# Fast Convergence in Consensus Control of Leader-Follower Multi-Agent Systems

David Buzorgnia

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Electrical and Computer Engineering) at

Concordia University

Montreal, Quebec, Canada

September 2019

© David Buzorgnia, 2019

**CONCORDIA UNIVERSITY**

**SCHOOL OF GRADUATE STUDIES**

This is to certify that the thesis prepared

By: David Buzorgnia

Entitled: Fast Convergence in Consensus Control of Leader-Follower Multi-Agent Systems

and submitted in partial fulfilment of the requirements for the degree of

**Master of Applied Science (Electrical and Computer Engineering)**

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Dr. Khashayar Khorasani, Chair

_____ Dr. Youmin Zhang, External Examiner

_____ Dr. Khashayar Khorasani, Examiner

_____ Dr. Amir G. Aghdam, Supervisor

Approved by: _____

Yousef R. Shayan, Chair
Department of Electrical and Computer Engineering

_____ 2019 _____

Amir Asif, Dean
Faculty of Engineering and Computer Science

# Abstract

**Fast Convergence in Consensus Control of Leader-Follower Multi-Agent Systems**

David Buzorgnia

In this thesis, different distributed consensus control strategies are introduced for a multi-agent network with a leader-follower structure. The proposed strategies are based on the nearest neighbor rule, and are shown to reach consensus faster than conventional methods. Matrix equations are given to obtain equilibrium state of the network based on which the average-based control input is defined accordingly. Two network control rules are subsequently developed, where in one of them the control input is only applied to the leader, and in the other one it is applied to the leader and its neighbors. The results are then extended to the case of a time-varying network with switching topology and a relatively large number of agents. The convergence performance under the proposed strategies in the case of a time-invariant network with fixed topology is evaluated based on the location of the dominant eigenvalue of the closed-loop system. For the case of a time-varying network with switching topology, on the other hand, the state transition matrix of the system is investigated to analyze the stability of the proposed strategies. Finally, the input saturation in agents' dynamics is considered and the stability of the network under the proposed methods in the presence of saturation is studied.

## Acknowledgements

First, I would like to express my sincere gratitude to my supervisor and mentor, Dr. Amir G. Aghdam, for his support and guidance throughout this research. I vastly appreciate his patient, encouragement, and advice during my studies.

My sincere thanks go to Dr. Mahya Rezaei for her support through finalizing my research.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them.

# List of Publications

[1] **D. Buzorgnia** and A. G. Aghdam, "A follower-based control allocation in multi-agent networks," in *Proceedings of the 2018 American Control Conference*, 2018, pp. 43-48.

[2] **D. Buzorgnia** and A. G. Aghdam, "Distributed consensus protocols for time-varying multi-agent networks with improved convergence properties," submitted to the *2020 American Control Conference.*

## Contribution of Authors

This thesis is written in a manuscript-based format. The presented papers are written by David Buzorgnia and co-authored by Dr. Amir G. Aghdam, both from the Department of Electrical and Computer Engineering, Concordia University. As the research supervisor, Dr. Aghdam provided guidance and direction on all aspects of the the work and gave constructive comments. All the research presented in this dissertation is performed by the author.

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

The study of population density distribution is an interesting subject in biology concerning both animals and plants, and has been investigated by researchers for more than half a century [1]. Multi-agent systems have attracted researchers from different disciplines in recent years due to their applications in a variety of engineering and science problems [2–6]. In particular, there has been an increasing interest in control community in the control of a network of multi-agent systems, where it is desired to achieve a global objective such as rendezvous, flocking, formation and consensus by using local information of the agents [7–12]. In control of multi-agent systems, it is desired to achieve a global objective in forming a group by properly guiding each individual agent [13,14]. Different

computer-based models and algorithms were developed in the past three decades based on some dynamical equations in order to simulate the individual and team behavior in this type of system [15, 16]. Several distributed control schemes are introduced in the literature to achieve the above objectives for networks with small or large number of agents, linear or nonlinear agent dynamics and fixed or switching topology [17–22]. In particular, the consensus control problem is of special interest in the coordination of multi-vehicle systems and data fusion in wireless sensor networks. Speed of convergence is one of the important objectives in the control of multi-agent networks.

## 1.2 Literature Review and Preliminaries

Different aspects of multi-agent systems have been thoroughly investigated in the past two decades and several global objectives in this type of network such as consensus, flocking, rendezvous and formation have been studied [11, 23–26]. Graph theory has been used extensively in the literature to analyze various aspects of multi-agent networks, where each agent is considered as a node and interaction between any pair of agents is represented by an edge [24]. The graph of a network can be either directed or undirected. In the case of an undirected graph, the communication between any pair of connected nodes is bidirectional [17, 27], while in a directed graph (digraph) it can be unidirectional [25, 28, 29]. A directed graph can be strongly or weakly connected [24]. In a strongly connected graph, there is a path between any pair of nodes. In [30], an

algorithm is proposed to reach average consensus for a strongly connected balanced or symmetric network, and in [31], a discrete-time average consensus control scheme is provided for a strongly connected network. In a weakly connected graph, on the other hand, there is no path from some nodes to some other nodes, which imposes some limitations on the applicability of the consensus control algorithms. For a network with a time-varying weighted digraphs, the authors in [32] investigate average consensus for a weakly connected balanced network. In [33], the consensus control problem is studied for a network with weakly connected subgraphs.

Different structures are used to model the information flow between the agents, e.g., behavior-based, virtual structure and leader follower, and efficient control algorithms are proposed for each [23]. In particular, the consensus problem is of special importance in this type of network, where all agents are desired to asymptotically reach a common state value by limited exchange of information between neighboring agents [34–36]. This problem has application in emerging technologies such as the coordination of autonomous vehicles and data fusion in wireless sensor networks [37–39]. Various optimization approaches are proposed in the literature to improve the overall performance of the network [40–43]. It is well-known that the network structure plays a key role in the choice of an efficient control algorithm for the agents, and in particular, the leader-follower structure has attracted much attention in the literature [44–48].

Several distributed algorithms are proposed in the literature for consensus control

of leader-follower multi-agent networks [11, 25, 49]. In [11], the infinity norm of the state transition matrix is investigated for the stability analysis of a network with switching topology. The authors in [20] provide a consensus control method for a leader-follower network under both fixed and switching topologies. Consensus control of a network with time-varying edge weights in the presence of communication noise is studied in [50]. In [51] and [52] , the consensus problem for a network with switching topology and time-varying delay is studied. One of the simple yet effective classes of distributed consensus control methods is based on the nearest-neighbor rule [11, 17]. Nearest-neighbor-based methods are effectively used in both homogeneous networks, where all agents have the same dynamics, and heterogeneous networks, where the dynamic models of different agents are not the same [23, 37, 53–55]. It is also used in networks with both linear and nonlinear agent dynamics [56–59].

While the methods cited in the previous paragraph are widely used in consensus control of multi-agent networks, it is usually assumed that the topology of the network is fixed and its parameters are time-invariant. Many practical multi-agent systems, however, are subject to change during the mission. Furthermore, it is often assumed that the control system does not reach a physical limit, i.e., no saturation occurs in the system. However, it is known that in real-world control systems saturation is ubiquitous and any actuator or sensor is subject to saturation [60, 61]. In [60], the input saturation is studied for a symmetric network with fixed topology. For a homogeneous asymmetric

network with fixed topology, [62] proposes a method based on "bang-bang" type of consensus protocol in the presence of saturation. In [61], input saturation is studied for a linear multi-agent system with switching topology using an observer-based method. Input saturation for a heterogeneous network with first-order and second-order dynamics is investigated in [63].

## 1.3 Thesis Contributions

In this thesis, different algorithms are introduced for consensus control of leader-follower multi-agent systems. The proposed algorithms are based on adding the control input to the nearest-neighbor rule, and use the local information of a certain subset of agents to generate the control command. The main objective is to achieve consensus with a high convergence rate. The algorithms are then properly modified for the case of switching topology and time-varying network parameters as well as input saturation.

Consensus control is studied in the discrete-time domain. Using the properties of the transition matrix of the Markov chain [24], a method is proposed for a leaderless network with a fixed topology and undirected graph to find the equilibrium state of the network. The eigenvalues of the closed-loop system for certain topologies are also derived for stability analysis.

A distributed control strategy is also proposed based on the relative state of the leader with respect to its neighbors, which is applicable to any symmetric network with

weighted links. A convergence analysis is provided for a multi-agent system with fixed topology based on the location of the dominant eigenvalue of the system matrix of the network. A control command is also derived for the agents based on the states of the neighbors of the leader. The convergence rate is evaluated by investigating the dominant eigenvalue of the system. Furthermore, the stability of the proposed control scheme for a network with switching topology and time-varying weights is investigated based on the state transition matrix of the system. By letting every agent acts as a leader, an algorithm is developed based on the hierarchical structure of the network, and its stability is investigated accordingly using the state transition matrix. Finally, input saturation is considered in the agents' dynamics and a convergent control algorithm is proposed in this case. Several numerical examples are provided to demonstrate the effectiveness of the proposed strategies.

## 1.4 Thesis Layout

The structure of the thesis is as follows.

- **Chapter 1** includes the motivation behind this study, the literature review on the consensus control for various graph structures, and the conclusion of the current work.

- **Chapter 2** introduces an average-based control law which can be applied to the

leader in order to achieve the consensus objective. The proposed method is further developed to generate control inputs for the leader's immediate followers. It is shown that under a mild condition on the topology of the network, the proposed follower-based control allocation strategy converges faster than a leader-based control rule. Simulations confirm the effectiveness of the proposed method.

