S(t)$, $S^+(t)$ and $\dot{\mathcal{M}}(t)$ are bounded. In addition, since $z_i(t)$ and $\dot{z}_i(t)$ are bounded by assumption, $\zeta(t)$ and $\dot{\zeta}(t)$ are bounded. Consequently, $\varepsilon(t)$, $\omega(t)$, $\dot{\varepsilon}(t)$, and $\dot{\omega}(t)$ are bounded. Therefore, $q_1(t)$ and $q_2(t)$ are bounded such as $\|q_1(t)\|_2 \leq q_1^*$ and $\|q_2(t)\|_2 \leq q_2^*$. (10.43) and (10.42) are now can be rewritten as

$$\dot{\delta}(t) = -\alpha F(t)\delta(t) + \gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)e(t) + q_1(t),\ \delta(0) = \delta_0. \tag{10.46}$$

$$\dot{e}(t) = -\gamma(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\delta(t) - \gamma\sigma e(t) + q_2(t),\ e(0) = e_0. \tag{10.47}$$

Theorem 10.4.1. *Consider a sensor network with N nodes given by (10.33) and (10.34), where nodes exchange information using local measurements under a connected, undirected graph $\mathcal{G}$. Then, the closed-loop error dynamics (10.46) and (10.47) are uniformly ultimately bounded.*

Proof. Consider the Lyapunov function candidate given by (10.20). By taking time derivative of (10.20) along the trajectories of (10.46) and (10.47), we obtain

$$\begin{aligned}
\dot{V}(\delta(t), e(t)) &= -\alpha\delta^{\mathrm{T}}(t)F(t)\delta(t) + \gamma\delta^{\mathrm{T}}(t)(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)e(t) + \delta(t)^{\mathrm{T}}q_1(t) - \gamma e^{\mathrm{T}}(t)(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\delta(t) \\
&\quad -\gamma\sigma e^{\mathrm{T}}(t)e(t) + e^{\mathrm{T}}(t)q_2(t) \\
&= -\alpha\delta^{\mathrm{T}}(t)F(t)\delta(t) - \gamma\sigma e^{\mathrm{T}}(t)e(t) + \delta^{\mathrm{T}}(t)q_1(t) + e^{\mathrm{T}}(t)q_2(t) \\
&\leq -\alpha\delta^{\mathrm{T}}(t)F(t)\delta(t) + \delta^{\mathrm{T}}(t)q_1(t) - \gamma\sigma\|e(t)\|_2^2 + \|e(t)\|_2 q_2^* \\
&\leq -\alpha\delta^{\mathrm{T}}(t)F(t)\delta(t) + \delta^{\mathrm{T}}(t)q_1(t) - \gamma\sigma\|e(t)\|_2\big(\|e(t)\|_2 - \phi_e\big),
\end{aligned} \tag{10.48}$$

where $\phi_e \triangleq \frac{q_2^*}{\gamma\sigma}$. Let

$$\mathcal{H} \triangleq -\alpha\delta^{\mathrm{T}}(t)F(t)\delta(t) + \delta^{\mathrm{T}}(t)q_1(t), \tag{10.49}$$

and

$$\psi(t) \triangleq J\delta(t) = J\begin{bmatrix} \delta_{11}(t) \\ \vdots \\ \delta_{1n}(t) \\ \vdots \\ \delta_{N1}(t) \\ \vdots \\ \delta_{Nn}(t) \end{bmatrix} = \begin{bmatrix} \delta_{11}(t) \\ \vdots \\ \delta_{N1}(t) \\ \vdots \\ \delta_{1n}(t) \\ \vdots \\ \delta_{Nn}(t) \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \psi_1(t) \\ \vdots \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \psi_n(t) \end{matrix} = \begin{bmatrix} \psi_1(t) \\ \vdots \\ \psi_n(t) \end{bmatrix}, \tag{10.50}$$

where J is the same permutation matrix discussed in subsection 10.3.3 and $\delta_{ij}(t)$ indicates the error of substate j, $j = 1,2,\ldots,n$ of node i, $i = 1,2,\ldots,N$. Note that

$$\psi^{\mathrm{T}}(t)(JFJ^{\mathrm{T}})\psi(t) = \delta^{\mathrm{T}}(t)J^{\mathrm{T}}(JFJ^{\mathrm{T}})J\delta(t) = \delta^{\mathrm{T}}(t)F\delta(t). \tag{10.51}$$

Utilize (10.51) and (10.44), (10.49) can be rewritten as

$$\mathcal{H} = -\alpha\psi^{\mathrm{T}}(t)(JFJ^{\mathrm{T}})\psi(t) - \psi(t)J(\mathbf{1}_N \otimes \dot{\varepsilon}(t)). \tag{10.52}$$

Note also that

$$J(\mathbf{1}_N \otimes \dot{\varepsilon}(t)) = J\begin{bmatrix} \dot{\varepsilon}(t) \\ \vdots \\ \dot{\varepsilon}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{1}_N\dot{\varepsilon}_1(t) \\ \vdots \\ \mathbf{1}_N\dot{\varepsilon}_n(t) \end{bmatrix}, \tag{10.53}$$

where $\dot{\varepsilon}_j(t)$ for $j = 1,2,\ldots,n$ is the j-th substate of $\varepsilon(t)$. By assuming that at any time moment t, for each substate of the process, there is at least one node that is active and has a valid information on that substate, the structure of $JF(t)J^{\mathrm{T}}$ in (10.26) shows us that for all $r = 1,2,\ldots,n$, $F_r(t)$ can be rewritten as

$$F_r(t) = \begin{bmatrix} \mathcal{L}_{11} + k_1(t)m_{1r} & \cdots & \mathcal{L}_{1N} \\ \vdots & \ddots & \vdots \\ \mathcal{L}_{N1} & \cdots & \mathcal{L}_{NN} + k_N(t)m_{Nr} \end{bmatrix} = \mathcal{L}(\mathcal{G}) + K_r(t) \tag{10.54}$$

where $K_r(t) = \text{diag}(\begin{bmatrix} k_1(t)m_{1r} & \dots & k_N(t)m_{Nr} \end{bmatrix}^{\mathrm{T}})$ with at least one of the element on the diagonal $\beta_i \triangleq k_i(t)m_{ir} \in \mathbb{R}_+$ for some $i \in \mathcal{V}_\mathcal{G}$. Thus, we can write $K_r(t) = K_{r0} + K_{r1}(t)$ where $K_{r0} \triangleq \text{diag}(\begin{bmatrix} 0 & \dots & 0 & \beta_i & 0 & \dots & 0 \end{bmatrix}^{\mathrm{T}})$, and $K_{r1} \triangleq K_r(t) - K_{r0}$ is also a diagonal matrix with nonnegative elements on the diagonal. As a result, we have

$$F_r(t) = \mathcal{L}(\mathcal{G}) + K_{r0} + K_{r1}(t) = F_{r0} + K_{r1}(t), \tag{10.55}$$

where $F_{r0} \triangleq \mathcal{L}(\mathcal{G}) + K_{r0}$ is a positive definite matrix by, for example, Lemma 2 in Ref. [102] or Lemma 3.3 in Ref. [16]. We now can write

$$JF(t)J^{\mathrm{T}} = F_0 + \mathcal{M}_0(t), \tag{10.56}$$

where $F_0 \triangleq \text{block–diag}_{i=1,2}(F_{10}, F_{20}, \dots, F_{n0})$ is a positive definite matrix and $\mathcal{M}_0(t) \triangleq \text{block–diag}_{i=1,2}(K_{11}(t), K_{21}(t), \dots, K_{n1}(t))$ is a diagonal matrix with nonnegative elements on the diagonal. From (10.56), (10.52) now becomes

$$\begin{aligned} \mathcal{H} &= -\alpha\psi^{\mathrm{T}}(t)\big(F_0 + \mathcal{M}_0(t)\big)\psi(t) - \psi(t)J(\mathbf{1}_N \otimes \dot{\varepsilon}(t)) \\ &\leq -\alpha\psi^{\mathrm{T}}(t)F_0\psi(t) - \psi(t)J(\mathbf{1}_N \otimes \dot{\varepsilon}(t)) \\ &\leq -\alpha\lambda_{\min}(F_0)\|\psi(t)\|_2^2 + \|\psi(t)\|_2 q_1^* \\ &\leq -\alpha\lambda_{\min}(F_0)\|\psi(t)\|_2(\|\psi(t)\|_2 - \phi_\delta), \end{aligned} \tag{10.57}$$

where $\phi_\delta = \frac{q_1^*}{\alpha\lambda_{\min}(F_0)}$ with $\|J(\mathbf{1}_N \otimes \dot{\varepsilon}(t))\|_2 = \|(\mathbf{1}_N \otimes \dot{\varepsilon}(t))\|_2 \leq q_1^*$. We note here again that $\psi(t) \triangleq J\delta(t)$; that is, $\psi(t)$ is a permutation of $\delta(t)$, and since a vector norm is preserved under permutation, $\|\psi(t)\|_2 = \|\delta(t)\|_2$. Therefore, $\mathcal{H} \leq 0$ outside the compact set $\Omega_\delta \triangleq \{(\delta(t), e(t)) : \|\psi(t)\|_2 \leq \phi_\delta\} = \{(\delta(t), e(t)) : \|\delta(t)\|_2 \leq \phi_\delta\}$. By combining the result of (10.48) and (10.57), we have $\dot{V}(\delta(t), e(t)) \leq 0$ outside the compact set given by

$$\Omega \triangleq \{(\delta(t), e(t)) : \|\delta(t)\|_2 \leq \phi_\delta\} \cap \{(\delta(t), e(t)) : \|e(t)\|_2 \leq \phi_e\}. \tag{10.58}$$

Consequently, the closed-loop error dynamics given by (10.46) and (10.47) are uniformly bounded. ■

The following corollary to the above theorem is immediate.

Corollary 10.4.1. *Consider a sensor network with N nodes given by (10.33) and (10.34), where nodes exchange information using local measurements under a connected, undirected graph $\mathcal{G}$. Then, the ultimately bound of $\delta(t)$ for $t \geq T$ is given by*

$$\|\delta(t)\|_2^2 \leq \frac{(q_1^*)^2}{\alpha^2 \lambda_{\min}(F_0)^2} + \frac{(q_2^*)^2}{\gamma^2 \sigma^2}, \tag{10.59}$$

where q_1^* and q_2^* are the upper bounds of $\|q_1(t)\|_2^2$ and $\|q_2(t)\|_2^2$ defined in (10.44) and (10.45).

Proof. From the proof of Theorem 10.4.1, we have $\dot{V}(\delta(t), e(t)) \leq 0$ outside the compact set Ω given by (10.58). Therefore, the evolution of $V(\delta(t), e(t))$ is upper bounded by

$$V(\delta(t), e(t)) \leq \max_{(\delta(t), e(t)) \in \Omega} V(\delta(t), e(t)) = \frac{1}{2}(\phi_\delta^2 + \phi_e^2). \tag{10.60}$$

Note that $\frac{1}{2}\delta^{\mathrm{T}}(t)\delta(t) \leq V(\delta(t), e(t))$, thus (10.59) is immediate ■

Remark 10.4.1. *It should be noted that by definition of $q_2(t)$ in (10.45), the upper bound q_2^* can be rewritten as $q_2^* = \alpha q_3^*$. As a result, Corollary 10.4.1 indicates if the gains α, γ, and σ are chosen properly such that $\frac{1}{\alpha^2}$ and $\frac{\alpha^2}{\gamma^2 \sigma^2}$ are small, then the ultimate bound (10.59) of $\delta(t)$ is small when $t \geq T$; and hence, the overall performance of the sensor network can be improved.*

10.5 Information Validity Monitor Layer

In this section, we present a dynamic average consensus that is parallel to the active-passive consensus filter in order to monitor the validity of the information (see Figure 10.1). For this purpose, consider the dynamics given by

$$\dot{q}_i(t) = -\beta \sum_{i \sim j} \big(q_i(t) - q_j(t)\big) + \beta \sum_{i \sim j} \big(r_i(t) - r_j(t)\big) - \beta \big(q_i(t) - h_i(t)\big), \quad q_i(0) = q_{i0}, \tag{10.61}$$

$$\dot{r}_i(t) = -\mu \sum_{i \sim j} \big(q_i(t) - q_j(t)\big), \quad r_i(0) = r_{i0}, \tag{10.62}$$

where $q_i(t) \in \mathbb{R}^n$ denotes the information validity vector for node i, $i = 1, 2, \ldots, N$, $h_i(t) \triangleq k_i(t) m_i \in \mathbb{R}^n$ with m_i is the diagonal of the value of information matrix M_i, and $\beta, \mu \in \mathbb{R}_+$ denote the gains.

Note that the structure of (10.61) and (10.62) is a special case of (10.6) and (10.7), where it becomes a dynamic average consensus, for which a proof can be found in, for example [170] and [61] and references therein. Therefore, $q_i(t)$ is tracking the neighborhood of the dynamic average $\bar{h}(t) \triangleq \frac{\mathbf{1}\mathbf{1}^{\mathrm{T}}}{N} h(t) \in \mathbb{R}^n$ with $h(t) = \begin{bmatrix} h_1^{\mathrm{T}}(t) & h_2^{\mathrm{T}}(t) & \dots & h_N^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}$. Since $\bar{h}(t)$ is the dynamic average of the $k_i(t)m_i$ for all $i = 1,2,\dots,N$, $q_i(t)$ provides us the information on the average of active weights of the whole network at each time moment for each substate of $z_i(t)$, $i = 1,2,\dots,N$. Therefore, $q_i(t)$ can be considered as a confidence factor to check the reliability of the information. For example, the value of a substate of $q_i(t)$ increases as the number of valid active corresponding substates of $z_i(t)$ for all $i = 1,2,\dots,N$ in the whole network increases; that is, the higher the value of $q_i(t)$ is, the more reliable the information is. This point would become more apparent as illustrated in examples of Section 10.6.