- **Chapter 3** introduces different algorithms for consensus control of leader-follower multi-agent systems. The proposed algorithms are based on the nearest-neighbor rule, and use the local information of a certain subset of agents to generate the control command. The main objective is to achieve consensus with a high convergence rate. The results are then extended to the case of a time-varying network with switching topology and a relatively large number of agents. The convergence performance under the proposed strategies in the case of a time-invariant network with fixed topology is evaluated based on the location of the dominant eigenvalue of the closed-loop system. For the case of a time-varying network with switching topology, on the other hand, the state transition matrix of the system is investigated to analyze the stability of the proposed strategies. Finally, the input saturation in agents' dynamics is considered and the stability of the network under the proposed methods in the presence of saturation is studied. The effectiveness of the theoretical findings is verified by several numerical examples.

- **Chapter 4** presents the conclusion and the possible direction for the future work.

# Chapter 2

# A Follower-Based Control Allocation in Multi-Agent Networks

This chapter investigates the consensus problem in a multi-agent system with a leader, using the concept of swarm intelligence. Matrix equations are given to obtain equilibrium state of the network, and the average-based control input is defined accordingly. Two network control rules are subsequently developed, where in one of them the control input is only applied to the leader, and in the other one it is applied to the leader and its neighbors (follower-based control allocation strategy or swarm intelligence approach). It is shown that the latter control strategy has a faster convergence rate. Simulations confirm the efficacy of the proposed follower-based control allocation strategy.

This chapter is based on the following publication:

D. Buzorgnia and A. G. Aghdam, "A Follower-Based Control Allocation in Multi-Agent Networksm" in *2018 American Control Conference*, 2018, pp. 43-48.

## 2.1 Introduction

The study of population density distribution is an interesting subject in biology concerning both animals and plants, and has been investigated by researchers for more than half a century [1]. Some fundamental research in the control of multi-agent systems was inspired by sophisticated biological systems such as flock of birds, swarm of insects, and school of fish. In control of multi-agent systems, it is desired to achieve a global objective in forming a group by properly guiding each individual agent [13, 14]. Different computer-based models and algorithms were developed in the past three decades based on some dynamical equations in order to simulate the individual and team behavior in this type of system [15, 16]. It is well-known that graph-theoretic techniques can be very effective in the formulation and analysis of multi-agent network control systems [11, 23, 24].

Different aspects of multi-agent systems have been thoroughly investigated in the past two decades and several global objectives in this type of network such as consensus, flocking, rendezvous and formation have been studied [11, 23–26]. Cooperation between agents is of utmost importance in achieving these global objectives in the network. Various optimization approaches are proposed in the literature to improve the overall

performance of the network [40–43]. It is well-known that network structure plays a key role in establishing efficient cooperation between the agents, and in particular, leader-follower structures have attracted much attention in the literature [44–48]. Several algorithms are developed for the control of multi-agent systems with different structures.

While several high-performance strategies are developed for control of multi-agent networks, their convergence rate may not be desirable for many applications. An average-based control law is introduced, which can be applied to the leader in order to achieve the consensus objective. Motivated by the swarm intelligence paradigm, the proposed method is further developed to generate control inputs for the leader's immediate followers. It is shown that under a mild condition on the topology of the network, the proposed follower-based control allocation strategy converges faster than a leader-based control rule. Simulations confirm the effectiveness of the proposed method.

The outline of the chapter is as follows. In Section 2.2, some preliminaries on consensus problem in multi-agent networks are presented and the problem is formulated. Then in Section 2.3 the swarm intelligence-based control allocation technique is introduced as the main result of the chapter and numerical examples are given to verify the theoretical findings. Conclusions are drawn in Section 2.4.

## 2.2 Consensus Problem

Consider a multi-agent network in a 2D space, represented by directed graph (digraph) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, ..., n\}$ is the set of vertices representing the agents, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges representing the communication links between the agents. Consensus is one of the important global objectives in this type of network, which is often desired to be achieved using a distributed control strategy [26, 40, 41].

### 2.2.1 Leaderless Network

Assume all agents are similar in terms of their functionality with respect to the other agents. A discrete-time model of a simple consensus algorithm for the network, where the agents move with the same speed but different heading angles is given below [11]:

$$x_i(k+1) = [x_i(k)]_r, \tag{2.1}$$

$$[x_i(k)]_r = \frac{1}{1+d_i}\left(x_i(k) + \sum_{j \in \mathcal{N}_i} x_j(k)\right), \tag{2.2}$$

where $k \in \{0, 1, 2, ...\}$ is a discrete-time index, $x_i \in \mathbb{R}$ is the state of the $i^{th}$ agent (heading angle), $\mathcal{N}_i$ is the set of neighbors of agent $i$ (i.e., the set of agents that can exchange information with agent $i$), and $d_i$ is the cardinality of this set. Using a graph-theoretic

approach, (2.1) can be written in matrix form as:

$$\boldsymbol{x}(k+1) = (I+D)^{-1}(I+A)\boldsymbol{x}(k), \tag{2.3}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the network state vector, $A$ is the adjacency matrix, $D = diag\left([d_i]_{i=1}^n\right)$ is the degree matrix, and $I$ is an identity matrix of appropriate dimension.

**Assumption 2.1.** Throughout this work, it is assumed that the graph representing the multi-agent system is undirected and connected.

The steady-state characteristics of multi-agent system (2.3) is investigated in the sequel.

**Remark 2.1.** Assume that $\mathcal{G}$ is a complete graph (i.e., there is a link between every pair of nodes), representing a multi-agent system (2.1). Then, under the consensus algorithm (2.1), the state vector of the network converges to the equilibrium state $\boldsymbol{x}_{eq} = \begin{bmatrix} \alpha & \alpha & \cdots & \alpha \end{bmatrix}^T$ in the first iteration, where $\alpha \in \mathbb{R}$.

Let $\boldsymbol{x}(0)$ be the initial state in (2.1); the average of the states is then given by:

$$x_{avg}(0) = \alpha = \frac{1}{n}\sum_{i=1}^{n} x_i(0). \tag{2.4}$$

Similarly,

$$x_{avg}(1) = \frac{x_1(1) + x_2(1) + \cdots + x_n(1)}{n}. \tag{2.5}$$

Since $\mathcal{G}$ is a complete graph, $(I + A)$ is a $n \times n$ matrix with all elements equal to one and $(I + D)$ is equal to $nI$. Thus, using (2.3) and (2.5):

$$x_{avg}(1) = \frac{\frac{1}{n}\sum_{i=1}^{n} x_i(0) + \cdots + \frac{1}{n}\sum_{i=1}^{n} x_i(0)}{n} = \alpha. \tag{2.6}$$

Consider now a connected but not complete graph, and define $P$ as:

$$P = (I + D)^{-1}(I + A). \tag{2.7}$$

Let the $(i, j)$ element of the above matrix be denoted by $p_{ij}$. It is straightforward to show that:

$$p_{ij} = \begin{cases} \dfrac{A_{ij}}{d_i + 1}, & i \neq j \\ \dfrac{1}{d_i + 1}, & i = j \end{cases}. \tag{2.8}$$

The above matrix is, in fact, a Markov chain transition matrix, i.e., it satisfies the following conditions [64]:

1. $0 \leq p_{ij} \leq 1, \forall i, j \in \mathbb{N}_n$,
2. $\sum_{j=1}^{n} p_{ij} = 1$ , $\forall i \in \mathbb{N}_n$.

Using the new notation, (2.3) can be written as:

$$\boldsymbol{x}(k + 1) = P\boldsymbol{x}(k). \tag{2.9}$$

Note that $\boldsymbol{x}(k+1)$ depends only on its previous value $\boldsymbol{x}(k)$. Let the *steady-state matrix*

associated with the system matrix $P$ be defined as:

$$P_{ss} := \lim_{k \to \infty} P^k. \tag{2.10}$$

The equilibrium state can then be obtained from the initial state as follows:

$$\boldsymbol{x}_{eq} = P_{ss}\boldsymbol{x}(0). \tag{2.11}$$

An alternative simple solution is given next for $P_{ss}$ using the properties of the transition matrix.

**Lemma 2.1.** *Matrix $P_{ss}$ has identical rows, given by the following vector:*

$$\boldsymbol{p}_{ss} = \frac{1}{d_1 + \cdots + d_n + n} \begin{bmatrix} (d_1 + 1) & \cdots & (d_n + 1) \end{bmatrix}_{1 \times n}. \tag{2.12}$$

*Proof.* The state equation in (2.3) is said to have converged to consensus when all states have the same value. Therefore, it follows from (2.11) that all rows of $P_{ss}$ have to be identical. It is required now to check if a $n \times n$ matrix with the rows given by (2.12) has the characteristics of a transition matrix as well as the steady-state matrix given by (2.10). To this end, the following two conditions are verified:

1. To check the validity of the two conditions of a transition matrix described earlier, it is noted that the denominator in (2.12) is greater than all numerators.

Furthermore, the sum of the elements of vector $\boldsymbol{p}_{ss}$ is:

$$\sum_{i=1}^{n} p_i^{ss} = \frac{1}{d_1 + ... + d_n + n}(d_1 + 1 + \cdots + d_n + 1) = 1, \tag{2.13}$$

where $p_i^{ss}$ is the $i^{th}$ element of vector $\boldsymbol{p}_{ss}$.

2. The steady-state matrix associated with (2.9) remains unchanged if multiplied by $P$ from left or right, i.e.

$$\begin{aligned} P \times P_{ss} &= P_{ss}, \\ P_{ss} \times P &= P_{ss}. \end{aligned} \tag{2.14}$$

The above relations follow immediately from the definition of $P_{ss}$ given by (2.10). By substituting $P$ with (2.7) and the rows of $P_{ss}$ with (2.12), the above relations can be easily verified.

■

**Remark 2.2.** It is to be noted that Lemma 2.1 provides an algebraic solution for matrix $P_{ss}$ defined in (2.10). Note also that once matrix $P_{ss}$ is obtained, the consensus state can be determined as $\alpha = \boldsymbol{p}_{ss}\boldsymbol{x}(0)$

**Remark 2.3.** Assume that graph $\mathcal{G}$ is not complete (note that still by assumption, however, it is strongly connected). Then, it follows directly from (2.12) that under consensus equation (2.9), the state of every agent converges to a weighted average of the

states of different agents, leaning towards the states of agents with more connections in $\mathcal{G}$.

### 2.2.2 Network with Leader

So far, the equilibrium characteristics in a network without a leader was investigated. Now, consider the consensus problem in a network with a leader. Without loss of generality, re-order the indices of the agents such that the index of the leader is 1. Consensus rule (2.2) can then be rewritten as follows [11]:

$$x_i(k+1) = \frac{1}{1+d_i+b_i}\left(x_i(k) + \sum_{j\in\mathcal{N}_i} x_j(k) + b_i x_1\right), \tag{2.15}$$

where $b_i$ is equal to one for the neighbors of the leader, and zero for all other agents.

**Network control rule 1.** *Assume that leader $x_1$ has the same dynamics as the other agents but is also driven by a control input $u(.)$ as follows:*

$$x_1(k+1) = \frac{1}{1+d_1}\left(x_1(k) + \sum_{j\in\mathcal{N}_1} x_j(k)\right) + u(k). \tag{2.16}$$

*All other agents obey consensus rule (2.2) as before.*

It is desired now to find a control input for (2.16) such that the entire network obeys the same rule as (2.2). This would help compare (2.15) with other consensus rules.

**Average-based control:** Let the control input $u$ be given in terms of the states of neighbors as follows:

$$u(k) = x_d - \frac{1}{1+d_1}\left(x_1(k) + \sum_{j\in\mathcal{N}_1} x_j(k)\right), \tag{2.17}$$

where $x_d$ is the desired state of the leader. It can be easily verified that the state of the leader under the above control input turns out to be the same as (2.15).

Note that in the presence of the control input, equation (2.9) is rewritten as:

$$\boldsymbol{x}(k+1) = P\boldsymbol{x}(k) + \boldsymbol{u}(k), \tag{2.18}$$

where $\boldsymbol{u}$ is a $n \times 1$ control vector with all zero elements except $\boldsymbol{u}_1(k) = u(k)$.

**Remark 2.4.** For any initial state in (2.3), there is an equilibrium state $\boldsymbol{x}_{eq}$. Furthermore, in the absence of the control input, the equilibrium state can be determined from the state at any point in time using an equation similar to (2.11). For instance, for any $k \in \mathbb{N}$:

$$\boldsymbol{x}_{eq} = P_{ss}\boldsymbol{x}(0) = P_{ss}P^k\boldsymbol{x}(0) = P_{ss}\boldsymbol{x}(k+1). \tag{2.19}$$

However, in a network with a leader introduced in (2.18), $\boldsymbol{x}_{eq}$ cannot be obtained as above. In fact, the control input changes the equilibrium state at every step unless $\alpha = x_d$ or $u(k) = 0$.

The following theorem sheds some light on the evolution of the equilibrium state

under the average-based control law.

**Theorem 2.1.** *Consider a multi-agent system with a leader described by* (2.18) *and the average-based control law* (2.17)*. Convergence to equilibrium is faster when the leader has a higher number of links.*

*Proof.* Computing the equilibrium state in terms of only the initial state yields:

$$\boldsymbol{x}_{eq}^{0} = P_{ss}\boldsymbol{x}(0), \tag{2.20}$$

where the superscript 0 indicates that $\boldsymbol{x}_{eq}$ is obtained from the state at $k = 0$ (analogous notation will be used in the sequel). Note that the equilibrium state for this multi-agent system would be equal to $\boldsymbol{x}_{eq}^{0}$ if there was no control input applied to the system for $k \geq 0$. In the next step, the equilibrium state is computed in term of the state at $k = 1$ only, which yields:

$$\boldsymbol{x}_{eq}^{1} = P_{ss}\boldsymbol{x}(1) = P_{ss}\left(P\boldsymbol{x}(0) + \boldsymbol{u}(0)\right). \tag{2.21}$$

Vector $\boldsymbol{u}$ has only one non-zero element corresponding to the leader, as noted before. Thus

$$\boldsymbol{x}_{eq}^{1} = P_{ss}\boldsymbol{x}(0) + p_{1}^{ss}u(0)\mathbf{1}_{n} = \boldsymbol{x}_{eq}^{0} + p_{1}^{ss}u(0)\mathbf{1}_{n}, \tag{2.22}$$

where $\mathbf{1}_{n}$ is an all-one vector of appropriate length. Similar to the previous case, if the control input was zero for all $k \geq 1$, then $\boldsymbol{x}_{eq} = \boldsymbol{x}_{eq}^{1}$. Hence, $\boldsymbol{x}_{eq}^{k}$ is equal to its previous value plus a multiple of the control input. Note that vector $\boldsymbol{p}_{ss}$ is given in

(2.12), where each element of it is proportional to the number of links of the agent associated with it. From Remark 2.3, a leader with more links yields a larger coefficient ($p_1^{ss}$) and consequently has a higher impact on the input control in (2.22), which in turn decreases the convergence time. ■

The following example compares the convergence behavior of a system with and without a leader.

**Example 2.1.** Consider a network of 10 mobile robots with the following adjacency matrix

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

Assume that all agents have the same speed but different heading angles as follows:

$$\boldsymbol{x}(0) = \begin{bmatrix} 2 & 4 & 5 & 1 & 7 & 7 & 5 & 3 & 5 & 2 \end{bmatrix}^T \times \frac{\pi}{5}.$$

Three systems are examined here: (i) a network without a leader, (ii) a network with a leader and minimum number of links, and (iii) a network with a leader and maximum number of links. Two different quantities are evaluated: the average state

$$x_{avg}(k) = \sum_{i=1}^{n} x_i(k) \tag{2.23}$$

and the consensus state

$$\alpha^k = \boldsymbol{p}_{ss}\boldsymbol{x}(k) \tag{2.24}$$

for all $k$. Figure 2.1 shows the above quantities for the network without a leader. As expected, $\alpha$ is fixed (i.e., it is independent of $k$). Furthermore, $x_{avg}(k)$ converges to $\alpha$ with an error less than 0.01 after $k = 15$ time steps.

Consider now the network with a leader. From the adjacency matrix $A$, the $6^{th}$ agent has the highest number of links, while the $2^{th}$ and $5^{th}$ agents are the ones with lowest number of links. Let $x_d$ in (2.17) be equal to $7\pi/5$ . Figure 2.2 compares $\alpha^k$ for the two cases. For the case of a leader with the highest number of links, $\alpha^k$ converges to the desired state with an error less than 0.01 in 36 steps which is much faster than the case where the network has a leader with the lowest number of links, which converges

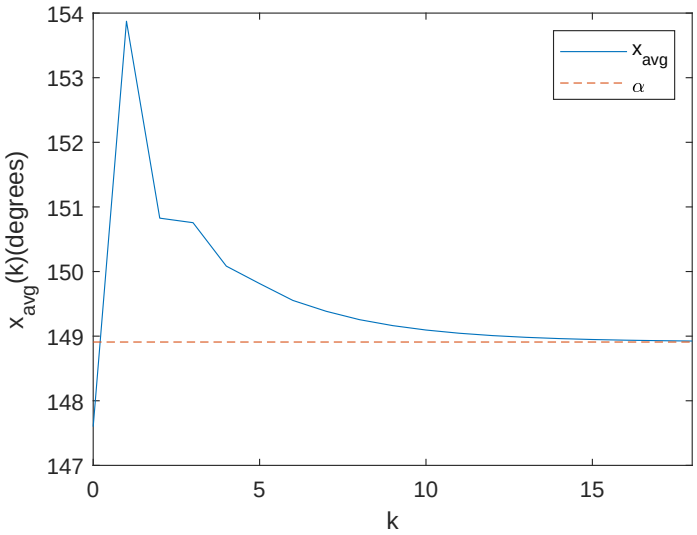


Figure 2.1: Convergence of $x_{avg}(k)$ to $\alpha$ in the network of Example 2.1 without a leader.

after 137 steps. Furthermore, Figure 2.3 shows that the discrepancy of the average state $|x_{avg}(k)-\alpha^k|$ in the network where leader has the highest number of links is much smaller than the case of a leader with lowest number of links.

## 2.3 Follower-Based Control Allocation

In this section, a new consensus rule is proposed with a faster convergence time compared to the one proposed in the previous section.