10.6 Discussion and Examples

In this section, we present several numerical examples to illustrate the results given in previous section. For this purpose, consider a process composed of two subprocesses with the dynamics given by (10.1), where

$$A = \underset{i=1,2}{\text{block–diag}}\left(A_1, A_2\right) = \begin{bmatrix} 0.0150 & 0 & 0 & 0 \\ 0 & -0.0250 & 0 & 0 \\ 0 & 0 & -0.0005 & 0.1000 \\ 0 & 0 & -0.2500 & 0 \end{bmatrix}. \tag{10.63}$$

with $A_1 \triangleq \begin{bmatrix} 0.0150 & 0 \\ 0 & -0.0250 \end{bmatrix}$ and $A_2 \triangleq \begin{bmatrix} -0.0005 & 0.1000 \\ -0.2500 & 0 \end{bmatrix}$. We consider a sensor network with 4 nodes exchange information among each other using their local measurements according to a connected, undirected ring graph. Each node's sensing capability is represented by (10.3) with the output matrices

$$^{I}C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \tag{10.64}$$

$$^{III}C_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}, \tag{10.65}$$

$$^{I}C_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}, \tag{10.66}$$

$$^{I}C_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}, \tag{10.67}$$

and hence, for the local observers ${}^1\bar{C}_1 = [1\ 0]$, ${}^1\bar{C}_2 = [0\ 1]$ corresponding to A_1 and ${}^2\bar{C}_3 = [1\ 0]$, ${}^2\bar{C}_4 = [0\ 1]$ corresponding to A_2. Note that sensors 1, 3, and 4 are Category I sensors since the pair $(A_1, {}^1\bar{C}_1)$ is detectable, and the pairs $(A_2, {}^2\bar{C}_3)$ and $(A_2, {}^2\bar{C}_4)$ are observable. On the other hand, the pair $(A_1, {}^1\bar{C}_2)$ is unobservable, so sensor 2 is a Category III sensor. As a result, the observer structure (10.4) is only applied to sensors 1, 3, and 4. For example,

$$\dot{s}_1(t) = A_1 s_1(t) + {}^1L_1(y_1 - {}^1\bar{C}_1 s_1(t)), \tag{10.68}$$

where $s_1(t) \in \mathbb{R}^2$ and ${}^1L_1 = \begin{bmatrix} 2.0302 \\ 0 \end{bmatrix}$. Similarly, for sensors 3 and 4, $s_3(t), s_4(t) \in \mathbb{R}^2$ and we choose

$$ {}^2L_3 = \begin{bmatrix} 5.3737 \\ 3.9039 \end{bmatrix}, \tag{10.69}$$

$$ {}^2L_4 = \begin{bmatrix} -1.8052 \\ 2.4094 \end{bmatrix}. \tag{10.70}$$

As discussed in Section 10.2.2, after using the local observers to estimate the states of the subprocesses, we are now able to construct $z_i(t) \in \mathbb{R}^4$ such that

$$ z_1(t) = \begin{bmatrix} s_1^{\mathrm{T}}(t) & 0 & 0 \end{bmatrix}^{\mathrm{T}}, \tag{10.71}$$

$$ z_2(t) = \begin{bmatrix} 0 & y_2(t) & 0 & 0 \end{bmatrix}^{\mathrm{T}}, \tag{10.72}$$

$$ z_3(t) = \begin{bmatrix} 0 & 0 & s_3^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}, \tag{10.73}$$

$$ z_4(t) = \begin{bmatrix} 0 & 0 & s_4^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}. \tag{10.74}$$

Next, we define the value of information matrix for each node. Since the pair $(A_1, {}^1\bar{C}_1)$ is detectable, sensor 1 can observe the first substate of the process, yet the estimation of second substate of the process is not so well. Thus, we can choose $M_1 = \mathrm{diag}\left(\begin{bmatrix} 2 & 0.5 & 0 & 0 \end{bmatrix}\right)$. In the same manner, we choose $M_2 = \mathrm{diag}\left(\begin{bmatrix} 0 & 2 & 0 & 0 \end{bmatrix}\right)$, $M_3 = \mathrm{diag}\left(\begin{bmatrix} 0 & 0 & 2 & 1 \end{bmatrix}\right)$ and $M_4 = \mathrm{diag}\left(\begin{bmatrix} 0 & 0 & 1 & 2 \end{bmatrix}\right)$.

In addition, all nodes are subject to random initial conditions and we set $\alpha = 15$, $\gamma = 10$, $\sigma = 0.1$, $\beta = 10$ and $\mu = 5$. Furthermore, for the transition of $k_i(t)$, we use the function $k_i(t) = e^{-\theta t}$ when node i is switching from 1 to 0, and $k_i(t) = 1 - e^{-\theta t}$ when node i is switching from 0 to 1, where $\theta = 5$. The simulations are run for 100 second. All below examples share the same setup but are different in node roles over time.

10.6.1 Example 1

In this example, the role of each node are fixed over time. Specifically, nodes 1, 2, and 3 are active (i.e, $k_1(t) = k_2(t) = k_3(t) = 1$) while node 4 is passive (i.e, $k_4(t) = 0$). Note that this information node role setup satisfies the assumption in Section 10.4 that for each process's substate, there is at least one node that is active and has a valid information on that substate. Figure 10.2 shows the performance of the sensor networks under the proposed active-passive consensus filter given by (10.33) and (10.34), where nodes quickly converge to consensus and are able to closely estimate the process states. In addition, Figure 10.3 shows the result of the information validity monitor layer given by (10.61) and (10.62). Since the information node role setup in this example is fixed, the information validity vectors converge to constants. As can be seen, for example, in the bottom plot of Figure 10.3, the information validity vector of the process' fourth substate converges to a low value due to the fact that node 4, which can directly senses this substate, is passive and the information in this substate is obtained only from the local observer of node 3.

10.6.2 Example 2

In this example, the role of each node varies over time. For the first 25 seconds, nodes 1 and 4 are active (i.e, $k_1(t) = k_4(t) = 1$ and $k_2(t) = k_3(t) = 0$); for $t \in (25, 50]$, nodes 1 and 3 are active; for $t \in (50, 75]$, all 4 nodes are active; for $t \in (75, 100]$, nodes 1, 2 and 3 are active. This information node role configuration satisfies the assumption in Section 10.4 that for each process's substate, there is at least one node that is active and has a valid information on that substate. Figure 10.4 shows the performance of the sensor networks under the proposed active-passive consensus filter given by (10.33) and (10.34), where nodes quickly converge to consensus and are able to closely estimate the process states. In addition, Figure 10.5 shows the result of the information validity monitor layer given by (10.61) and (10.62). Since the information node roles change with respect to time in this example, the information validity vectors converge to different values over time. For example, for the first 50 seconds, the value of the information

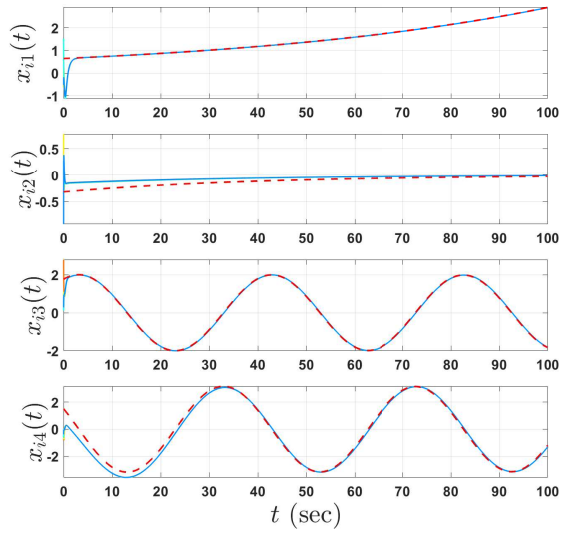


Figure 10.2: State estimates of the sensor network with four nodes in a connected, undirected ring graph in Example 1 under the proposed active-passive consensus filter architecture (10.33) and (10.34) (the dash lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

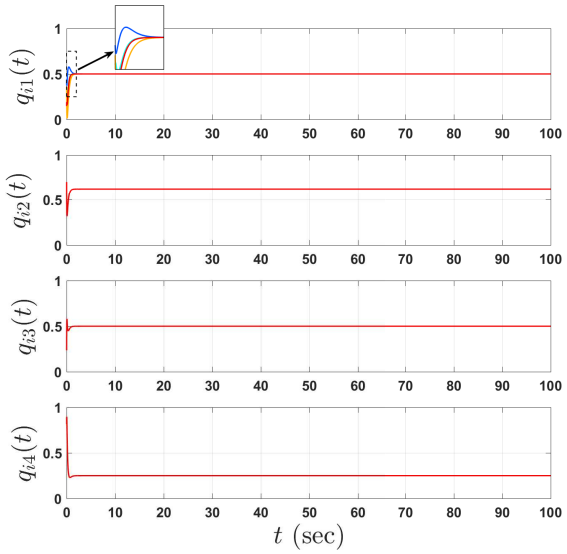


Figure 10.3: The evolution of information validity vector of the sensor network with four nodes in a connected, undirected ring graph in Example 1 under the monitor layer given by (10.61) and (10.62).

validity vectors $q_{i2}(t)$ on the second substate of the process is small since the information on this substate is obtained from the local observer of node 1, which has the weight of 0.5 only. For the last 50 seconds, the value of $q_{i2}(t)$ has increased since node 2 becomes active and adds more validity on the information.

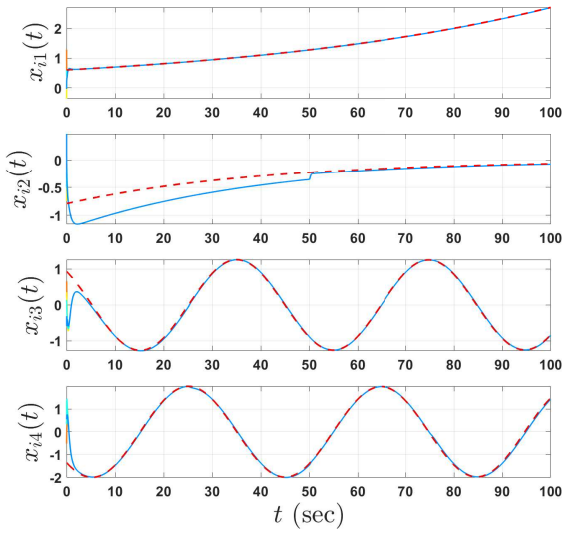


Figure 10.4: State estimates of the sensor network with four nodes in a connected, undirected ring graph in Example 2 under the proposed active-passive consensus filter architecture (10.33) and (10.34) (the dash lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

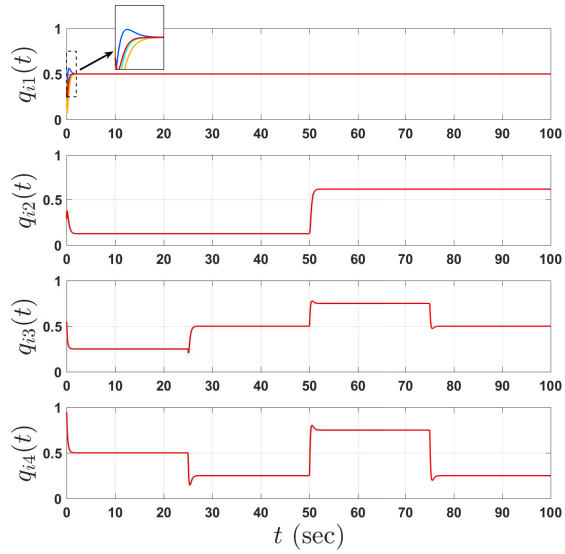


Figure 10.5: The evolution of information validity vector of the sensor network with four nodes in a connected, undirected ring graph in Example 2 under the monitor layer given by (10.61) and (10.62).

10.6.3 Example 3

In this example, the role of each node varies over time such that for the first 25 second, nodes 2 and 4 are active (i.e, $k_2(t) = k_4(t) = 1$ and $k_1(t) = k_3(t) = 0$); for $t \in (25, 50]$, nodes 3 and 4 are active; for $t \in (50, 75]$, all 4 nodes are active; for $t \in (75, 100]$, nodes 1 and 2 are active. This information node role

configuration indeed violates the assumption in Section 10.4 that for each process's substate, there is at least one node that is active and has a valid information on that substate. In fact, this configuration allows some substates becomes completely passive at some time instants, and it can be another important practical case. For example, if the process of interest represents multiple targets with each subprocess corresponding to a target, then at any time instant, a target can or can not be observed by the sensor network.

Figure 10.6 shows the performance of the sensor networks under the proposed active-passive consensus filter given by (10.33) and (10.34), where nodes quickly converge to consensus and are able to closely estimate the actual value if that particular process substate can be observed by at least one node (i.e., at least one node is active and has valid information on that substate). By utilizing the information validity monitor layer given by (10.61) and (10.62), one can monitor if the information of a substate is valid or not (that is, if $q_{ij}(t) = 0$, then the substate j is completely passive, and thus the information is invalid and not reliable) as shown in Figure 10.7. It can be seen that, for example, during the first 50 seconds the first substate of the process is not observable (i.e., completely passive), and hence the information from the sensors on this substate is not valid. Another example is that during the last 25 seconds, the third and fourth substates of the process are completely passive (see Figure 10.7) and the corresponding substates obtained from the sensor network are constants during this time period as seen in Figure 10.6. As discussed in Remark 10.3.4 of Section 10.3, completely passive substates still result in nodes reaching consensus, yet the information is invalid. Note that at time $t = 75$ seconds, the nodes' estimates already reached the consensus, thus when these substates becomes completely passive, the sensors retains the last values they sense from the process until at least one of the sensor becomes active and has valid information on these substates.

Intuitively, from the analysis in Section 10.3, we expect that when extending the architecture (10.6) and (10.7) to the time-varying case, that is, the proposed active-passive consensus filter in Section 10.4 given by (10.33) and (10.34), the sensor networks can converge to the null space of $F(t)$. However, without the assumption in Section 10.4 (that is, for each process's substate, there is at least one node that is active and has a valid information on that substate), at each time instant $F(t)$ can be either positive definite or nonnegative definite, and hence, the null space of $F(t)$ is time-varying as well. Utilizing Lyapunov analysis in this case is a good and challenging future research direction to the authors.