**Network control rule 2.** *Consider the same system introduced in* (2.18) *but instead of applying the control input only to the leader, assume that the leader dictates a control input to all of its neighbors and itself, and no input is applied to any other agent.*

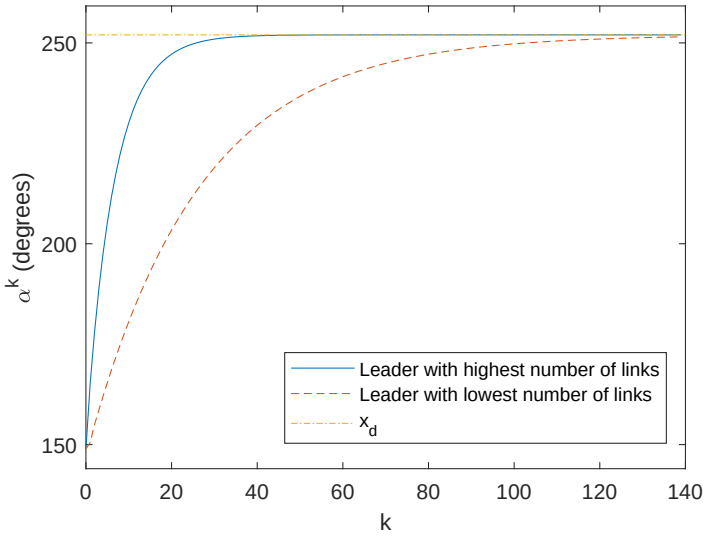


Figure 2.2: Comparison of the convergence of $\alpha^k$ to $x_d$ in the network of Example 2.1 for two scenarios, when a leader with the highest and lowest number of links is selected.

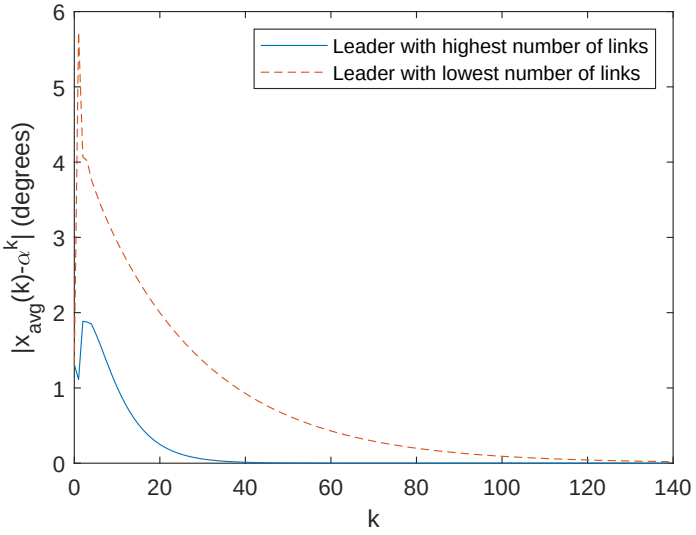


Figure 2.3: Comparison of error $|x_{avg}(k) - \alpha^k|$ in the network of Example 2.1 for two scenarios, when a leader with the highest and lowest number of links is selected

.


Figure 2.4: Two network topologies in which the leader is connected to all agents.

Assume that the leader is connected to all agents as demonstrated in Figure 2.4, where the leader is labeled as node 1. Figure 2.4 (a) shows a *complete graph* in which all agents have the same number of links. Figure 2.4 (b), on the other hand, depicts a *star graph* in which the leader is the only neighbor of any other agent.

**Theorem 2.2.** *Consider a network represented by a complete graph or a star graph. The convergence of the network to consensus is faster under control rule 2, compared to that under control rule 1.*

*Proof.* First, consider a complete graph with $n$ agents and system matrix $P = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$. From (2.18)

$$\boldsymbol{x}(k+1) = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\boldsymbol{x}(k) + \boldsymbol{u}(k). \tag{2.25}$$

Control input (2.17) for a complete graph under control rule 2 is

$$\boldsymbol{u}(k) = \mathbf{1}_n\left(x_d - \frac{1}{n}\mathbf{1}_n^T\boldsymbol{x}(k)\right). \tag{2.26}$$

Substituting (2.26) into (2.25) yields $\boldsymbol{x}(k+1) = x_d \mathbf{1}_n$, which means that the set of eigenvalues of the closed-loop system matrix $P^{r2}$ with control rule 2 is $\Lambda(P^{r2}) = \{0, \ldots, 0\}$. Now, consider the same topology with Network control rule 1. The closed-loop system matrix $P^{r1}$ is

$$P^{r1} = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ \frac{1}{n} & \frac{1}{n} & \ldots & \frac{1}{n} \\ \vdots & & & \vdots \\ \frac{1}{n} & \frac{1}{n} & \ldots & \frac{1}{n} \end{bmatrix}. \tag{2.27}$$

Hence, the set of eigenvalues of $P^{r1}$ is $\Lambda(P^{r1}) = \{\frac{n-1}{n}, 0, \ldots, 0\}$. Comparing the two sets of eigenvalues obtained by the two control rules, it is concluded that the network under control rule 2 converges faster to consensus.

Consider now a star topology with $n$ agents. Since the leader is connected to all agents, the control input is the same as (2.26). The closed-loop system matrix under control rule 2 is then given by

$$P^{r2} = \begin{bmatrix} 0 & 0 & 0 & 0 & \ldots & 0 \\ (\frac{1}{2} - \frac{1}{n}) & (\frac{1}{2} - \frac{1}{n}) & -\frac{1}{n} & -\frac{1}{n} & \ldots & -\frac{1}{n} \\ (\frac{1}{2} - \frac{1}{n}) & -\frac{1}{n} & (\frac{1}{2} - \frac{1}{n}) & -\frac{1}{n} & \ldots & -\frac{1}{n} \\ \vdots & & \vdots & & & \vdots \\ (\frac{1}{2} - \frac{1}{n}) & -\frac{1}{n} & -\frac{1}{n} & \ldots & & (\frac{1}{2} - \frac{1}{n}) \end{bmatrix}. \tag{2.28}$$

The set of eigenvalues of the above matrix is $\Lambda(P^{r2}) = \{0, \frac{-0.5n+1}{n}, 0.5, \ldots, 0.5\}$. Similarly, consider the same network topology with control rule 1. The closed-loop system matrix in this case is

$$P^{r1} = \begin{bmatrix} 0 & 0 & & \ldots & & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & \ldots & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \ldots & 0 \\ \vdots & & & \ddots & & \\ \frac{1}{2} & 0 & 0 & \ldots & & \frac{1}{2} \end{bmatrix} \tag{2.29}$$

with the set of eigenvalues $\Lambda(P^{r1}) = \{0, 0.5, \ldots, 0.5\}$. By comparing the set of eigenvalues of the closed-loop system matrices obtained by control rules 1 and 2 for star topology, it is concluded that the network under control rule 2 converges faster to consensus because $|\frac{-0.5n+1}{n}| < 0.5$, for all $n > 1$. ■

**Example 2.2.** Consider a network with the same adjacency matrix and initial state as in Example 2.1, and let $x_d = 8\pi/5$. It is desired to investigate the convergence of the multi-agent system with leader under network control rule 1 and 2 with average-based control law. In particular, note that the closed-loop dynamics under network control rule 1 in this case is equal to (2.15). Figure 2.5 compares consensus state (2.24) resulted by using the two network control rules in this example. It can be observed from this figure that convergence under the second rule is much faster.

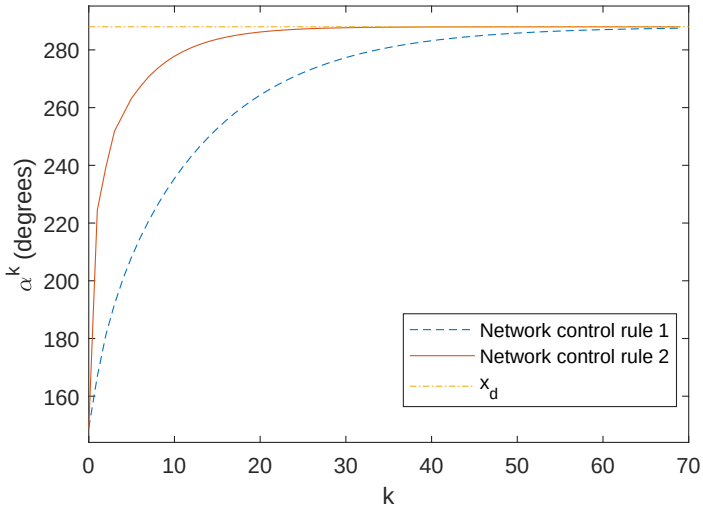


Figure 2.5: Comparison of the convergence of consensus state for the multi-agent system of Example 2.2 under network control rule 1 and 2.

**Remark 2.5.** The control input (and its performance) is subject to the physical limitations of the agents. For instance, the heading angle in mobile robots is between 0 and 360 degrees. Hence, applying a control input larger than a certain value may have a negative impact on the convergence rate.

## 2.4 Conclusion

In this chapter, the equilibrium characteristics in multi-agent networks are studied. Using a Markov chain model, a simple technique for computing the steady-state matrix is provided. It is then shown how the number of links can impact the convergence time. A swarm intelligence-type network control rule is subsequently proposed, which is shown

to have a faster convergence rate compared to conventional control rules. Simulation results confirm the superior performance of the proposed follower-based control allocation strategy.

# Chapter 3

# Distributed Consensus Protocols for Time-Varying Multi-Agent Networks with Improved Convergence Properties

In this chapter, different distributed consensus control strategies are introduced for a multi-agent network with a leader-follower structure. The proposed strategies are based on the nearest neighbor rule, and are shown to reach consensus faster than conventional methods. The results are then extended to the case of a time-varying network with

switching topology and a relatively large number of agents. The convergence performance under the proposed strategies in the case of a time-invariant network with fixed topology is evaluated based on the location of the dominant eigenvalue of the closed-loop system. For the case of a time-varying network with switching topology, on the other hand, the state transition matrix of the system is investigated to analyze the stability of the proposed strategies. Finally, the input saturation in agents' dynamics is considered and the stability of the network under the proposed methods in the presence of saturation is studied. A number of numerical examples are provided to verify the effectiveness of the proposed control schemes.