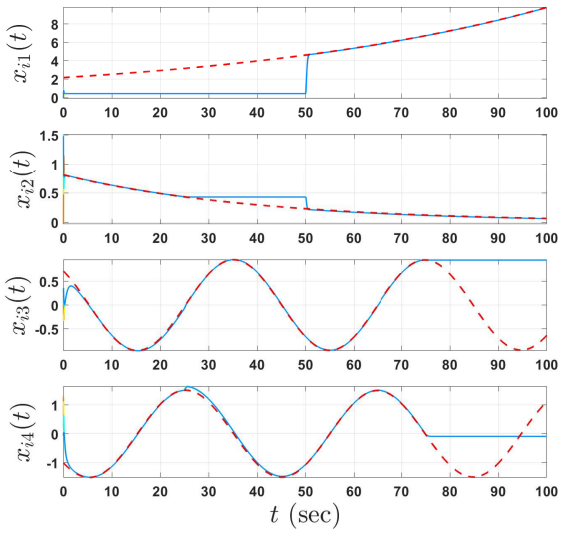


Figure 10.6: State estimates of the sensor network with four nodes in a connected, undirected ring graph in Example 3 under the proposed active-passive consensus filter architecture (10.33) and (10.34) (the dash lines denote the states of the actual process and the solid lines denote the state estimates of nodes).

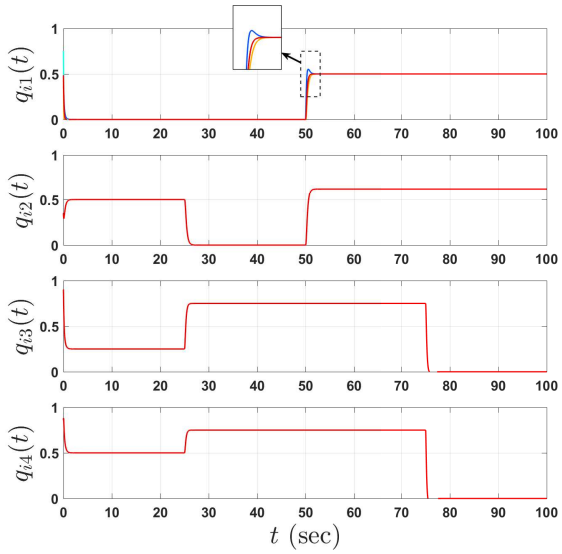


Figure 10.7: The evolution of information validity vector of the sensor network with four nodes in a connected, undirected ring graph in Example 3 under the monitor layer given by (10.61) and (10.62).

10.7 Conclusion

This paper contributed to the previous studies in heterogeneous sensor networks through proposing a dynamic information fusion framework for sensor networks with the integration of local observers, value of information, active-passive consensus filters, and a layer to monitor the validity of information. The proposed framework considered a process of interest with multiple subprocesses and the sensor network

that allows nodes with heterogeneous modalities, heterogeneous information node roles, and heterogeneous quality of information. Furthermore, the extra layer allows operators to evaluate the reliability of the fused information based on the local feedbacks received from the sensor network. In addition to the presented theoretical algorithms, illustrative examples had shown the efficacy of the proposed structure and prompted a discussion on the practical aspects when relaxing some certain assumptions. Recommended future studies of the algorithm presented in this paper may include extensions to agents with switching graph topologies as well as experimental verification.

Chapter 11: Concluding Remarks and Future Research

11.1 Concluding Remarks

This dissertation's purpose has been to introduce novel tools and methods for allowing spatiotemporal control of multiagent systems as well as dynamic information fusion in heterogeneous sensor networks. Specifically, the proposed distributed control architectures allow spatial and temporal properties of multiagent systems to be changed in both real-time and a decentralized manner. In addition, the proposed distributed estimation architectures allow for improved performance in dynamic information fusion of sensor networks, where heterogeneities coming from differences in sensor modalities, quality of sensing information (i.e., the value of information) and information roles of nodes (i.e., active and passive) were all taken into account.

To address the challenge of changing the spatial and temporal properties of multiagent systems, Chapter 2 proposed a distributed control architecture predicated on multiplex information networks and focused on the application in formation control with scalar integrator dynamics. The key feature of this distributed control architecture is that the desired parameters (e.g., the desired scaling factors, rotation angle) are distributedly spread over the network in a decentralized manner through updates in the secondary layer and the updates on these parameters directly influence the main layer leading to the change in spatial and temporal properties of the multiagent systems. In addition, numerical examples as well as experiments on ground mobile robots (Qbots) to justify the efficacy of the proposed structure are also presented. The multiplex information-based architecture was then generalized to higher-order dynamics in Chapter 3. Chapter 4 showed that the bandwidth of multiagent systems can be also manipulated through multiplex information networks.

The new Laplacian matrix was introduced in Chapter 5, which allows the user to assign a desired nullspace for the network. This new Laplacian matrix was defined and showed that it inherits the fundamental properties of the standard Laplacian matrix. In addition, the structure of this new matrix not only

explains the interactions between agents in the network better but also allows more complex cooperative behaviors in multiagent systems.

New time-critical control structures were proposed in Chapters 6 and 7. Concretely, Chapter 6 considered networks as systems. Finite-time algorithms were proposed and analyzed based on the separation principle and time transformation method. The practical considerations were also discussed and experiments were presented to justify the proposed algorithms. Chapter 7 then introduced a new class of scalar, time-varying gain function to convert a baseline control algorithm into a time-varying one for time-critical applications. The time transformation was then used to simplify the stability analysis of the system. The key feature of the structures proposed in these two chapters is that the convergence time can be assigned arbitrarily by the users.

Finally, the dynamic information fusion architectures for heterogeneous sensor networks were studied and analyzed in Chapters 8, 9 and 10. Specifically, Chapter 8 introduced a preliminary structure for dynamic information fusion in heterogeneous sensor networks. Chapter 9 then provided some extensions for improving the performance and simplifies the tunning procedure of the structure proposed in Chapter 8. Chapter 10 proposes another dynamic information fusion framework with the integration of local observers, value of information, and active-passive consensus filter as well as a layer to monitor the validity of the information. The key feature of these structures is that they only utilize local information at both execution and design stages. In addition, the stability of these proposed architectures was theoretically analyzed, then justified through numerical examples.

11.2 Future Research

There are many future research directions for the proposed tools and methods in this dissertation. Specifically, regarding the multiplex information-based architecture, we can further investigate theoretical developments and applications for controlling the spatial and temporal properties of a group of the heterogeneous ground and aerial robots with exogenous disturbances, system uncertainties, and communication constraints. Next, algorithms predicated on the new Laplacian matrix can be investigated toward composing complex cooperative behaviors in multiagent systems through nullspace assignment and control. For our current finite-time control framework, we assume the system will stop after the convergence at the user-assigned time τ. Further investigation on algorithms for keeping the system at the equilibrium point after τ can be considered. Finally, the dynamic information fusion frameworks presented in this dissertation can be

useful for several future research directions for dynamic data-driven applications including but not limited to: i) Consider the uncertainty and disturbances in the process of interest; ii) nonlinear process; iii) tools and methods from recent event-triggered control theory developments can be utilized in order to reduce the communication cost between sensor nodes; iv) for capturing cases when the process of interest leaves the sensing field of the network for certain time periods, tools and methods from filtering and estimation theories can be utilized with the presented results of this paper; v) the presented setup can be extended to involve mobile sensor nodes; and vi) the presented results can be applied to real-world sensor networks through experiments in order to bridge the gap between theory and practice.

References

[1] D. Tran, T. Yucelen, S. B. Sarsilmaz, and S. Jagannathan, "Distributed input and state estimation using local information in heterogeneous sensor networks," *Frontiers in Robotics and AI*, vol. 4, p. 30, 2017.

[2] P. Millán, L. Orihuela, C. Vivas, F. Rubio, D. V. Dimarogonas, and K. H. Johansson, "Sensor-network-based robust distributed control and estimation," *Control Engineering Practice*, vol. 21, no. 9, pp. 1238–1249, 2013.

[3] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *IEEE Conference on Decision and Control*, pp. 5492–5498, 2007.

[4] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008.

[5] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[6] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[7] M. B. Egerstedt and X. Hu, "Formation constrained multi-agent control," 2001.

[8] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 324–333, 2008.

[9] Z. Peng, S. Yang, G. Wen, A. Rahmani, and Y. Yu, "Adaptive distributed formation control for multiple nonholonomic wheeled mobile robots," *Neurocomputing*, vol. 173, pp. 1485–1494, 2016.

[10] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community structure in time-dependent, multiscale, and multiplex networks," *Science*, vol. 328, no. 5980, pp. 876–878, 2010.

[11] J. Gómez-Gardenes, I. Reinares, A. Arenas, and L. M. Floría, "Evolution of cooperation in multiplex networks," *Scientific reports*, 2012.

[12] A. Sole-Ribalta, M. De Domenico, N. E. Kouvaris, A. Diaz-Guilera, S. Gomez, and A. Arenas, "Spectral properties of the laplacian of multiplex networks," *Physical Review E*, vol. 88, no. 3, p. 032807, 2013.

[13] M. Rebollo, E. del Val, C. Carrascosa, A. Palomares, and F. Pedroche, "Consensus over multiplex network to calculate user influence in social networks," *Int. J. Complex Systems in Science*, vol. 3, no. 1, pp. 71–75, 2013.

[14] C. Granell, S. Gómez, and A. Arenas, "Dynamical interplay between awareness and epidemic spreading in multiplex networks," *Physical review letters*, vol. 111, no. 12, p. 128701, 2013.

[15] W. Ren and R. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*. Springer, 2007.

[16] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative control of multi-agent systems: Optimal and adaptive design approaches*. Springer Science & Business Media, 2014.

[17] M. Mesbahi and F. Y. Hadaegh, "Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 369–377, 2001.

[18] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE transactions on robotics and automation*, vol. 18, no. 5, pp. 813–825, 2002.

[19] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[20] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic control*, vol. 52, no. 5, pp. 863–868, 2007.

[21] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, 2008.

[22] M. M. Zavlanos, H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Hybrid control for connectivity preserving flocking," *Transactions on Automatic Control*, vol. 54, no. 12, pp. 2869–2875, 2009.

[23] Z. Li, G. Wen, Z. Duan, and W. Ren, "Designing fully distributed consensus protocols for linear multi-agent systems with directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 1152–1157, 2015.

[24] S. Su and Z. Lin, "Distributed consensus control of multi-agent systems with higher order agent dynamics and dynamically changing directed interaction topologies," *IEEE Transactions on Automatic Control*, vol. 61, no. 2, pp. 515–519, 2016.

[25] X.-K. Liu, Y.-W. Wang, J.-W. Xiao, and W. Yang, "Distributed hierarchical control design of coupled heterogeneous linear systems under switching networks," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 8, pp. 1242–1259, 2017.

[26] S. P. Bhat and D. S. Bernstein, "Continuous finite-time stabilization of the translational and rotational double integrators," *IEEE Transactions on automatic control*, vol. 43, no. 5, pp. 678–682, 1998.

[27] S. P. Bhat and D. S. Bernstein, "Finite-time stability of continuous autonomous systems," *SIAM Journal on Control and Optimization*, vol. 38, no. 3, pp. 751–766, 2000.

[28] J. Cortés, "Finite-time convergent gradient flows with applications to network consensus," *Automatica*, vol. 42, no. 11, pp. 1993–2000, 2006.

[29] Q. Hui, W. M. Haddad, and S. P. Bhat, "Finite-time semistability and consensus for nonlinear dynamical networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1887–1900, 2008.

[30] F. Xiao, L. Wang, J. Chen, and Y. Gao, "Finite-time formation control for multi-agent systems," *Automatica*, vol. 45, no. 11, pp. 2605–2611, 2009.

[31] F. Jiang and L. Wang, "Finite-time information consensus for multi-agent systems with fixed and switching topologies," *Physica D: Nonlinear Phenomena*, vol. 238, no. 16, pp. 1550–1560, 2009.

[32] L. Wang and F. Xiao, "Finite-time consensus problems for networks of dynamic agents," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 950–955, 2010.

[33] M. Basin, Y. Shtessel, and F. Aldukali, "Continuous finite-and fixed-time high-order regulators," *Journal of the Franklin Institute*, vol. 353, no. 18, pp. 5001–5012, 2016.

[34] H. Hong, W. Yu, G. Wen, and X. Yu, "Distributed robust fixed-time consensus for nonlinear and disturbed multiagent systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1464–1473, 2017.

[35] B. Tian, Z. Zuo, and H. Wang, "Leader–follower fixed-time consensus of multi-agent systems with high-order integrator dynamics," *International Journal of Control*, vol. 90, no. 7, pp. 1420–1427, 2017.

[36] W. Lu, X. Liu, and T. Chen, "A note on finite-time and fixed-time stability," *Neural Networks*, vol. 81, pp. 11–15, 2016.

[37] H. B. Oza, Y. V. Orlov, and S. K. Spurgeon, "Robust finite time stability and stabilisation: A survey of continuous and discontinuous paradigms," in *2014 International Workshop on Variable Structure Systems*, pp. 1–7, IEEE, 2014.

[38] A. Polyakov, "Nonlinear feedback design for fixed-time stabilization of linear control systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 8, pp. 2106–2110, 2012.

[39] H. Ríos and A. R. Teel, "A hybrid observer for fixed-time state estimation of linear systems," in *2016 Conference on Decision and Control*, pp. 5408–5413, IEEE, 2016.

[40] T. Yucelen, Z. Kan, and E. Pasiliao, "Finite-time cooperative engagement," *IEEE Transactions on Automatic Control*, 2018.

[41] Z. Kan, T. Yucelen, E. Doucette, and E. Pasiliao, "A finite-time consensus framework over time-varying graph topologies with temporal constraints," *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 7, p. 071012, 2017.

[42] E. Arabi, T. Yucelen, and J. R. Singler, "Robustness of finite-time distributed control algorithm with time transformation," in *American Control Conference*, 2019.

[43] J. D. Sánchez-Torres, E. N. Sanchez, and A. G. Loukianov, "Predefined-time stability of dynamical systems with sliding modes," in *2015 American Control Conference*, pp. 5842–5846, IEEE, 2015.

[44] E. Jiménez-Rodríguez, J. D. Sánchez-Torres, D. Gómez-Gutiérrez, and A. G. Loukianov, "Predefined-time tracking of a class of mechanical systems," in *2016 International Conference on Electrical Engineering, Computing Science and Automatic Control*, pp. 1–5, IEEE, 2016.