## 3.1 Introduction

Multi-agent systems have attracted researchers from different disciplines in recent years due to their applications in a variety of engineering and science problems [2–5]. In particular, there has been an increasing interest in the control of multi-agent networks, where it is desired to achieve a global objective such as rendezvous, flocking, formation and consensus, using information locally available to agents [7–11]. Several distributed control schemes are introduced in the literature to achieve the above objectives for networks with small or large number of agents, linear or nonlinear agent dynamics, and fixed or switching topology [17–21].

Graph theory has been used extensively in the literature to analyze various aspects

of multi-agent networks, where each agent is considered as a node and interaction between any pair of agents is represented by an edge [24]. Different structures are used to model the information flow between the agents, e.g., behavior-based, virtual structure and leader follower, and efficient control algorithms are proposed for each [16,23,65]. In particular, the consensus problem is of special importance in this type of network, where all agents are desired to asymptotically reach a common state value by limited exchange of information between neighboring agents [34, 35]. This problem has application in emerging technologies such as the coordination of autonomous vehicles and data fusion in wireless sensor networks [37–39].

Several distributed algorithms are proposed in the literature for consensus control in leader-follower multi-agent networks [11,25,49]. One of the simple yet effective classes of distributed consensus protocols is based on the nearest-neighbor rule [11,17]. According to this rule, every agent updates its state based on the average of the states of its neighbors. As a result, the state of every agent converges to the states of its neighbors, leading all agents to a global convergence under some mild conditions. Nearest-neighbor-based methods are effectively used in both homogeneous networks, where all agents have the same dynamics, and heterogeneous networks, where the dynamic models of different agents are not the same [23,37,53–55]. It is also used in networks with both linear and nonlinear agent dynamics [56–58].

While the methods cited in the previous paragraph are widely used in consensus

control of multi-agent networks, most of them assume that the topology of the network is fixed and its parameters are time-invariant. Many practical multi-agent systems, however, are subject to change during the mission. In addition, the convergence of the existing consensus protocols is not sufficiently fast for some applications. Furthermore, it is often assumed that the control system does not reach a physical limit, i.e., no saturation occurs in the system. However, it is known that in real-world control systems saturation is ubiquitous and any actuator or sensor is subject to saturation [60, 61, 63]. In this chapter, different algorithms are introduced for consensus control of leader-follower multi-agent systems with faster convergence rate compared to the standard protocols. The proposed algorithms are based on the nearest-neighbor rule, and use the local information of a certain subset of agents to generate the control command. The algorithms are then properly modified for the case of switching topology and time-varying network parameters as well as input saturation. The effectiveness of the theoretical findings is verified by several numerical examples.

The structure of the chapter is as follows. Section 3.2 formulates the problem under consideration. In Section 3.3, the convergence rate of the consensus algorithms is investigated using the notion of dominant eigenvalues. The position of the agents in the network structure and their importance in convergence rate is discussed in Section 3.4. Then in Section 3.5, the results are extended to the case of a network with switching topology and time-varying edge weights. In Section 3.6, a new algorithm is proposed

which allows a subset of agents to act as leaders. The algorithms are properly modified in Section 3.7 to account for input saturation. Finally, concluding remarks are provided in Section 3.8.

### Notations

$\mathbb{R}$ is the set of real numbers, $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers, and $\mathbb{R}_{\geq 0}^{n\times n}$ is an $n \times n$ matrix with non-negative real elements, referred to hereafter as a *non-negative matrix*. $\mathbb{N}_n$ denotes the finite set $\{1, \ldots, n\}$ and $\mathbb{Z}$ is the set of whole numbers. Furthermore, $\mathbf{1}_n$ is a vector of all ones.

For a vector $\boldsymbol{x} := [x_i]_{i=1}^n$ with $n$ elements, the $i^{th}$ element is denoted by $x_i$, and for a $n \times n$ matrix $A$, the $(i, j)$ element is represented by $a_{ij}$. For a diagonal matrix $A$, the $(i, i)$ element is denoted by $a_i$, for ease of display. Given a matrix $A$, $\|A\|_\infty$ denotes its infinity norm, which is $\max_i \sum_j |a_{ij}|$.

## 3.2 Consensus Based on the Leader's Command

Consider a multi-agent network in a 2D space, represented by a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V} = \mathbb{N}_n$ is the set of vertices denoting the agents, $\mathcal{E} = \{(i, j) : i, j \in \mathcal{V}\}$ is the set of edges denoting the observation links between the agents, and $W$ is the weight matrix whose $(i, j)$ element $w_{ij}$ is the weight of the edge from vertex $j$ to vertex $i$, and is a non-negative real number. This weight is greater than zero if agent $i$ can observe

agent $j$, and zero otherwise (note that $w_{ij} = w_{ji}$ for any $i, j \in \mathcal{V}$). Assume that $\mathcal{G}$ has no self-loops (i.e. $(i,i) \notin \mathcal{E}$, $\forall i \in \mathcal{V}$) or repeated edges. Denote the leader as agent 1, and assume that all agents are similar in terms of dynamics as well as their functionality and ability to exchange information with others. Inspired by the consensus algorithm proposed in [11], the following update rule is considered:

$$x_i(k+1) = \frac{1}{1+d_i}\left(x_i(k) + \sum_{j\in\mathcal{N}_i} w_{ij}x_j(k) + u_i(k)\right), \qquad i \in \mathbb{N}_n - \{1\},\ k \in \mathbb{Z}, \tag{3.1}$$

where $x_i \in \mathbb{R}$ is the state of the $i^{th}$ agent, defined as its heading angle, $u_i$ is the command generated by the leader for agent $i$, if this agent is in the neighborhood set of the leader, and is zero otherwise, $\mathcal{N}_i$ is the set of neighbors of agent $i$ (i.e., the set of agents observed by agent $i$), and $d_i$ is the degree of agent $i$, defined as:

$$d_i = \sum_{j\in\mathcal{N}_i} w_{ij}. \tag{3.2}$$

It is assumed that the leader has a fixed heading angle, i.e.:

$$x_1(k+1) = x_1(k). \tag{3.3}$$

Using a graph-theoretic approach, (3.1) and (3.3) can be written in the matrix form given below:

$$\boldsymbol{x}(k+1) = P\boldsymbol{x}(k) + B\boldsymbol{u}(k), \tag{3.4}$$

$$P = (I + D)^{-1}(I + W), \tag{3.5}$$

$$B = (I + D)^{-1}, \tag{3.6}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^n$ is the control input vector, $D$ is the degree matrix, which is diagonal and its $(i, i)$ element is equal to $d_i$, $\forall i \in \mathbb{N}_n$, $W = [w_{ij}]_{i,j=1}^n$ is the weight matrix, and $P$, $B$ are $n \times n$ real matrices. Moreover, $I$ is the identity matrix of appropriate dimension.

**Assumption 3.1.** Assume that graph $\mathcal{G}$ is connected. By removing the leader and its links, the graph will be divided into multiple subgraphs (and in the special case, just one subgraph). Each subgraph is, in fact, a weighted undirected graph (and in the special case, it represents just a single agent).

**Remark 3.1.** Matrix $P$ has the following properties [24, 66]:

- $0 \leq p_{ij} \leq 1$
- $\sum_{j=1}^n p_{ij} = 1$

  Define the error vector as:

$$\boldsymbol{e}(k) = x_1 \mathbf{1}_n - \boldsymbol{x}(k). \tag{3.7}$$

Multiplying (3.4) by a negative sign and adding $x_1\mathbf{1}_n$ to both sides yields:

$$x_1\mathbf{1}_n - \boldsymbol{x}(k+1) = x_1\mathbf{1}_n - P\boldsymbol{x}(k) - B\boldsymbol{u}(k). \tag{3.8}$$

It follows from Remark 3.1 that:

$$P\mathbf{1}_n = \mathbf{1}_n. \tag{3.9}$$

Thus, (3.8) can be rewritten as:

$$x_1\mathbf{1}_n - \boldsymbol{x}(k+1) = P\Big(x_1\mathbf{1}_n - \boldsymbol{x}(k)\Big) - B\boldsymbol{u}(k). \tag{3.10}$$

From (3.7), the above equation can be expressed in the following form:

$$\boldsymbol{e}(k+1) = P\boldsymbol{e}(k) - B\boldsymbol{u}(k), \tag{3.11}$$

where the control command $\boldsymbol{u}$, sent by the leader to its neighbors, is defined as:

$$u_i(k) = \begin{cases} e_i(k), & i \in \mathcal{N}_1 \\ 0, & i \notin \mathcal{N}_1 \end{cases}. \tag{3.12}$$

Note that the relative error $e_i$ is available to the leader.

## 3.3 Convergence Rate

In this section, the convergence rate of the error dynamics characterized by (3.11) with the control input defined in (3.12) is investigated. Note that the convergence of the error to zero implies that the state of every agent in (3.4) converges to that of the leader. Subsection 3.3.1 discusses the structure of the system matrix with and without control input.

### 3.3.1 Structure of System Matrix

For the case where $\boldsymbol{u}(k) = 0$, from (3.3) and (3.7) the error of the leader's state is given by:

$$e_1(k) = x_1 - x_1(k) = 0. \tag{3.13}$$

Therefore, one can remove the first row and column of $P$ in (3.11) to obtain the new equation for the error as:

$$\tilde{\boldsymbol{e}}(k+1) = \tilde{P}\tilde{\boldsymbol{e}}(k), \tag{3.14}$$

where $\tilde{\boldsymbol{e}}(k) := [e_i(k)]_{i=2}^{n}$, and $\tilde{P}$ is a matrix characterized in the next remark.