[45] E. Jiménez-Rodríguez, J. D. Sánchez-Torres, and A. G. Loukianov, "On optimal predefined-time stabilization," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 17, pp. 3620–3642, 2017.

[46] C. Yong, X. Guangming, and L. Huiyang, "Reaching consensus at a preset time: Single-integrator dynamics case," in *2012 Chinese Control Conference*, pp. 6220–6225, IEEE, 2012.

[47] C. Yong, X. Guangming, and L. Huiyang, "Reaching consensus at a preset time: Double-integrator dynamics case," in *2012 Chinese Control Conference*, pp. 6309–6314, IEEE, 2012.

[48] C. Wang, G. Xie, and M. Cao, "Forming circle formations of anonymous mobile agents with order preservation," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3248–3254, 2013.

[49] J. Wen, C. Wang, W. Luo, and G. Xie, "Finite-time consensus of networked multiagent systems with time-varying linear control protocols," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[50] N. Harl and S. Balakrishnan, "Impact time and angle guidance with sliding mode control," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 6, pp. 1436–1449, 2012.

[51] H. Wang, C. Wang, and G. Xie, "Finite-time containment control of multi-agent systems with static or dynamic leaders," *Neurocomputing*, vol. 226, pp. 1–6, 2017.

[52] Y. Wang, Y. Song, D. J. Hill, and M. Krstic, "Prescribed-time consensus and containment control of networked multiagent systems," *IEEE Transactions on Cybernetics*, 2018.

[53] 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*. (accepted, available online).

[54] R. Aldana-López, D. Gómez-Gutiérrez, E. Jiménez-Rodríguez, J. Sánchez-Torres, and M. Defoort, "On the design of new classes of predefined-time stable systems: A time-scaling approach," *arXiv preprint arXiv:1901.02782*, 2019.

[55] A. Makarenko and H. Durrant-Whyte, "Decentralized data fusion and control in active sensor networks," in *Proceedings of the Seventh International Conference on Information Fusion*, vol. 1, pp. 479–486, 2004.

[56] A. Cunningham, V. Indelman, and F. Dellaert, "DDF-SAM 2.0: Consistent distributed smoothing and mapping," pp. 5220–5227, 2013.

[57] G. A. Hollinger, S. Yerramalli, S. Singh, U. Mitra, and G. S. Sukhatme, "Distributed data fusion for multirobot search," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 55–66, 2015.

[58] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *IFAC world congress*, pp. 1–6, Citeseer, 2005.

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

[60] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 8179–8184, IEEE, 2005.

[61] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *IEEE Conference on Decision and Control*, 2006.

[62] M. A. Demetriou, "Natural consensus filters for second order infinite dimensional systems," *Systems & Control Letters*, vol. 58, no. 12, pp. 826–833, 2009.

[63] H. Bai, R. A. Freeman, and K. M. Lynch, "Robust dynamic average consensus of time-varying inputs," in *Conference on Decision and Control*, pp. 3104–3109, 2010.

[64] C. N. Taylor, R. W. Beard, and J. Humpherys, "Dynamic input consensus using integrators," in *2011 American Control Conference*, pp. 3357–3362, 2011.

[65] F. Chen, Y. Cao, and W. Ren, "Distributed average tracking of multiple time-varying reference signals with bounded derivatives," *IEEE Transactions on Automatic Control*, vol. 57, no. 12, pp. 3169–3174, 2012.

[66] D. Ustebay, R. Castro, and M. Rabbat, "Efficient decentralized approximation via selective gossip," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 805–816, 2011.

[67] B. Mu, G. Chowdhary, and J. P. How, "Efficient distributed sensing using adaptive censoring-based inference," *Automatica*, vol. 50, no. 6, pp. 1590–1602, 2014.

[68] D. W. Casbeer, Y. Cao, E. Garcia, and D. Milutinović, "Average bridge consensus: Dealing with active-passive sensors," in *ASME Dynamic Systems and Control Conference*, 2015.

[69] T. Yucelen, "On networks with active and passive agents," *arXiv preprint arXiv:1405.1480*, 2014.

[70] T. Yucelen and J. D. Peterson, "Active-passive networked multiagent systems," in *2014 Conference on Decision and Control*, pp. 6939–6944, IEEE, 2014.

[71] 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*, pp. 4112–4117, 2015.

[72] J. D. Peterson and T. Yucelen, "An active–passive networked multiagent systems approach to environment surveillance," in *AIAA Guidance, Navigation, and Control Conference*, 2015.

[73] J. D. Peterson, T. Yucelen, and E. Pasiliao, "Generalizations on active-passive dynamic consensus filters," in *American Control Conference*, pp. 3740–3745, 2016.

[74] J. D. Peterson and T. Yucelen, "Application of active-passive dynamic consensus filter approach to multitarget tracking problem for situational awareness in unknown environments," in *AIAA Guidance, Navigation, and Control Conference*, p. 1857, 2016.

[75] T. Yucelen and J. D. Peterson, "Distributed control of active-passive networked multiagent systems," in *IEEE Transactions on Control of Network Systems*, 2016. (to appear, available online).

[76] D. Tran, T. Yucelen, and E. L. Pasiliao, "Formation control with multiplex information networks," *IEEE Transactions on Control Systems Technology*, 2018.

[77] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008.

[78] K.-K. Oh and H.-S. Ahn, "A survey of formation of mobile agents," in *IEEE International Symposium on Intelligent Control*, pp. 1470–1475, 2010.

[79] S. Gomez, A. Diaz-Guilera, J. Gomez-Gardeñes, C. J. Perez-Vicente, Y. Moreno, and A. Arenas, "Diffusion dynamics on multiplex networks," *Physical review letters*, vol. 110, no. 2, p. 028701, 2013.

[80] M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M. A. Porter, S. Gómez, and A. Arenas, "Mathematical formulation of multilayer networks," *Physical Review X*, vol. 3, no. 4, p. 041022, 2013.

[81] V. Nicosia and V. Latora, "Measuring and modelling correlations in multiplex networks," *arXiv preprint arXiv:1403.1546*, 2014.

[82] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *arXiv preprint arXiv:1309.7233*, 2014.

[83] A. Chapman, M. Nabi-Abdolyousefi, and M. Mesbahi, "On the controllability and observability of cartesian product networks," in *IEEE Conferece on Decision and Control*, pp. 80–85, 2012.

[84] A. Chapman, M. Nabi-Abdolyousefi, and M. Mesbahi, "Controllability and observability of network-of-networks via cartesian products," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2668–2679, 2014.

[85] M. Asllani, D. M. Busiello, T. Carletti, D. Fanelli, and G. Planchon, "Turing instabilities on cartesian product networks," *arXiv preprint arXiv:1412.7055*, 2014.

[86] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," *Automatica*, vol. 45, no. 10, pp. 2406–2411, 2009.

[87] S. P. Hou, C.-C. Cheah, and J. Slotine, "Dynamic region following formation control for a swarm of robots," in *IEEE International Conference on Robotics and Automation*, pp. 1929–1934, IEEE, 2009.

[88] L. Brinón-Arranz, A. Seuret, and C. Canudas-de Wit, "Cooperative control design for time-varying formations of multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 8, pp. 2283–2288, 2014.

[89] S. Coogan and M. Arcak, "Scaling the size of a formation using relative position feedback," *Automatica*, vol. 48, no. 10, pp. 2677–2685, 2012.

[90] D. Tran and T. Yucelen, "Control of multiagent formations: A multiplex information networks-based approach," in *ASME Dynamic Systems and Control Conference*, 2015.

[91] D. Tran, T. Yucelen, and E. Pasiliao, "Multiplex information networks for spatially evolving multiagent formations," in *American Control Conference*, pp. 1912–1917, 2016.

[92] C. D. Godsil, G. Royle, and C. Godsil, *Algebraic graph theory*, vol. 207. Springer New York, 2001.

[93] T.-H. Cheng, Z. Kan, J. Rosenfeld, W. E. Dixon, *et al.*, "Decentralized formation control with connectivity maintenance and collision avoidance under limited and intermittent sensing," in *American Control Conference*, pp. 3201–3206, 2014.

[94] Z. Kan, L. Navaravong, J. M. Shea, E. L. Pasiliao, and W. E. Dixon, "Graph matching-based formation reconfiguration of networked agents with connectivity maintenance," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 1, pp. 24–35, 2015.

[95] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[96] C.-L. Liu and Y.-P. Tian, "Formation control of multi-agent systems with heterogeneous communication delays," *International Journal of Systems Science*, vol. 40, no. 6, pp. 627–636, 2009.

[97] Q. Gao, A. S. Kammer, U. Zalluhoglu, and N. Olgac, "Critical effects of the polarity change in delayed states within an LTI dynamics with multiple delays," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 3018–3022, 2015.

[98] Q. Gao and N. Olgac, "Bounds of imaginary spectra of LTI systems in the domain of two of the multiple time delays," *Automatica*, vol. 72, pp. 235–241, 2016.

[99] Q. Gao and N. Olgac, "Stability analysis for LTI systems with multiple time delays using the bounds of its imaginary spectra," *Systems & Control Letters*, vol. 102, pp. 112–118, 2017.

[100] R. Cepeda-Gomez and N. Olgac, "An exact method for the stability analysis of linear consensus protocols with time delay," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1734–1740, 2011.

[101] Y. Cao and W. Ren, "Distributed coordinated tracking with reduced interaction via a variable structure approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 33–48, 2012.

[102] T. Yucelen and J. D. Peterson, "Active-passive networked multiagent systems," in *IEEE Conference on Decision and Control*, pp. 6939–6944, 2014.

[103] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," in *IEEE Conference on Decision and Control*, pp. 416–421, 1993.

[104] R. G. Bartle and D. R. Sherbert, *Introduction to real analysis*. Hoboken, NJ: Wiley, 2011.

[105] J. Li, W. Ren, and S. Xu, "Distributed containment control with multiple dynamic leaders for double-integrator dynamics using only position measurements," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1553–1559, 2012.

[106] M. M. Zavlanos, A. Jadbabaie, and G. J. Pappas, "Flocking while preserving network connectivity," in *IEEE Conference on Decision and Control*, pp. 2919–2924, 2007.

[107] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1416–1428, 2008.

[108] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," November 2012. Version 20121115.

[109] H. Safadi, "Local path planning using virtual potential field," *McGill University School of Computer Science, Tech. Rep*, 2007.

[110] J.-O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, 1992.

[111] T. Balch, "Avoiding the past: a simple but effective strategy for reactive navigation," in *IEEE International Conference on Robotics and Automation*, pp. 678–685, 1993.

[112] D. Tran and T. Yucelen, "On control of multiagent formations through local interactions," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 5177–5182, IEEE, 2016.

[113] D. Tran and T. Yucelen, "Control of multiagent formations: A multiplex information networks-based approach," in *Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, 2015.

[114] D. Tran and T. Yucelen, "Multiplex information networks for spatially evolving multiagent formations," in *American Control Conference (to appear)*, 2016.

[115] H. K. Khalil, *Nonlinear systems*, vol. 3. Prentice Hall, 1996.

[116] G. Wen, Z. Duan, W. Ren, and G. Chen, "Distributed consensus of multi-agent systems with general linear node dynamics and intermittent communications," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 16, pp. 2438–2457, 2014.

[117] D. S. Bernstein, *Matrix mathematics: Theory, facts, and formulas*. Princeton University Press, 2009.

[118] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[119] H. K. Khalil, *Nonlinear systems*, vol. 3. Prentice Hall, 2002.

[120] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein, "Future directions in control in an information-rich world," *IEEE Control Systems Magazine*, vol. 23, no. 2, pp. 20–33, 2003.

[121] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference*, pp. 1859–1864, 2005.

[122] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control systems magazine*, vol. 2, no. 27, pp. 71–82, 2007.

[123] F. Lamnabhi-Lagarrigue, A. Annaswamy, S. Engell, A. Isaksson, P. Khargonekar, R. M. Murray, H. Nijmeijer, T. Samad, D. Tilbury, and P. Van den Hof, "Systems & control for the future of humanity, research agenda: Current and future roles, impact and grand challenges," *Annual Reviews in Control*, vol. 43, pp. 1–64, 2017.

[124] X. Ge, Q.-L. Han, D. Ding, X.-M. Zhang, and B. Ning, "A survey on recent advances in distributed sampled-data cooperative control of multi-agent systems," *Neurocomputing*, vol. 275, pp. 1684–1701, 2018.

[125] L. D. DeVries and M. D. Kutzer, "Kernel design for coordination of autonomous, time-varying multi-agent configurations," in *2016 American Control Conference (ACC)*, pp. 1975–1980, IEEE, 2016.

[126] L. DeVries, A. Sims, and M. D. Kutzer, "Kernel design and distributed, self-triggered control for coordination of autonomous multi-agent configurations," *Robotica*, vol. 36, no. 7, pp. 1077–1097, 2018.

[127] D. Tran, T. Yucelen, and B. Sarsilmaz, "Control of multiagent networks as systems: Finite-time algorithms, time transformation, and separation principle," *IEEE Conference on Decision and Control*, 2018. (accepted).

[128] J. Huang, *Nonlinear output regulation: Theory and applications*, vol. 8. SIAM, 2004.

[129] P. Benner, R. Findeisen, D. Flockerzi, U. Reichl, K. Sundmacher, and P. Benner, *Large-scale networks in engineering and life sciences*. Springer, 2016.

[130] J. P. Hespanha, *Linear systems theory*. Princeton university press, 2009.

[131] D. Tran and T. Yucelen, "A generalized time transformation method for finite-time control," in *American Control Conference*, 2019.

[132] H. Jeffreys and B. Jeffreys, *Methods of mathematical physics*. Cambridge university press, 1999.

[133] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[134] M. Ji, G. Ferrari-Trecate, M. Egerstedt, and A. Buffa, "Containment control in mobile networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1972–1975, 2008.

[135] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[136] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, vol. 15. SIAM, 1994.

[137] K. Mohamed, M. Chadli, and M. Chaabane, "Unknown inputs observer for a class of nonlinear uncertain systems: An LMI approach," *International Journal of Automation and Computing*, vol. 9, no. 3, pp. 331–336, 2012.