**Remark 3.2.** Let $\tilde{P}_i$ denote the system matrix of the $i$th subgraph obtained by removing the leader, for any $i \in \mathbb{N}_m$, where $m$ is the total number of subgraphs (see Assumption 3.1). Then, the eigenvalues of $\tilde{P}$ in (3.14) are the union of the eigenvalues

of all subgraphs, multiplicities included, i.e.:

$$\Lambda(\tilde{P}) = \{\Lambda(\tilde{P}_1), \ldots, \Lambda(\tilde{P}_m)\}, \tag{3.15}$$

where $\Lambda(.)$ denotes the spectrum of a matrix.

Assume that subgraph $i$ has $n_i$ agents. The corresponding system matrix can be written as follows:

$$\tilde{P}_i = (I_i + D_i)^{-1}(I_i + W_i), \tag{3.16}$$

where $D_i$ and $W_i$ are, respectively, the $n_i \times n_i$ degree and weight matrices.

Now, let the control input $\boldsymbol{u}(k)$ given by (3.12) be applied to the system. From (3.11), the error dynamics for agent $i$ can be expanded as follows:

$$e_i(k+1) = \frac{1}{1+d_i}\left(e_i(k) + \sum_{j\in\mathcal{N}_i} w_{ij}e_j(k) - u_i(k)\right). \tag{3.17}$$

If agent $i$ is a neighbor of the leader, then it results from (3.12) that $u_i(k) = e_i(k)$ and:

$$e_i(k+1) = \frac{1}{1+d_i}\left(\sum_{j\in\mathcal{N}_i} w_{ij}e_j(k)\right). \tag{3.18}$$

Otherwise, $u_i(k) = 0$ and:

$$e_i(k+1) = \frac{1}{1+d_i}\left(e_i(k) + \sum_{j\in\mathcal{N}_i} w_{ij}e_j(k)\right). \tag{3.19}$$

By considering the above two equations, (3.11) can be written in closed-loop form as:

$$\boldsymbol{e}(k+1) = P^{cl}\boldsymbol{e}(k), \tag{3.20}$$

$$P^{cl} = (I+D)^{-1}(H+W), \tag{3.21}$$

where $P^{cl}$ is the closed-loop system matrix and $H$ is a diagonal matrix whose $i^{th}$ diagonal element is $h_i$, for any $i \in \mathbb{N}_n$, where:

$$h_i = \begin{cases} 0, & i \in \mathcal{N}_1 \\ 1, & i \notin \mathcal{N}_1 \end{cases}. \tag{3.22}$$

Again, from (3.13), the first row and column of $P^{cl}$, which correspond to the leader, can be removed. Denote the system matrix after these removals by $\tilde{P}^{cl}$. Hence:

$$\tilde{\boldsymbol{e}}(k+1) = \tilde{P}^{cl}\tilde{\boldsymbol{e}}(k). \tag{3.23}$$

**Remark 3.3.** Let $\tilde{P}_i^{cl}$, $i = 1, \ldots, m$, denote the system matrix of the subgraphs obtained by removing the leader. Then, the set of eigenvalues of $\tilde{P}^{cl}$ is the union of the set of eigenvalues of all of these matrices, multiplicities included, i.e.:

$$\Lambda(\tilde{P}^{cl}) = \{\Lambda(\tilde{P}_1^{cl}), \ldots, \Lambda(\tilde{P}_m^{cl})\}. \tag{3.24}$$

The closed-loop system matrix of subgraph $i$ with $n_i$ agents can be written as:

$$\tilde{P}_i^{cl} = (I_i + D_i)^{-1}(H_i + W_i), \tag{3.25}$$

where $H_i$ is a diagonal matrix with the diagonal elements defined in (3.22).

### 3.3.2 Dominant Eigenvalue

Assume that $A \in \mathbb{R}^{n\times n}$ is a diagonalizable Hurwitz matrix, i.e., there exists an invertible matrix $V$ such that:

$$\Delta = V^{-1}AV, \tag{3.26}$$

where $\Delta$ is a diagonal matrix whose diagonal elements are the eigenvalues of $A$, denoted by $\lambda_1, \ldots, \lambda_n$ with $|\lambda_n| \leq |\lambda_{n-1}| \leq \ldots \leq |\lambda_2| < |\lambda_1| < 1$. Then, any non-zero vector $\boldsymbol{x} \in \mathbb{R}^n$ can be written as [67, 68]:

$$\boldsymbol{x} = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \ldots + c_n\boldsymbol{v}_n, \tag{3.27}$$

where $c_1, \ldots, c_n$ are scalar coefficients and $\boldsymbol{v}_i$ is the $i^{th}$ column of matrix $V$, $i = 1, \ldots, n$. As a result:

$$A\boldsymbol{x} = c_1\lambda_1\boldsymbol{v}_1 + c_2\lambda_2\boldsymbol{v}_2 + \ldots + c_n\lambda_n\boldsymbol{v}_n, \tag{3.28}$$

and consequently:

$$A^k\boldsymbol{x} = c_1\lambda_1^k\boldsymbol{v}_1 + c_2\lambda_2^k\boldsymbol{v}_2 + \ldots + c_n\lambda_n^k\boldsymbol{v}_n. \tag{3.29}$$

The above equation can be written as:

$$A^k \boldsymbol{x} = \lambda_1^k (c_1 \boldsymbol{v}_1 + c_2 (\frac{\lambda_2}{\lambda_1})^k \boldsymbol{v}_2 + \ldots + c_n (\frac{\lambda_n}{\lambda_1})^k \boldsymbol{v}_n). \tag{3.30}$$

Recall that by assumption $\lambda_1$ is the eigenvalue with the greatest magnitude, and hence is referred to as the *dominant* eigenvalue. Thus:

$$\lim_{k \to \infty} A^k \boldsymbol{x} = \lambda_1^k c_1 \boldsymbol{v}_1. \tag{3.31}$$

**Remark 3.4.** It is known that any real symmetric matrix is diagonalizable with real diagonal terms [69].

Given a symmetric network with no leader, there is a similarity transformation to convert matrix $P$ in the state equation into a symmetric matrix $A = T^{-1} P T$ [66, 70]. Since $A$ is symmetric, it has a dominant eigenvalue and so does $P$.

**Remark 3.5.** There exists a similarity transformation matrix $T_i$ for both $\tilde{P}_i$ and $\tilde{P}_i^{cl}$ (defined by (3.16) and (3.25)) which converts them into symmetric matrices. This matrix is diagonal and is given by:

$$T_i = (I_i + D_i)^{-\frac{1}{2}}. \tag{3.32}$$

The resultant transformed matrix is then obtained as:

$$A_i = T_i^{-1} \tilde{P}_i T_i = (I_i + D_i)^{-\frac{1}{2}} (I_i + W_i)(I_i + D_i)^{-\frac{1}{2}}. \tag{3.33}$$

Note that $I_i$ is diagonal and $W_i$ is symmetric. It can be easily verified that multiplying the symmetric matrix $(I_i + W_i)$ by the diagonal matrix $(I_i + D_i)^{-\frac{1}{2}}$ form both sides results in a symmetric matrix. Now, consider $\tilde{P}_i^{cl}$ given in (3.25) and compute $A_i^{cl}$ as follows:

$$A_i^{cl} = T_i^{-1}\tilde{P}_i^{cl}T_i = (I_i + D_i)^{-\frac{1}{2}}(H_i + W_i)(I_i + D_i)^{-\frac{1}{2}}. \tag{3.34}$$

Similarly, $A_i^{cl}$ is a symmetric matrix.

**Remark 3.6.** The subgraphs represented by the system matrices $\tilde{P}_i$ and $\tilde{P}_i^{cl}$ are assumed to be strongly connected. Therefore, these matrices are irreducible, and from the Perron-Frobenius theorem, their dominant eigenvalues are positive [69].

**Theorem 3.1.** *Error dynamics* (3.11) *with control input* (3.12) *converges to zero faster than that with no control input.*

*Proof.* $\tilde{P}^{cl}$ and $\tilde{P}$ are the system matrices of the network with and without control input, respectively. From Remarks 3.2 and 3.3, the eigenvalues of each matrix can be obtained from the eigenvalues of their subgraphs $\tilde{P}_i^{cl}$ and $\tilde{P}_i$, for any $i \in \mathbb{N}_m$. From the similarity transformation (3.34), $A_i^{cl}$ and $\tilde{P}_i^{cl}$ have the same eigenvalues. From Remark 3.4, $A_i^{cl}$ is a symmetric matrix. Assume that $\lambda_{i,1}^{cl}$ is the dominant eigenvalue of $A_i^{cl}$. Thus:

$$\lambda_{i,1}^{cl} = \boldsymbol{v}_{i,1}^T A_i^{cl} \boldsymbol{v}_{i,1} = \boldsymbol{v}_{i,1}^T T^{-1} \tilde{P}_i^{cl} T \boldsymbol{v}_{i,1}, \tag{3.35}$$

where $\boldsymbol{v}_{i,1}$ is the normalized eigenvector ($\|\boldsymbol{v}_{i,1}\|_2 = 1$) associated with the eigenvalue $\lambda_{i,1}^{cl}$.