[138] M. Corless and J. Tu, "State and input estimation for a class of uncertain systems," *Automatica*, vol. 34, no. 6, pp. 757–764, 1998.

[139] W. Chen and F. N. Chowdhury, "Simultaneous identification of time-varying parameters and estimation of system states using iterative learning observers," *International Journal of Systems Science*, vol. 38, no. 1, pp. 39–45, 2007.

[140] A. Akhenak, M. Chadli, D. Maquin, and J. Ragot, "State estimation of uncertain multiple model with unknown inputs," in *IEEE Conference on Decision and Control*, 2004.

[141] S. Bowong and J. J. Tewa, "Unknown inputs' adaptive observer for a class of chaotic systems with uncertainties," *Mathematical and Computer Modelling*, vol. 48, no. 11, pp. 1826–1839, 2008.

[142] K. Kim, "K-modification and a novel approach to output feedback adaptive control," 2011.

[143] T. Yucelen, "Advances in adaptive control theory: Gradient-and derivative-free approaches," 2011.

[144] K. Kim, T. Yucelen, and A. J. Calise, "A parameter dependent Riccati equation approach to output feedback adaptive control," in *AIAA Guid., Nav., and Contr. Conf., Portland, OR*, 2011.

[145] T. Yucelen, K. Kim, and A. J. Calise, "Derivative-free output feedback adaptive control," in *AIAA Guid., Nav., and Contr. Conf., Portland, OR*, 2011.

[146] K. Kim, T. Yucelen, and A. J. Calise, "Systems and methods for parameter dependent Riccati equation approaches to adaptive control," 2015. US Patent 9,058,028.

[147] T. Yucelen, K. Kim, and A. J. Calise, "Systems and methods for derivative-free adaptive control," 2015. US Patent 8,996,195.

[148] H. K. Khalil, *Nonlinear control*. Prentice Hall, 2015.

[149] P. A. Ioannou and J. Sun, *Robust adaptive control*. Courier Corporation, 2012.

[150] K. Y. Volyanskyy, W. M. Haddad, and A. J. Calise, "A new neuroadaptive control architecture for nonlinear uncertain dynamical systems: Beyond σ- and e-modifications," *IEEE Transactions on Neural Networks*, vol. 20, no. 11, pp. 1707–1723, 2009.

[151] T. Yucelen and W. M. Haddad, "Low-frequency learning and fast adaptation in model reference adaptive control," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 1080–1085, 2013.

[152] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525–1540, 2011.

[153] P. DeLellis, M. di Bernardo, and G. Russo, "On quad, lipschitz, and contracting vector fields for consensus and synchronization of networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 3, pp. 576–583, 2010.

[154] Y. Cao, D. Casbeer, E. Garcia, and X. Zhang, "Towards cost-effective distributed information fusion with partially active sensors in directed networks," in *IEEE Conference on Decision and Control*, pp. 4308–4313, 2016.

[155] S. Mou, E. Garcia, and D. Casbeer, "Distributed algorithms for the average bridge consensus," in *IEEE Conference on Control Technology and Applications*, 2017.

[156] D. M. Tran, T. Yucelen, and S. Jagannathan, "Dynamic information fusion with the integration of local observers, value of information, and active-passive consensus filters," in *AIAA Scitech 2019 Forum*, p. 2262, 2019.

[157] D. Tran, T. Yucelen, and S. Jagannathan, "On local design and execution o a distributed input and state estimation architecture for heterogeneous sensor networks," in *American Control Conference (ACC), 2017*, IEEE, 2017. (accepted).

[158] D. Tran, T. Yucelen, and S. Jagannathan, "A new result on distributed input and state estimation for heterogeneous sensor networks," in *ASME Dynamic Systems and Control Conference*, pp. V002T14A003–V002T14A003, 2017.

[159] D. Tran, T. Yucelen, S. Jagannathan, and D. Casbeer, "Distributed coestimation in heterogeneous sensor networks with time-varying active and passive node roles," in *American Control Conference*, pp. 1033–1038, 2018.

[160] S. Bialas and M. Gora, "On the existence of a common solution to the Lyapunov equations," *Bulletin of the Polish Academy of Sciences Technical Sciences*, vol. 63, no. 1, pp. 163–168, 2015.

[161] D. C. Ramos and P. L. Peres, "An LMI condition for the robust stability of uncertain continuous-time linear systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 675–678, 2002.

[162] K. S. Narendra and J. Balakrishnan, "A common Lyapunov function for stable LTI systems with commuting a-matrices," *IEEE Transactions on Automatic Control*, vol. 39, no. 12, pp. 2469–2471, 1994.

[163] R. N. Shorten and K. S. Narendra, "On common quadratic Lyapunov functions for pairs of stable LTI systems whose system matrices are in companion form," *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 618–621, 2003.

[164] Y. Mori, T. Mori, and Y. Kuroe, "A solution to the common Lyapunov function problem for continuous-time systems," in *IEEE Conference on Decision and Control*, vol. 4, pp. 3530–3531, 1997.

[165] A. G. Ladde and G. S. Ladde, *An Introduction to Differential Equations: Stochastic Modeling, Methods and Analysis (Volume 2)*. World Scientific Publishing Company, 2013.

[166] R. Khasminskii, *Stochastic stability of differential equations*, vol. 66. Springer, 2011.

[167] H. Deng, M. Krstic, and R. J. Williams, "Stabilization of stochastic nonlinear systems driven by noise of unknown covariance," *IEEE Transactions on Automatic Control*, vol. 46, no. 8, pp. 1237–1253, 2001.

[168] L. Arnold, "Stochastic differential equations," *New York*, 1974.

[169] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[170] T. Yucelen and J. D. Peterson, "Distributed control of active-passive networked multiagent systems," *IEEE Transactions on Control of Network Systems*, 2016.

[171] J. D. Peterson, T. Yucelen, S. Jagannathan, and E. Pasiliao, "Event-triggered active-passive dynamic consensus filters," in *American Control Conference*, pp. 3900–3905, IEEE, 2017.

[172] R. Olfati-Saber and P. Jalalkamali, "Coupled distributed estimation and control for mobile sensor networks," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2609–2614, 2012.

[173] T. Yucelen and E. N. Johnson, "Control of multivehicle systems in the presence of uncertain dynamics," *International Journal of Control*, vol. 86, no. 9, pp. 1540–1553, 2013.

[174] D. Viegas, P. Batista, P. Oliveira, C. Silvestre, and C. P. Chen, "Distributed state estimation for linear multi-agent systems with time-varying measurement topology," *Automatica*, vol. 54, pp. 72–79, 2015.

[175] Y. Hu, J. Lam, and J. Liang, "Consensus of multi-agent systems with luenberger observers," *Journal of the Franklin Institute*, vol. 350, no. 9, pp. 2769–2790, 2013.

[176] T. Sadikhov, M. A. Demetriou, W. M. Haddad, and T. Yucelen, "Adaptive estimation using multiagent network identifiers with undirected and directed graph topologies," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 2, p. 021018, 2014.

[177] I. Gutman and W. Xiao, "Generalized inverse of the laplacian matrix and some applications," *Bulletin: Classe des sciences mathematiques et natturalles*, vol. 129, no. 29, pp. 15–23, 2004.

Appendix A: Proof of (3.25)

We first state the following lemma:

Lemma A.0.1 ([117]). *Let $A_1 \in \mathbb{R}^{n\times n}$, $A_2 \in \mathbb{R}^{n\times m}$, $A_3 \in \mathbb{R}^{m\times n}$ and $A_4 \in \mathbb{R}^{m\times m}$. If A_1 and $A_4 - A_3A_1^{-1}A_2$ are nonsingular, then*

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}^{-1} = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix},$$

where

$$M_2 = -A_1^{-1}A_2(A_4 - A_3A_1^{-1}A_2)^{-1}. \tag{A.1}$$

Let

$$q(t) \triangleq [x^{\mathrm{T}}(t), z^{\mathrm{T}}(t), \gamma^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{N(n+p+1)}, \tag{A.2}$$

$$p_c(t) \triangleq [c^{\mathrm{T}}(t), \gamma^*(t)]^{\mathrm{T}} \in \mathbb{R}^{p+1}. \tag{A.3}$$

Since $c(t)$ and $\gamma^*(t)$ are both constants, then $p_c(t) \equiv p_c$. Note that (3.21), (3.23), and (3.24) can be rewritten in a compact form as

$$\dot{q}(t) = A_q q(t) + B_q p_c, \tag{A.4}$$

where

$$A_q \triangleq \begin{bmatrix} \mathrm{I}_N \otimes (A - BK_1) & -\mathrm{I}_N \otimes BK_2 & 0 \\ F \otimes C & 0 & -(F \otimes \mathrm{I}_p)\psi \\ 0 & 0 & -\alpha F \end{bmatrix}, \tag{A.5}$$

$$B_q \triangleq \begin{bmatrix} 0 & 0 \\ -G\otimes \mathrm{I}_p & 0 \\ 0 & \alpha G \end{bmatrix}. \tag{A.6}$$

Note from Assumption 3.3.1 that

$$\begin{bmatrix} \mathrm{I}_N\otimes(A-BK_1) & -\mathrm{I}_N\otimes BK_2 \\ F\otimes C & 0 \end{bmatrix}, \tag{A.7}$$

is Hurwitz (see, for example, [173]). In addition, note from Lemma 3.2.2 that $-\alpha F$ is Hurwitz since $\alpha > 0$. Then, it follows from the upper triangular structure in (A.5) that A_q is Hurwitz.

Since A_q is Hurwitz, then there exists a unique positive-definite matrix P_q such that

$$0 = A_q^{\mathrm{T}}P_q + P_qA_q + R_q, \tag{A.8}$$

holds for a positive-definite matrix R_q. Now, consider the Lyapunov function candidate given by

$$V\big(q+A_q^{-1}B_qp_c\big) = \big(q+A_q^{-1}B_qp_c\big)^{\mathrm{T}}P_q\big(q+A_q^{-1}B_qp_c\big). \tag{A.9}$$

Note that A_q is invertible (since it has a nonzero determinant), $V(0)=0$, $V(q+A_q^{-1}B_qp_c)>0$ for all $q+A_q^{-1}B_qp_c\neq 0$, and $V(q+A_q^{-1}B_qp_c)$ is radially unbounded. The time derivative of (A.9) along the trajectory of (A.4) is given by

$$\begin{aligned} \dot{V}(\cdot) &= (q(t)+A_q^{-1}B_qp_c)^{\mathrm{T}}(A_q^{\mathrm{T}}P_q+P_qA_q)(q(t)+A_q^{-1}B_qp_c) \\ &= -(q(t)+A_q^{-1}B_qp_c(t))^{\mathrm{T}}R_q(q(t)+A_q^{-1}B_qp_c(t)) < 0, \end{aligned} \tag{A.10}$$

and hence,

$$\lim_{t\to\infty} q(t) = -A_q^{-1}B_qp_c. \tag{A.11}$$

Next, since (A.11) implies $A_qq(t)+B_qp_c\to 0$ as $t\to\infty$, we investigate the steady-state behavior when

$$0 = A_qq+B_qp_c. \tag{A.12}$$

From the last row in (A.12), we have

$$-\alpha F\gamma+\alpha G\gamma^* = 0, \tag{A.13}$$

or, equivalently,

$$\gamma = F^{-1}G\gamma^*. \tag{A.14}$$

Since

$$\begin{aligned} F\mathbf{1}_N &= \Big(\mathcal{L}(\mathcal{G})+\text{diag}(G)\Big)\mathbf{1}_N \\ &= \text{diag}(G)\mathbf{1}_N \\ &= G, \end{aligned} \tag{A.15}$$

then it follows from (A.14) that

$$\gamma = \mathbf{1}_N\gamma^*, \tag{A.16}$$

and hence,

$$\lim_{t\to\infty}\xi_i(t) = \xi_i^*\gamma^*. \tag{A.17}$$

From the first two rows of (A.12), we have

$$\begin{bmatrix} \mathrm{I}_N\otimes(A-BK_1) & -\mathrm{I}_N\otimes BK_2 \\ F\otimes C & 0 \end{bmatrix}\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ G\otimes\mathrm{I}_p & F\otimes\mathrm{I}_p \end{bmatrix}\begin{bmatrix} c \\ \xi \end{bmatrix}, \tag{A.18}$$

or, equivalently,

$$A_y\tilde{x} = B_y\tilde{c}, \tag{A.19}$$

where A_y, B_y, $\tilde{x}$, and $\tilde{c}$ in $(A.19)$ correspond to the terms in (A.18), and hence,

$$\tilde{x} = A_y^{-1}B_y\tilde{c}, \tag{A.20}$$

Now, let

$$E_y \triangleq \begin{bmatrix} \mathrm{I}_N \otimes C & 0 \end{bmatrix}, \tag{A.21}$$

then it follows from (A.20) that

$$y = E_y A_y^{-1} B_y \tilde{c}. \tag{A.22}$$

Finally, since Assumption 3.3.2 holds, A_y^{-1} exists. Let

$$A_y^{-1} = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}. \tag{A.23}$$

Note from Lemma A.0.1 that

$$M_2 = F^{-1} \otimes \bar{A}^{-1} \bar{B} (CA^{-1}\bar{B})^{-1}. \tag{A.24}$$

Now, it follows from (A.22) that

$$\begin{aligned}
y &= \begin{bmatrix} \mathrm{I}_N \otimes C & 0 \end{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ G \otimes \mathrm{I}_P & F \otimes \mathrm{I}_p \end{bmatrix} \begin{bmatrix} c \\ \xi \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{I}_N \otimes C & 0 \end{bmatrix} \begin{bmatrix} M_2(G \otimes \mathrm{I}_P) & M_2(F \otimes \mathrm{I}_p) \\ M_4(G \otimes \mathrm{I}_P) & M_4(F \otimes \mathrm{I}_p) \end{bmatrix} \begin{bmatrix} c \\ \xi \end{bmatrix} \\
&= \begin{bmatrix} (\mathrm{I}_N \otimes C) M_2(G \otimes \mathrm{I}_P) & (\mathrm{I}_N \otimes C) M_2(F \otimes \mathrm{I}_p) \end{bmatrix} \begin{bmatrix} c \\ \xi \end{bmatrix} \\
&= (\mathrm{I}_N \otimes C) M_2(G \otimes \mathrm{I}_P) c + (\mathrm{I}_N \otimes C) M_2(F \otimes \mathrm{I}_p) \xi \\
&= \left(F^{-1}G \otimes C\bar{A}^{-1}\bar{B}(CA^{-1}\bar{B})^{-1}\right) c + \left(F^{-1}F \otimes C\bar{A}^{-1}\bar{B}(CA^{-1}\bar{B})^{-1}\right) \xi \\
&= (\mathbf{1}_N \otimes \mathrm{I}_p) c + (\mathrm{I}_N \otimes \mathrm{I}_p) \xi.
\end{aligned} \tag{A.25}$$