Substituting (3.25) and (3.32) in the left side of the above equation yields:

$$\boldsymbol{v}_{i,1}^T T^{-1} \tilde{P}_i^{cl} T \boldsymbol{v}_{i,1} = \boldsymbol{v}_{i,1}^T (I_i + D_i)^{-\frac{1}{2}} (H_i + W_i)(I_i + D_i)^{-\frac{1}{2}} \boldsymbol{v}_{i,1}. \tag{3.36}$$

Since $H_i$ is a diagonal matrix with diagonal elements equal to one or zero (as defined in (3.22)), one can write:

$$\begin{aligned} \boldsymbol{v}_{i,1}^T (I_i + D_i)^{-\frac{1}{2}} H_i (I_i + D_i)^{-\frac{1}{2}} \boldsymbol{v}_{i,1} < \\ \boldsymbol{v}_{i,1}^T (I_i + D_i)^{-\frac{1}{2}} I_i (I_i + D_i)^{-\frac{1}{2}} \boldsymbol{v}_{i,1}. \end{aligned} \tag{3.37}$$

Adding $\boldsymbol{v}_{i,1}^T (I_i + D_i)^{-\frac{1}{2}} W_i (I_i + D_i)^{-\frac{1}{2}} \boldsymbol{v}_{i,1}$ to both sides of the above inequality results in:

$$\begin{aligned} \boldsymbol{v}_{i,1}^T (I_i + D_i)^{-\frac{1}{2}} (H_i + W_i)(I_i + D_i)^{-\frac{1}{2}} \boldsymbol{v}_{i,1} < \\ \boldsymbol{v}_{i,1}^T (I_i + D_i)^{-\frac{1}{2}} (I_i + W_i)(I_i + D_i)^{-\frac{1}{2}} \boldsymbol{v}_{i,1}, \end{aligned} \tag{3.38}$$

where:

$$\boldsymbol{v}_{i,1}^T (I_i + D_i)^{-\frac{1}{2}} (I_i + W_i)(I_i + D_i)^{-\frac{1}{2}} \boldsymbol{v}_{i,1} = \boldsymbol{v}_{i,1}^T T^{-1} \tilde{P}_i T \boldsymbol{v}_{i,1}. \tag{3.39}$$

From Remark 3.5, one arrives at:

$$\boldsymbol{v}_{i,1}^T T^{-1} \tilde{P}_i T \boldsymbol{v}_{i,1} = \boldsymbol{v}_{i,1}^T A_i \boldsymbol{v}_{i,1}. \tag{3.40}$$

Since the eigenvectors of $A_i$ are orthogonal and its dominant eigenvalue is positive, it

can be concluded that:

$$\boldsymbol{v}_{i,1}^T A_i \boldsymbol{v}_{i,1} \leq \sup_{\boldsymbol{v}\in\mathbb{R}^{n_i},\|\boldsymbol{v}\|=1} \boldsymbol{v}^T A_i \boldsymbol{v} = \lambda_{i,1}, \tag{3.41}$$

where $\lambda_{i,1}$ is the dominant eigenvalue of $A_i$. Therefore, from (3.35)-(3.41):

$$\lambda_{i,1}^{cl} < \lambda_{i,1}, \qquad \forall i \in \mathbb{N}_{n_i}. \tag{3.42}$$

The dominant eigenvalue of the system matrix of every subgraph can be made smaller by using an appropriate control command. By doing so, the dominant eigenvalue of $\tilde{P}^{cl}$ will be smaller than that of $\tilde{P}$. Hence, $\tilde{P}^{cl}$ converges to zero faster than $\tilde{P}$. ■

**Example 3.1.** Consider a network of 5 mobile robots with the topology and weights depicted in Figure 3.1. The objective is that the state of every agent (which is its heading angle) converges to the state of the leader. Consider the following initial values:

$$\boldsymbol{x}(0) = \begin{bmatrix} 0 & -120 & -62 & -45 & -75 \end{bmatrix}^T.$$

Figures 3.2 and 3.3 show the states of the network without and with the proposed control command, respectively. In both figures, each agent's heading converges to the leader's heading with approximately the same rate that its neighbors' headings do. Figure 3.4 gives the average of the states of all agents at every time step. It can be observed from this figure that under the proposed control command, the average state of the network

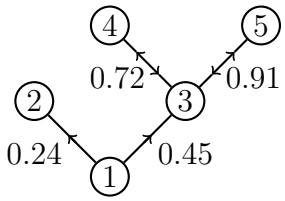


Figure 3.1: Topology of the network of Example 3.1.

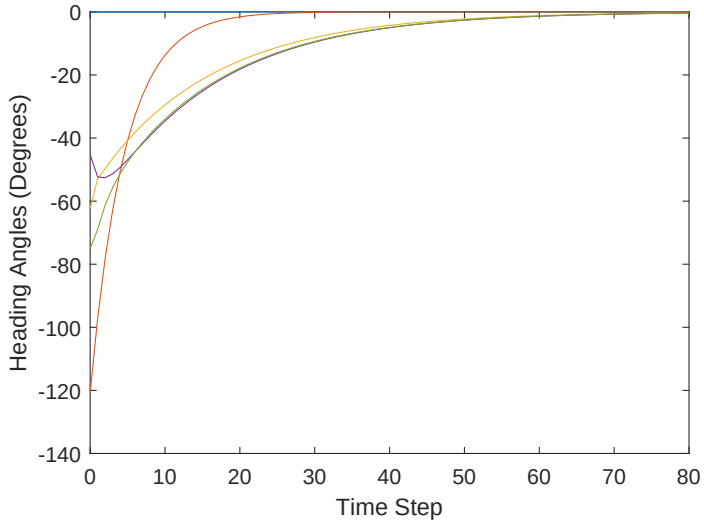


Figure 3.2: State of the network under the standard protocol.

converges to the state of the leader almost twice as fast as that of the network with the standard protocol.

## 3.4 Network Configuration: Leader and Subleaders

Convergence time is highly dependent on the topology of the network. For instance, consider two networks with five agents but different configurations for the leader as depicted in Figures 3.5 and 3.6. Assume that all weights are equal to one, and that both

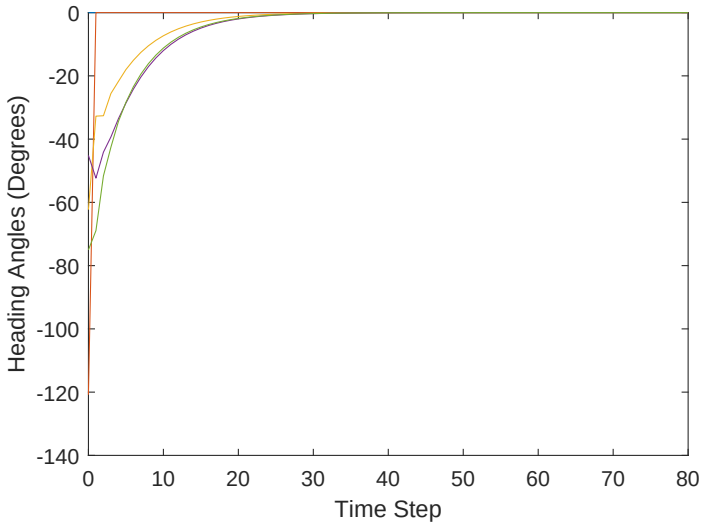


Figure 3.3: State of the network under the proposed control command.

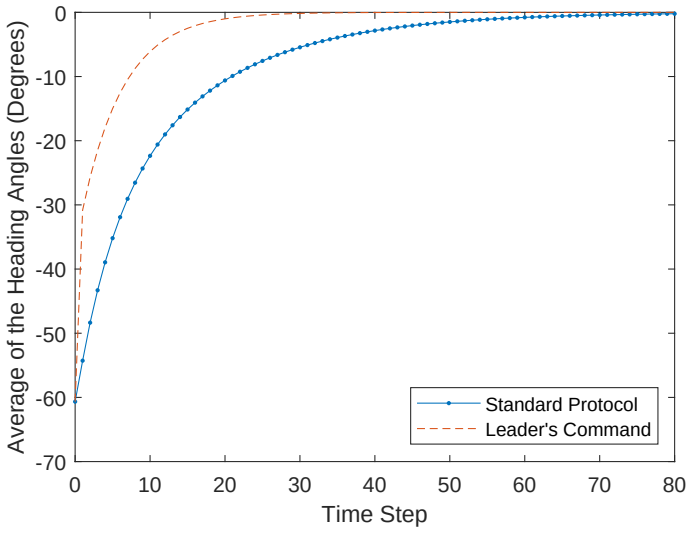


Figure 3.4: Comparison between the average of the states of all agents for the network with the standard protocol and proposed control command.

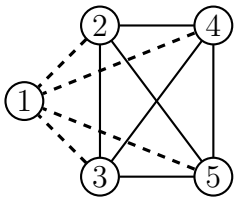


Figure 3.5: Examples of a network with a leader with 4 neighbors. Dashed lines are the leader's connections.

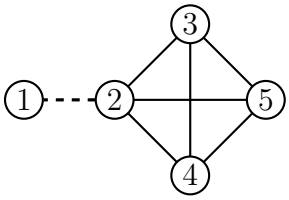


Figure 3.6: Examples of a network with a leader with one neighbor. Dashed lines are the leader's connections.

networks use the control rule (3.12). Convergence time of the network whose leader has four neighbors is almost twice faster than the network whose leader has only one neighbor. This section studies the impact of network configuration on the convergence time. It will be shown how the neighbors of the leader can be employed as so-called "subleaders" to improve the rate of convergence.

Assume that the neighbors of the leader can also send a command to their neighbors. The corresponding control rule, which has a distributed architecture and relies on the relative heading angle, can be described as:

$$u_{ij}(k) = x_1(k) - x_j(k), \quad i \in \mathcal{N}_j,\ j \in \mathcal{N}_1. \tag{3.43}$$

Note that the states of $x_1(k)$ and $x_i(k)$ are both available to agent $j$.

**Assumption 3.2.** A neighbor of the leader cannot act as a subleader for another neighbor of the leader. In other words, for any $i, j \in \mathcal{N}_1$, $u_{ij}(k) = 0$.