Hence, it follows from (A.17) and (A.25) that $\lim_{t\to\infty} y_i(t) = c + \xi_i^* \gamma^*$, $i = 1, \ldots, N$. ■

Appendix B: Proof of (3.28)

In the case when the position of the target and the scaling factor for the density of the resulting formation are time-varying, note that (A.4) becomes

$$\dot{q}(t) \quad = \quad A_q q(t) + B_q p_c(t). \tag{B.1}$$

Following the arguments from the proof of 3.25 in Appendix A, the time derivative of (A.9) along the trajectory of (B.1) can be computed by

$$\begin{aligned} \dot{V}(\cdot) &= -\tilde{q}^{\mathrm{T}}(t) R_q \tilde{q}(t) + 2\tilde{q}^{\mathrm{T}}(t) P_q A_q^{-1} B_q \dot{p}_c(t) \\ &\leq -\lambda_{\min}(R_q) \|\tilde{q}(t)\|_2^2 + 2\|\tilde{q}(t)\|_2 \|P_q A_q^{-1} B_q\|_F \beta \\ &= -\lambda_{\min}(R_q) \|\tilde{q}(t)\|_2 \big(\|\tilde{q}(t)\|_2 - \phi\big), \end{aligned} \tag{B.2}$$

where

$$\tilde{q}(t) \quad \triangleq \quad q(t) + A_q^{-1} B_q p_c(t), \tag{B.3}$$

$$\phi \quad \triangleq \quad 2\|P_q A_q^{-1} B_q\|_F (\beta_1 + \beta_2)/\lambda_{\min}(R_q). \tag{B.4}$$

Therefore, $\dot{V}(\cdot) \leq 0$ outside the compact set

$$\Omega \quad \triangleq \quad \{\tilde{q}(t) : \|\tilde{q}(t)\|_2 \leq \phi\}, \tag{B.5}$$

which proves the ultimate boundedness of the solution $q(t) + A_q^{-1} B_q p_c(t)$ [115], and hence, the result is immediate. ■

Appendix C: Further Discussion on the Stability in Section 8.3

To continue the discussion in Remark 8.3.2, consider, for example, the condition ii), v) and vi) are relaxed, that is, all of the sensors are active, the positive real condition $P_iB = C_i^{\mathrm{T}}J_i^{\mathrm{T}}$ is satisfied with full column rank B, and the inputs are constant, then we can drop some of the leakage terms like "$-\gamma P_i^{-1}\hat{x}_i(t)$" and "$-(\sigma_i K_i + \gamma \mathrm{I}_p)\hat{w}_i(t)$" in our algorithm (8.7) and (8.8) to make it become

$$\dot{\hat{x}}_i(t) = A\hat{x}_i(t) + B\hat{w}_i(t) + L_i\big(y_i(t) - C_i\hat{x}_i(t)\big) - \alpha P_i^{-1}\sum_{i\sim j}(\hat{x}_i(t) - \hat{x}_j(t)),\ \hat{x}_i(0) = \hat{x}_{i0}, \tag{C.1}$$

$$\dot{\hat{w}}_i(t) = J_i(y_i(t) - C_i\hat{x}_i(t)) - \alpha\sum_{i\sim j}(\hat{w}_i(t) - \hat{w}_j(t)),\ \ \hat{w}_i(0) = \hat{w}_{i0}. \tag{C.2}$$

Going through the same procedure in the subsection 8.3.2, we obtain the compact form of the error dynamics as

$$\dot{\tilde{x}}(t) = \bar{A}\tilde{x}(t) - (\mathrm{I}_N \otimes B)\tilde{w}(t) - \alpha P^{-1}(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\tilde{x}(t), \tag{C.3}$$

$$\dot{\tilde{w}}(t) = M\tilde{x}(t) - \alpha(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_p)\tilde{w}(t), \tag{C.4}$$

where $\bar{A}$, M and P are the same matrices as defined in (8.19), (8.20) and (8.23), with $g_i = 1$ for all nodes $i = 1,\ldots,N$. Note that the positive real condition $P_iB = C_i^{\mathrm{T}}J_i^{\mathrm{T}}$ where $P_i > 0$ can be written in the compact form as $P(\mathrm{I}_N \otimes B) = M^{\mathrm{T}}$. In addition, since the leakage terms are dropped, the condition (8.9) is now replaced with the condition $\bar{A}_i^{\mathrm{T}}P_i + P_i\bar{A}_i < 0$ with $P_i > 0$, which can be satisfied easily with appropriate choice of L_i since we assume the matrix A is Hurwitz. In this case, the matrix A is not necessary to be Hurwitz if we assume (A, C_i) is observable, and hence, we can always find L_i such that $\bar{A}$ is Hurwitz.

For the stability analysis of (C.3) and (C.4), we can choose the same Lyapunov function candidate (8.24), then taking the time derivative of $V(\tilde{x},\tilde{w})$ along the trajectories of (C.3) and (C.4) yields

$$\begin{aligned}\dot{V}(\cdot) &= \tilde{x}^{\mathrm{T}}(t)(\bar{A}^{\mathrm{T}}P + P\bar{A})\tilde{x}(t) - 2\tilde{x}^{\mathrm{T}}(t)P(\mathrm{I}_N \otimes B)\tilde{w}(t) - 2\alpha\tilde{x}^{\mathrm{T}}(t)(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\tilde{x}(t) \\ &\quad + 2\tilde{w}^{\mathrm{T}}(t)M\tilde{x}(t) - 2\alpha\tilde{w}^{\mathrm{T}}(t)(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_p)\tilde{w}(t) \\ &= -\tilde{x}^{\mathrm{T}}(t)Q\tilde{x}(t) - 2\alpha\tilde{x}^{\mathrm{T}}(t)(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n)\tilde{x}(t) - 2\alpha\tilde{w}^{\mathrm{T}}(t)(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_p)\tilde{w}(t) \le 0,\end{aligned} \tag{C.5}$$

where $\bar{A}^{\mathrm{T}}P + P\bar{A} \triangleq -Q < 0$ and the last equation is obtained directly from using the condition $\bar{A}_i^{\mathrm{T}}P_i + P_i\bar{A}_i < 0$, and the positive real condition $P(\mathrm{I}_N \otimes B) = M^{\mathrm{T}}$. Note that (C.5) shows that the error dynamics given by (C.3) and (C.4) are Lyapunov stable for all initial conditions. Let $z(t) \triangleq [\tilde{x}^{\mathrm{T}}(t), \tilde{w}^{\mathrm{T}}(t)]^{\mathrm{T}}$ and $\mathcal{S} = \{z(t) \in \mathbb{R}^{N(n+p)} \mid \dot{V}(z(t)) = 0\}$. When $\dot{V}(z(t)) = 0$, we have $\tilde{x}(t) = 0$ since the matrix $Q + (\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n) > 0$. Thus, $\mathcal{S} = \{z(t) \in \mathbb{R}^{N(n+p)} \mid \tilde{x}(t) = 0\}$. Let $\tilde{x}(t)$ be a solution that belongs identically to $\mathcal{S}$, then $\tilde{x}(t) \equiv 0$ means $\dot{\tilde{x}}(t) \equiv 0$, and hence $(\mathrm{I}_N \otimes B)\tilde{w}(t) \equiv 0$ from (C.3). Since B is full column rank, then $(\mathrm{I}_N \otimes B)\tilde{w}(t) \equiv 0$ implies that $\tilde{w}(t) \equiv 0$. Therefore, the only solution that can stay identically in $\mathcal{S}$ is $z(t) \equiv 0$. By Theorem 3.5 in [148], the origin is asymptotically stable. Since the system given by (C.3) and (C.4) is linear time-invariant, the matrix $\begin{bmatrix} \bar{A} - \alpha P^{-1}(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_n) & -(\mathrm{I}_N \otimes B) \\ M & -\alpha(\mathcal{L}(\mathcal{G}) \otimes \mathrm{I}_p) \end{bmatrix}$ is Hurwitz. While the above result is immediate based on the strict assumptions only considered in this appendix to show asymptotic stability, it is still not identical to the other results cited in Section 1 of this paper as well as the results presented in [174–176].

Appendix D: Parameters for Examples in Sections 8.3 and 8.4

In this appendix, we provide the parameters that we use for our illustrative examples in sections 3.3 and 4.3 with 15 decimal places in case the reader want to regenerate our simulation results.

For Example 1 in Section 8.3.3.1, regarding the observer gain L_i, the odd index nodes are subjected to

$$L_i = \begin{bmatrix} 18.969160655470404 & -1.907268388916050 \\ -0.487391998697633 & -0.075498787949882 \\ -1.939393950211166 & 19.129788461945985 \\ -0.284730910528483 & 2.491934726180443 \end{bmatrix}, \tag{D.1}$$

and the even index nodes are subject to

$$L_i = \begin{bmatrix} -2.387919504735957 & 0.357718861908060 \\ 5.830659255697644 & -0.803813918212959 \\ 0.428177912573512 & -2.397655697276556 \\ -1.037509640179160 & 6.765442143562447 \end{bmatrix}. \tag{D.2}$$

In addition,

$$\sigma_1 = \sigma_5 = 0.002061806076927, \tag{D.3}$$

$$\sigma_2 = \sigma_6 = 1.833655794790509 \times 10^{-6}, \tag{D.4}$$

$$\sigma_3 = \sigma_4 = \sigma_7 = \sigma_8 = \sigma_9 = \sigma_{10} = \sigma_{11} = \sigma_{12} = 0.002400000072044, \tag{D.5}$$

and $P_1 = P_5$, $P_2 = P_6$ and $P_3 = P_4 = P_7 = P_8 = P_9 = P_{10} = P_{11} = P_{12}$, where

$$P_1 = 10^3 \times \begin{bmatrix} 1.439835424802165 & -0.034377714548575 & 0.055117661315125 & -0.005443202673014 \\ -0.034377714548575 & 0.004148603989598 & 0.003907233598367 & 0.000148021815711 \\ 0.055117661315125 & 0.003907233598367 & 0.977251053041284 & -0.054353877874992 \\ -0.005443202673014 & 0.000148021815711 & -0.054353877874992 & 0.026092188711795 \end{bmatrix}, \tag{D.6}$$

$$P_2 = 10^2 \times \begin{bmatrix} 0.299361647347857 & 0.000002820408509 & 0.035830529875908 & 0.000000221819347 \\ 0.000002820408509 & 0.696303881680737 & 0.000001019351069 & -0.001153520717465 \\ 0.035830529875908 & 0.000001019351069 & 0.298140534700503 & 0.000013277207542 \\ 0.000000221819347 & -0.001153520717465 & 0.000013277207542 & 3.999240098499779 \end{bmatrix}, \quad \text{(D.7)}$$

$$P_{12} = \begin{bmatrix} 1.440000008960819 & 0.000000001949013 & 0 & 0 \\ 0.000000001949013 & 0.999999999913420 & 0 & 0 \\ 0 & 0 & 1.341667967078101 & 0.529441205638835 \\ 0 & 0 & 0.529441205638835 & 3.496288127071053 \end{bmatrix}. \quad \text{(D.8)}$$

For Example 2 in Section 8.3.3.2,

$$\sigma_1 = \sigma_3 = \sigma_5 = \sigma_7 = 0.002061806076927, \quad \text{(D.9)}$$

$$\sigma_2 = \sigma_4 = \sigma_6 = \sigma_8 = 1.833655794790509 \times 10^{-6}, \quad \text{(D.10)}$$

$$\sigma_9 = \sigma_{10} = \sigma_{11} = \sigma_{12} = 0.002400000072044. \quad \text{(D.11)}$$

For Example 3 in Section 8.3.3.3, the observer gain L_i is chosen such that

$$L_1 = \begin{bmatrix} 19.068611174404705 & -3.813722234880941 \\ -0.485700606350880 & 0.097140121270176 \\ -0.000000000000000 & 0.000000000000000 \\ 0.000000000000000 & -0.000000000000000 \end{bmatrix}, \quad \text{(D.12)}$$

$$L_2 = \begin{bmatrix} -2.387919504735957 & 0.357718861908060 \\ 5.830659255697644 & -0.803813918212959 \\ 0.428177912573512 & -2.397655697276556 \\ -1.037509640179160 & 6.76544214356244 \end{bmatrix}, \quad \text{(D.13)}$$

$$L_3 = \begin{bmatrix} .000000000000002 & -0.000000000000012 \\ 0.000000000000002 & -0.000000000000009 \\ -3.844972067593563 & 19.224860337967815 \\ -0.501192168103495 & 2.505960840517474 \end{bmatrix}, \quad \text{(D.14)}$$

with $L_1 = L_5 = L_9$, $L_2 = L_4 = L_6 = L_8 = L_{10} = L_{12}$ and $L_3 = L_7 = L_{11}$. In addition,

$$\sigma_1 = \sigma_5 \ = 0.001982811972340, \tag{D.15}$$

$$\sigma_2 = \sigma_4 \ = \sigma_6 \ = \sigma_8 \ = 1.833655794790509 \times 10^{-6}, \tag{D.16}$$

$$\sigma_3 = \sigma_7 \ = 0.002400002327086, \tag{D.17}$$

$$\sigma_9 = \sigma_{10} = \sigma_{11} = \sigma_{12} = 0.002400000072044, \tag{D.18}$$

and

$$P_1 = 10^3 \times \begin{bmatrix} 1.478035081597691 & -0.034321006113263 & 0 & 0 \\ -0.034321006113263 & 0.004059666480104 & 0 & 0 \\ 0 & 0 & 0.001491086838912 & 0.000317849850723 \\ 0 & 0 & 0.000317849850723 & 0.005841716956271 \end{bmatrix}, \tag{D.19}$$

$$P_3 = 10^2 \times \begin{bmatrix} 0.014399993247139 & 0.000000000383925 & 0 & 0 \\ 0.000000000383925 & 0.010000000455351 & 0 & 0 \\ 0 & 0 & 6.978268681664298 & -0.230181821168044 \\ 0 & 0 & -0.230181821168044 & 0.188324170653730 \end{bmatrix}, \tag{D.20}$$

with $P_1 = P_5$, $P_2 = P_4 = P_6 = P_8$, $P_3 = P_7$, and $P_9 = P_{10} = P_{11} = P_{12}$, where P_2 and P_{12} are the same as (D.7) and (D.8), respectively.

For Example 4 in Section 8.4.3.1, regarding the observer gain L_i, the odd index nodes are subject to

$$L_i = \begin{bmatrix} 71.783953996258958 & -7.252963875491536 \\ -1.358611885022830 & 0.000475040815981 \\ -7.256525958548080 & 71.801764411541683 \\ -0.011569149768944 & -0.128688835617882 \end{bmatrix}, \tag{D.21}$$

while the even index nodes are subject to

$$L_i = \begin{bmatrix} -21.724842048641705 & 2.256236946732968 \\ 70.098170640403481 & -7.134526961104474 \\ 2.326854788384256 & -21.731989850585336 \\ -7.373583349180001 & 71.293452580781121 \end{bmatrix}. \tag{D.22}$$

In addition,

$$\sigma_1 = \sigma_3 = \sigma_5 = \sigma_7 = \sigma_9 \ = \sigma_{11} = 0.04342963222631, \tag{D.23}$$

$$\sigma_2 = \sigma_4 = \sigma_6 = \sigma_8 = \sigma_{10} = \sigma_{12} = 0.027557828266522, \tag{D.24}$$

and

$$P_1 = \begin{bmatrix} 15.365812598325633 & 0.288438698583006 & 2.891134645171142 & -0.519291422390345 \\ 0.288438698583006 & 8.058139150438166 & 0.489805786441967 & 1.796242594832787 \\ 2.891134645171142 & 0.489805786441967 & 20.900129728630198 & 0.981256246505427 \\ -0.519291422390345 & 1.796242594832787 & 0.981256246505427 & 76.693334412288678 \end{bmatrix}, \tag{D.25}$$

$$P_2 = \begin{bmatrix} 10.026109249319704 & 2.195377067059541 & 0.281414382843850 & -1.215486262312381 \\ 2.195377067059541 & 10.170435495903876 & 0.066429931103742 & 1.090421131679111 \\ 0.281414382843850 & 0.066429931103742 & 9.472902963103000 & 2.834028735894950 \\ -1.215486262312381 & 1.090421131679111 & 2.834028735894950 & 25.585410287198304 \end{bmatrix}, \tag{D.26}$$

with $P_1 = P_3 = P_5 = P_7 = P_9 = P_{11}$ and $P_2 = P_4 = P_6 = P_8 = P_{10} = P_{12}$.

For Example 5 in Section 8.4.3.2, the observer gain L_i is chosen such that

$$L_{1a} = \begin{bmatrix} 3.43 & -2.06 \\ 1.96 & -1.78 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \end{bmatrix}, \tag{D.27}$$

$$L_{1b} = \begin{bmatrix} 0.00 & 0.00 \\ 0.00 & 0.00 \\ 0.00 & 4.90 \\ 0.00 & 2.03 \end{bmatrix}, \tag{D.28}$$

$$L_{2b} = \begin{bmatrix} 0.00 & 0.00 \\ 0.00 & 0.00 \\ 0.31 & -6.14 \\ -0.03 & 1.25 \end{bmatrix}, \tag{D.29}$$

$$L_{3a} = \begin{bmatrix} -16.7 & 0.00 \\ 5.10 & 0.00 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \end{bmatrix}, \tag{D.30}$$

$$L_5 = \begin{bmatrix} 71.78 & -7.25 \\ -1.36 & 0.00 \\ -7.26 & 71.80 \\ -0.01 & -0.13 \end{bmatrix}, \tag{D.31}$$

$$L_6 = \begin{bmatrix} -21.72 & 2.26 \\ 70.10 & -7.13 \\ 2.33 & -21.73 \\ -7.37 & 71.29 \end{bmatrix}, \tag{D.32}$$

with $L_{1a} = L_{2a}$, $L_{1b} = L_{3b}$, $L_{2b} = L_{4b}$, $L_{3a} = L_{4a}$, $L_5 = L_7 = L_9$ and $L_6 = L_8$. In addition,

$$\sigma_{1a} = \sigma_{2a} = 0.176910352176191, \tag{D.33}$$

$$\sigma_{1b} = \sigma_{3b} = 1.899948288575375, \tag{D.34}$$

$$\sigma_{2b} = \sigma_{4b} = 0.088022361660829, \tag{D.35}$$

$$\sigma_{3a} = \sigma_{4a} = 0.622664424352870, \tag{D.36}$$

$$\sigma_5 = \sigma_7 = \sigma_9 = 0.042867890125997, \tag{D.37}$$

$$\sigma_6 = \sigma_8 = 0.027071950457169, \tag{D.38}$$

and

$$P_{1a} = \begin{bmatrix} 15.694244208757699 & 2.033323268063762 & 0.000000000000000 & 0.000000000000000 \\ 2.033323268063762 & 4.361865209185537 & 0.000000000000000 & 0.000000000000000 \\ 0.000000000000000 & 0.000000000000000 & 11.186803471688481 & 8.425791501883902 \\ 0.000000000000000 & 0.000000000000000 & 8.425791501883902 & 28.234219806829103 \end{bmatrix}, \tag{D.39}$$

$$P_{1b} = \begin{bmatrix} 58.803434580028032 & 14.695626639657313 & 0.000000000000000 & 0.000000000000000 \\ 14.695626639657313 & 23.324653578136509 & 0.000000000000000 & 0.000000000000000 \\ 0.000000000000000 & 0.000000000000000 & 18.374966669627405 & 2.314132184510947 \\ 0.000000000000000 & 0.000000000000000 & 2.314132184510947 & 40.176291696525283 \end{bmatrix}, \quad \text{(D.40)}$$

$$P_{2b} = \begin{bmatrix} 5.835227092365172 & 1.443618793797672 & 0.000000000000000 & 0.000000000000000 \\ 1.443618793797672 & 3.236785013594619 & 0.000000000000000 & 0.000000000000000 \\ 0.000000000000000 & 0.000000000000000 & 4.712498556711108 & 6.096255186288076 \\ 0.000000000000000 & 0.000000000000000 & 6.096255186288076 & 41.684901733851333 \end{bmatrix}, \quad \text{(D.41)}$$

$$P_{3a} = \begin{bmatrix} 2.599136817168446 & 2.884026096694552 & 0.000000000000000 & 0.000000000000000 \\ 2.884026096694552 & 8.493188312361060 & 0.000000000000000 & 0.000000000000000 \\ 0.000000000000000 & 0.000000000000000 & 20.899094105514450 & 16.216490598122018 \\ 0.000000000000000 & 0.000000000000000 & 16.216490598122018 & 47.082566058448556 \end{bmatrix}, \quad \text{(D.42)}$$

$$P_5 = \begin{bmatrix} 3.854149037906603 & 0.067878511647866 & 0.758175986616365 & -0.095402213615805 \\ 0.067878511647866 & 2.013031029413099 & 0.108516797222526 & 0.501148644494243 \\ 0.758175986616365 & 0.108516797222526 & 4.625387782296224 & 0.210098567192018 \\ -0.095402213615805 & 0.501148644494243 & 0.210098567192018 & 17.124367695246701 \end{bmatrix}, \quad \text{(D.43)}$$

$$P_6 = \begin{bmatrix} 2.468882758815266 & 0.534438597958104 & 0.047207215780697 & -0.289667612641583 \\ 0.534438597958104 & 2.527205119130702 & 0.006894277635073 & 0.244394471559582 \\ 0.047207215780697 & 0.006894277635073 & 2.258962041615192 & 0.668789059887803 \\ -0.289667612641583 & 0.244394471559582 & 0.668789059887803 & 5.943357864360471 \end{bmatrix}, \quad \text{(D.44)}$$

with $P_{1a} = P_{2a}$, $P_{1b} = P_{3b}$, $P_{2b} = P_{4b}$, $P_{3a} = P_{4a}$, $P_5 = P_7 = P_9$ and $P_6 = P_8$.

Appendix E: Selecting Design Parameter Procedure in Section 8.3

In this appendix, we summarize the effect of the design parameters $\alpha, \gamma, \sigma_i, J_i, K_i, L_i$ and P_i discussed in Remarks 8.3.4 and 8.3.5. The following main points recap the procedure of selecting design parameters:

- The observer gain matrix L_i should be judiciously chosen such that active agents can closely estimate a target of interest when its input is time-invariant. A good performance of active agents will improve the overall performance of the networked system.

- As discussed in Remark 8.3.4, parameters such as γ, σ_i and K_i should be chosen small to limit the effect of leakage terms. However, since σ_i and K_i contribute an important role in the feasibility of the linear matrix inequality condition, one should tune σ_i, K_i and J_i such that the linear matrix inequality condition is satisfied and the norm $\|\sigma_i K_i\|_2$ is small simultaneously.

- A large value can be chosen for α as discussed in Remark 8.3.5. Note that, a large value for α not only helps reduce the ultimate bound but also helps increase the convergence rate.

- Once σ_i, K_i, L_i and J_i are chosen, P_i is obtained from solving the linear matrix inequality given by (8.9).

Note that similar steps can be followed also for choosing the same design parameters used in Section 8.4.

Appendix F: Notation for Chapter 9

In this paper, $\mathbb{R}$ stands for the set of real numbers, $\mathbb{R}^n$ stands for the set of $n \times 1$ real column vectors, $\mathbb{R}^{n\times m}$ stands for the set of $n \times m$ real matrices, $\mathbb{R}_+$ (respectively, $\overline{\mathbb{R}}_+$) stands for the set of positive (respectively, nonnegative-definite) real numbers, $\mathbb{R}_+^{n\times n}$ (respectively, $\overline{\mathbb{R}}_+^{n\times n}$) stands for the set of $n \times n$ positive-definite (respectively, nonnegative-definite) real matrices, $\mathbf{1}_n$ stands for the $n \times 1$ vector of all ones, I_n stands for the $n \times n$ identity matrix, and $\otimes$ stands for the Kronecker product operation. We also use $(\cdot)^{\mathrm{T}}$ for the transpose, $\lambda_{\min}(A)$ (respectively, $\lambda_{\max}(A)$) for the minimum (respectively, maximum) eigenvalue of a square matrix A, $\lambda_i(A)$ for the i-th eigenvalue of a square matrix A, where the eigenvalues of A are ordered from least to greatest value, $\det(\cdot)$ for the determinant, $\mathrm{diag}(a)$ for the diagonal matrix with the vector a on its diagonal, $[x]_i$ for the i-th entry of the vector x, and $[A]_{ij}$ for the i-th row and j-th column entry of the matrix A.

Finally, we recall several graph-theoretical notions (see [5] and [92] for details). An undirected graph $\mathcal{G}$ is defined by a set $\mathcal{V}_\mathcal{G} = \{1,\ldots,n\}$ of nodes and a set $\mathcal{E}_\mathcal{G} \subset \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ of edges. If $(i,j) \in \mathcal{E}_\mathcal{G}$, then the nodes i and j are neighbors and $i \sim j$ indicates the neighboring relation. The number of a node's neighbors are its degree. Specifically, if we let d_i be the degree of node i, then $\mathcal{D}(\mathcal{G}) \triangleq \mathrm{diag}(d) \in \mathbb{R}^{N\times N}$ with $d = [d_1,\ldots,d_N]^{\mathrm{T}}$ is the degree matrix of a graph $\mathcal{G}$. A path $i_0 i_1 \ldots i_L$ is a (finite) sequence of nodes such that $i_{k-1} \sim i_k$, $k = 1,\ldots,L$, and a graph $\mathcal{G}$ is said to be connected if a path exists between any distinct node pairs. $\mathcal{A}(\mathcal{G}) \in \mathbb{R}^{N\times N}$ is the adjacency matrix of a graph $\mathcal{G}$ defined by $[\mathcal{A}(\mathcal{G})]_{ij} = 1$ when $(i,j) \in \mathcal{E}_\mathcal{G}$ and $[\mathcal{A}(\mathcal{G})]_{ij} = 0$ otherwise. The Laplacian matrix of a graph, $\mathcal{L}(\mathcal{G}) \in \overline{\mathbb{R}}_+^{N\times N}$ is now defined by $\mathcal{L}(\mathcal{G}) \triangleq \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$. For an undirected and connected graph $\mathcal{G}$, note that the spectrum of the Laplacian can be ordered as $0 = \lambda_1(\mathcal{L}(\mathcal{G})) < \lambda_2(\mathcal{L}(\mathcal{G})) \leq \cdots \leq \lambda_N(\mathcal{L}(\mathcal{G}))$ with the eigenvector $\mathbf{1}_N$ corresponds to the zero eigenvalue $\lambda_1(\mathcal{L}(\mathcal{G}))$ and $\mathcal{L}(\mathcal{G})\mathbf{1}_N = 0_N$ and both $\mathrm{e}^{\mathcal{L}(\mathcal{G})}\mathbf{1}_N = \mathbf{1}_N$ hold.

Appendix G: Mathematical Preliminaries for Chapter 10

The notation used in this paper is fairly standard. Specifically, $\mathbb{Z}_+$ denotes the set of positive integer numbers, $\mathbb{R}_+$ denotes the set of positive real numbers, $\mathbb{R}^n$ denotes the set of $n \times 1$ real column vectors, $\mathbb{R}^{n\times m}$ denotes the set of $n \times m$ real matrices, $\mathbb{R}_+^{n\times n}$ (resp., $\overline{\mathbb{R}}_+^{n\times n}$) denotes the set of $n \times n$ positive-definite (resp., nonnegative definite) real matrices, $\mathbf{1}_n$ denotes the $n \times 1$ vector of all ones, and I_n denotes the $n \times n$ identity matrix. In addition, we write $(\cdot)^{\mathrm{T}}$ for transpose, $(\cdot)^+$ for generalized inverse, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ for the minimum and maximum eigenvalue of the symmetric matrix A, respectively, $\lambda_i(A)$ for the i-th eigenvalue of A, where A is symmetric and the eigenvalues are ordered from least to greatest value, block–diag$_{i=1,2}(A_1,\ldots,A_n)$ for the block diagonal matrix with $A_1,\ldots,A_n$ are square matrices lying along the diagonal and all other entries of the matrix equal 0, diag(a) for the diagonal matrix with the vector a on its diagonal, $[x]_i$ for the entry of the vector x on the i-th row, and A_{ij} for the entry of the matrix A on the i-th row and j-th column. In addition, for $A \in \mathbb{R}^{n\times m}$, $\mathcal{R}(A)$ denotes the range of A, rank(A) denotes the rank of A, $\mathcal{N}(A)$ denotes the null space of A, def$(A) \triangleq \dim\mathcal{N}(A)$ denotes the defect of A.

Next, we recall some basic notions from graph theory and refer to textbooks Refs. [5] and [92] for details. Specifically, an undirected graph $\mathcal{G}$ is defined by a set $\mathcal{V}_\mathcal{G} = \{1,\ldots,N\}$ of nodes and a set $\mathcal{E}_\mathcal{G} \subset \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ of edges. If $(i,j) \in \mathcal{E}_\mathcal{G}$, then the nodes i and j are neighbors and the neighboring relation is indicated with $i \sim j$. The degree of a node is given by the number of its neighbors. Letting d_i be the degree of node i, then the degree matrix of a graph $\mathcal{G}$, $\mathcal{D}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is given by $\mathcal{D}(\mathcal{G}) \triangleq \mathrm{diag}(d)$, $d = [d_1,\ldots,d_N]^{\mathrm{T}}$. A path $i_0 i_1 \ldots i_L$ is a finite sequence of nodes such that $i_{k-1} \sim i_k$, $k = 1,\ldots,L$, and a graph $\mathcal{G}$ is connected if there is a path between any pair of distinct nodes. The adjacency matrix of a graph $\mathcal{G}$, $\mathcal{A}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is given by $[\mathcal{A}(\mathcal{G})]_{ij} = 1$ if $(i,j) \in \mathcal{E}_\mathcal{G}$ and $[\mathcal{A}(\mathcal{G})]_{ij} = 0$ otherwise. The Laplacian matrix of a graph, $\mathcal{L}(\mathcal{G}) \in \overline{\mathbb{R}}_+^{N\times N}$, playing a central role in many graph-theoretic treatments of sensor networks, is given by $\mathcal{L}(\mathcal{G}) \triangleq \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$. The spectrum of the Laplacian of an undirected and connected graph can be ordered as $0 = \lambda_1(\mathcal{L}(\mathcal{G})) < \lambda_2(\mathcal{L}(\mathcal{G})) \leq \cdots \leq \lambda_N(\mathcal{L}(\mathcal{G}))$ with $\mathbf{1}_N$ as the eigenvector corresponding to the

zero eigenvalue $\lambda_1(\mathcal{L}(\mathcal{G}))$ and $\mathcal{L}(\mathcal{G})\mathbf{1}_N = 0_N$ and $e^{\mathcal{L}(\mathcal{G})}\mathbf{1}_N = \mathbf{1}_N$. Here, we assume that the graph $\mathcal{G}$ of a given sensor network is undirected and connected.

The following lemmas are necessary for the main results of this paper.

Lemma G.0.1 (**Lemma 3, [177].**). *The Laplacian of a connected, undirected graph satisfies* $\mathcal{L}(\mathcal{G})\mathcal{L}^{+}(\mathcal{G}) = \mathrm{I}_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^{\mathrm{T}}$

Lemma G.0.2 (**Fact 2.10.12, [117].**). *Let* $A \in \mathbb{R}^{n\times m}$ *and* $B \in \mathbb{R}^{m\times l}$. *Then,* $rank(AB) = rank(A)$ *if and only if* $\mathcal{R}(AB) = \mathcal{R}(A)$.

Lemma G.0.3 (**Fact 6.4.43, [117].**). *Let* $A \in \mathbb{R}^{n\times m}$ *and* $B \in \mathbb{R}^{n\times l}$. *Then,* $\mathcal{R}(A) \subseteq \mathcal{R}(B)$ *if and only if* $BB^{+}A = A$.

Lemma G.0.4 (**Theorem 2.4.3, [117].**). *Let* $A \in \mathbb{R}^{n\times m}$, *then* $\mathcal{N}(A) = \mathcal{N}(A^{\mathrm{T}}A)$.

Appendix H: Proof of Lemma 10.3.1

Let $H \triangleq (\mathbf{1}_N^{\mathrm{T}} \otimes \mathrm{I}_n)$, then (10.12) becomes

$$\varepsilon \;=\; S^{+} H\mathcal{M}\zeta = (H\mathcal{M}H^{\mathrm{T}})^{+} H\mathcal{M}\zeta. \tag{H.1}$$

It should be noted that

$$(\mathbf{1}_N^{\mathrm{T}} \otimes \mathrm{I}_n)\mathcal{M}(\mathbf{1}_N \otimes \varepsilon) \;=\; H\mathcal{M}(\mathbf{1}_N \otimes \varepsilon) = S\varepsilon. \tag{H.2}$$

Utilize (H.1) and (H.2), the left hand side of (10.16) can be rewritten as

$$\begin{aligned}
(\mathbf{1}_N^{\mathrm{T}} \otimes \mathrm{I}_n)\mathcal{M}\omega &= (\mathbf{1}_N^{\mathrm{T}} \otimes \mathrm{I}_n)\mathcal{M}\big((\mathbf{1}_N \otimes \varepsilon) - \zeta\big) \\
&= H\mathcal{M}(\mathbf{1}_N \otimes \varepsilon) - H\mathcal{M}\zeta \\
&= S\varepsilon - H\mathcal{M}\zeta \\
&= (H\mathcal{M}H^{\mathrm{T}})(H\mathcal{M}H^{\mathrm{T}})^{+} H\mathcal{M}\zeta - H\mathcal{M}\zeta \\
&= \big((H\mathcal{M}H^{\mathrm{T}})(H\mathcal{M}H^{\mathrm{T}})^{+} H\mathcal{M} - H\mathcal{M}\big)\zeta \\
&= R\zeta,
\end{aligned} \tag{H.3}$$

where $R \triangleq \big((H\mathcal{M}H^{\mathrm{T}})(H\mathcal{M}H^{\mathrm{T}})^{+} H\mathcal{M} - H\mathcal{M}\big)$.

The matrix $H\mathcal{M}$ can be rewritten as

$$\begin{aligned}
H\mathcal{M} &= \begin{bmatrix} k_1 M_1 & k_2 M_2 & \ldots & k_N M_N \end{bmatrix} \\
&= \begin{bmatrix} k_1\,\mathrm{diag}(m_1) & k_2\,\mathrm{diag}(m_2) & \ldots & k_N\,\mathrm{diag}(m_N) \end{bmatrix} \in \mathbb{R}^{n \times Nn}.
\end{aligned} \tag{H.4}$$

We now define $\bar{m} \triangleq k_1 m_1 + k_2 m_2 + \ldots + k_N m_N$, and note that $N > 1$. Clearly, $\mathrm{rank}(H\mathcal{M}) \leq n$. In addition, since elements of m_i for $i = 1, \ldots, N$ are nonnegative and the column vectors of $H\mathcal{M}$ are multiples of

$e_1, e_2, \ldots, e_n$ where e_j is the unit vector with the j-th element is 1 and 0 otherwise, $\bar{m}$ only obtains an 0 element when $H\mathcal{M}$ has a zero row. Therefore,

$$\begin{aligned} \text{rank}(H\mathcal{M}) &= \text{number of positive elements in } \bar{m} \\ &= n - (\text{number of 0 elements in } \bar{m}). \end{aligned} \tag{H.5}$$

Similarly, S can be rewritten as

$$\begin{aligned} S &= \text{diag}(k_1 m_1 + k_2 m_2 + \ldots + k_N m_N) \\ &= \text{diag}(\bar{m}). \end{aligned} \tag{H.6}$$

Hence, it follows directly that

$$\begin{aligned} \text{rank}(S) &= \text{number of positive elements in } \bar{m} \\ &= n - (\text{number of 0 elements in } \bar{m}). \end{aligned} \tag{H.7}$$

From (H.5) and (H.7), we have

$$\text{rank}(H\mathcal{M}) = \text{rank}(S) = \text{rank}(H\mathcal{M}H^{\mathrm{T}}). \tag{H.8}$$

Utilize (H.8) and the result of Lemma G.0.2 with $A \triangleq H\mathcal{M}$ and $B \triangleq H^{\mathrm{T}}$, we obtain

$$\mathcal{R}(H\mathcal{M}) = \mathcal{R}(S) = \mathcal{R}(H\mathcal{M}H^{\mathrm{T}}). \tag{H.9}$$

Therefore, it now follows directly from Lemma G.0.3 that

$$(H\mathcal{M}H^{\mathrm{T}})(H\mathcal{M}H^{\mathrm{T}})^{+}H\mathcal{M} = H\mathcal{M}, \tag{H.10}$$

or $R = 0$. As a result, (10.16) is now immediate. ■

Appendix I: Copyright Permissions

The permission below is for the use of material in Chapter 2.

8/20/2019 Rightslink® by Copyright Clearance Center

Copyright Clearance Center RightsLink® Home Create Account Help LIVE CHAT

IEEE Requesting permission to reuse content from an IEEE publication

Title:	Formation Control With Multiplex Information Networks
Author:	Dzung Tran
Publication:	Control Systems Technology, IEEE Transactions on
Publisher:	IEEE
Date:	Dec 31, 1969

Copyright © 1969, IEEE

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials.

Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

The permission below is for the use of material in Chapter 3.

Copyright Clearance Center RightsLink®

Home | Create Account | Help | LIVE CHAT

IEEE Requesting permission to reuse content from an IEEE publication

Title:	On control of multiagent formations through local interactions
Conference Proceedings:	2016 IEEE 55th Conference on Decision and Control (CDC)
Author:	Dzung Tran
Publisher:	IEEE
Date:	Dec. 2016

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK | CLOSE WINDOW

The permission below is for the use of material in Chapter 8.

USF

Dung Tran <dtran3@mail.usf.edu>

Permission to reuse paper in dissertation

2 messages

Dzung Tran <dtran3@mail.usf.edu> Tue, Aug 20, 2019 at 10:58 AM
To: roboticsandai.editorial.office@frontiersin.org

Hello,

My name is Dzung Tran, and I am the first author of the paper: "Distributed Input and State Estimation Using Local Information in Heterogeneous Sensor Networks" published in July 2017 (the online link to the paper is: https://www.frontiersin.org/articles/10.3389/frobt.2017.00030/full).

I would like to have permission to put this paper on my dissertation. Please let me know what I need to do to obtain permission in this case. Thank you very much.

Best regards,
Dzung Tran

Frontiers in Robotics and AI Editorial Office <roboticsandai.editorial.office@frontiersin.org> Wed, Aug 21, 2019 at 2:34 AM
To: Dzung Tran <dtran3@mail.usf.edu>

Dear Dr Tran,

Thank you for your email.

Under the Frontiers Terms and Conditions, authors retain the copyright to their work. Furthermore, all Frontiers articles are Open Access and distributed under the terms of the Creative Commons Attribution License (CC-BY 3,0), which permits the use, distribution and reproduction of material from published articles, provided the original authors and source are credited, and subject to any copyright notices concerning any third-party content.

You can therefore freely reuse a figure of an article published at Frontiers without having to ask for permission. More information about CC-BY can be found here: http://creativecommons.org/licenses/by/4.0/

However, you should contact the corresponding author of the original article to inform them about your intention to reproduce their figure(s) and to ask whether any third-party copyright might apply, to whom in such a case you would have to ask permission to reproduce the work.

Please also make sure to properly acknowledge, in your manuscript, the original authors and source of the figures that you intend to reproduce. We recommend to use the format below:
"Figure as originally published in Metpally RPR, Nasser S, Malenica I, Courtright A, Carlson E, Ghaffari L, Villa S, Tembe W and Van Keuren-Jensen K (2013) . Front. Genet. 4:20. doi: 10.3389/fgene.2013.00020".

Please let me know if you have any other questions or concerns.

Thank you for your time and attention,

With best regards,

Chiara

--
Chiara Sulpizio, PhD
Review Operations Specialist

Frontiers | Editorial Office - Collaborative Peer Review Team
Review Operations Manager: Judyta Sorokowska

The permission below is for the use of material in Chapter 10.

USF

Dung Tran <dtran3@mail.usf.edu>

Permission to republish papers in dissertation

2 messages

Dzung Tran <dtran3@mail.usf.edu> Tue, Aug 20, 2019 at 11:17 AM
To: heatherb@aiaa.org

Dear Heather Brennan,

My name is Dzung Tran, and I am the first authors of the following two conference papers (AIAA Scitech Forum):

1. "Dynamic Information Fusion with the Integration of Local Observers, Value of Information, and Active-Passive Consensus Filter" (2019)
2. "Resilient Control of Active-Passive Networked Multiagent Systems in the Presence of Persistent Disturbances" (2017).

I would like to have permission to put these two papers in my dissertation. Please let me know if the permission is granted. Thank you very much.

Best regards,
Dzung Tran

Heather Brennan <HeatherB@aiaa.org> Tue, Aug 20, 2019 at 11:44 AM
To: Dzung Tran <dtran3@mail.usf.edu>

Thank you for your inquiry. AIAA grants permission for you to reprint the two conference papers described below in your dissertation.

Please acknowledge within the main text or a footnote that these sections/chapters of your dissertation are reprinted with permission (e.g., “From [paper title and authors]; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.”). Note that the original source should be cited in full in the reference list.

If you have any further questions, please let me know.

Sincerely,

Heather A. Brennan

Director, Publications

-

American Institute of Aeronautics and Astronautics www.aiaa.org
12700 Sunrise Valley Drive, Suite 200

Reston, VA 20191-5807