**Remark 3.7.** Note that if an agent receives multiple commands from different subleaders, they will all be the same because according to (3.43), only the relative information of the leader and that agent is important, i.e., the subleaders' states play no role in the control process. The control input of agent $i$, described by (3.43), can be considered as an error signal $e_i(k)$. Thus, the control input $\boldsymbol{u}(k)$ resulted from the leader's and subleaders' commands can be rewritten as:

$$u_i(k) = \begin{cases} e_i(k), & i \in \mathcal{N}_1 \cup \mathcal{N}_j,\ j \in \mathcal{N}_1 \\ 0, & \text{otherwise} \end{cases}. \tag{3.44}$$

To investigate the convergence rate, consider the error dynamics (3.11) with the control rule (3.44). The closed-loop system can be expressed as:

$$\boldsymbol{e}(k+1) = P^s \boldsymbol{e}(k), \tag{3.45}$$

$$P^s = (I + D)^{-1}(H^s + W), \tag{3.46}$$

where $P^s$ is the system matrix and $H^s$ is a diagonal matrix whose diagonal element $h_i^s$ is given by:

$$h_i^s = \begin{cases} 0, & i \in \mathcal{N}_1 \cup \mathcal{N}_j,\ j \in \mathcal{N}_1 \\ 1, & \text{otherwise} \end{cases}. \tag{3.47}$$

From (3.13), the first row and column of $P^s$, which correspond to the leader, can be removed. Denote the system matrix after removing the first row and column by $\tilde{P}^s$. Hence:

$$\tilde{\boldsymbol{e}}(k+1) = \tilde{P}^s \tilde{\boldsymbol{e}}(k). \tag{3.48}$$

**Remark 3.8.** Let $\tilde{P}_i^s$ denote the system matrix of the $i^{th}$ subgraph after removing the leader, for any $i \in \mathbb{N}_m$. Then, the set of eigenvalues of $\tilde{P}^s$ is the union of the set of eigenvalues of all the above matrices, multiplicities included, i.e.:

$$\Lambda(\tilde{P}^s) = \{\Lambda(\tilde{P}_1^s), \ldots, \Lambda(\tilde{P}_m^s)\}. \tag{3.49}$$

The closed-loop system matrix of subgraph $i$ with $n_i$ agents can then be written as follows:

$$\tilde{P}_i^s = (I_i + D_i)^{-1}(H_i^s + W_i), \tag{3.50}$$

where $H_i^s$ is a diagonal matrix with the diagonal elements given by (3.47).

**Theorem 3.2.** *The state error of the network, defined by* (3.11)*, converges to zero faster under the control input* (3.44) *than that under the control input* (3.12)*.*

*Proof.* Using an argument similar to Remark 3.5, it follows that $\tilde{P}_i^s$ also has a similarity transformation matrix which converts it to a symmetric form. Consider the system matrices $\tilde{P}_i^{cl}$ and $\tilde{P}_i^s$. From (3.25) and (3.50), both $H_i$ and $H_i^s$ are diagonal matrices

with $H_i$ having more non-zero elements than $H_i^s$. Thus, for any non-zero vector $\boldsymbol{v}$:

$$\begin{aligned}\boldsymbol{v}^T(I_i+D_i)^{-\frac{1}{2}}H_i^s(I_i+D_i)^{-\frac{1}{2}}\boldsymbol{v} <\\ \boldsymbol{v}^T(I_i+D_i)^{-\frac{1}{2}}H_i(I_i+D_i)^{-\frac{1}{2}}\boldsymbol{v}.\end{aligned} \tag{3.51}$$

Hence, using an argument similar to the proof of Theorem 3.1, it can be shown that the dominant eigenvalue of $H_i^s$ is smaller than that of $H_i$. ■

**Example 3.2.** Consider the network given in Example 3.1 with the same initial values. By applying the control command (3.44) to the network, the state of every agent converges to that of the leader as depicted in Figure 3.7. Figure 3.8 provides a comparison between the average of the states of all agents in the network for three scenarios: (i) with the standard protocol; (ii) with the leader's control command (as given in Example 3.1), and (iii) with the leader and subleader's control commands. It can be observed that as expected, the system with leader and subleader's control commands converges faster. It is also observed that the states of the agents converge to the state of the leader.

## 3.5 Network with Switching Topology and Time-Varying Weights

Consider graph $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_{\sigma(k)}, W(k))$, where the set of vertices $\mathcal{V}$ is fixed the same as before but the set of edges $\mathcal{E}_{\sigma(k)}$ is not the same for different values of $\sigma(k) \in \mathcal{S}$, where

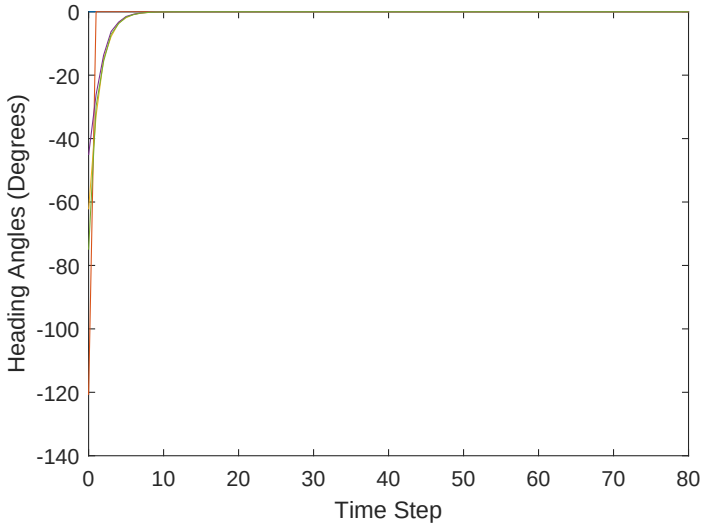


Figure 3.7: States of the agents with control command from the leader and subleader.

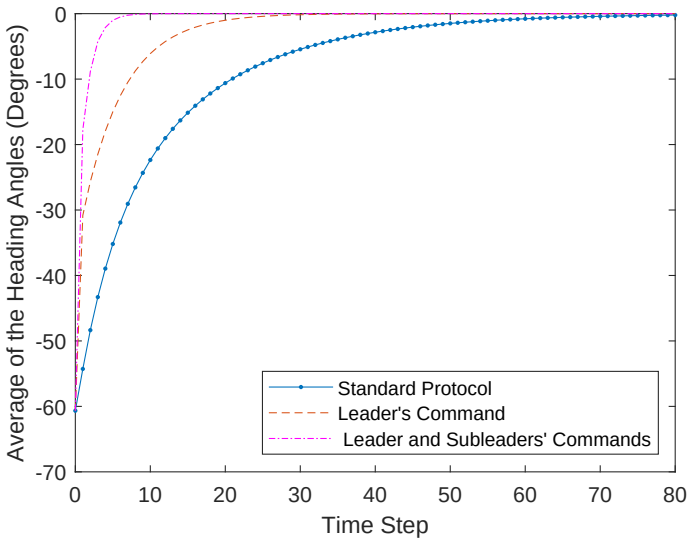


Figure 3.8: Comparison between the network with the standard protocol, with leader's control command, and with leader's and subleader's control commands.

the set $\mathcal{S} = \mathbb{N}_l$ represents the switching topology.

**Assumption 3.3.** Assume that $\mathcal{G}_k$, is connected for all $\sigma(k) \in \mathcal{S}$, and that it satisfies Assumption 3.1. Assume also that the weight matrix $W(k)$ can change at any time step, independently of the topology.

With the switching topology and time-varying weights, (3.1) can be rewritten as:

$$x_i(k+1) = \frac{1}{1+d_i(k)}\left(x_i(k) + \sum_{j\in\mathcal{N}_i(k)} w_{ij}(k)x_j(k) + u_i(k)\right), \qquad i \in \mathbb{N}_n - \{1\},\ k \in \mathbb{Z}, \tag{3.52}$$

where $\mathcal{N}_i(k)$ is the set of neighbors of agent $i$ at time $k$, $w_{ij}(k) \in \mathbb{R}_{\geq 0}$ is the weight of the link between agent $i$ and $j$ at time $k$, and $d_i(k)$ is the degree of agent $i$ at time $k$ given by:

$$d_i(k) = \sum_{j\in\mathcal{N}_i(k)} w_{ij}(k). \tag{3.53}$$

Let the leader have the same dynamics as (3.3). Using a graph-theoretic approach, (3.52) and (3.3) can be written in matrix form as:

$$\boldsymbol{x}(k+1) = P(k)\boldsymbol{x}(k) + B(k)\boldsymbol{u}(k), \tag{3.54}$$

$$P(k) = (I + D(k))^{-1}(I + W(k)), \tag{3.55}$$

$$B(k) = (I + D(k))^{-1}, \tag{3.56}$$

where $P(k) \in \mathbb{R}^{n\times n}_{\geq 0}$ is the system matrix, $B(k) \in \mathbb{R}^{n\times n}_{\geq 0}$ is the input channel matrix, $D(k)$ is the degree matrix, which is diagonal, and $W(k)$ is the weight matrix, all at time $k$. Note that Remark 3.1 still holds for $P(k)$ at all times. From (3.7), the error dynamics can be written as:

$$\boldsymbol{e}(k+1) = P(k)\boldsymbol{e}(k) - B(k)\boldsymbol{u}(k). \tag{3.57}$$

Consider the control command obtained in (3.44). The closed-loop system can be obtained as:

$$\boldsymbol{e}(k+1) = P^s(k)\boldsymbol{e}(k), \tag{3.58}$$

$$P^s(k) = (I + D(k))^{-1}(H^s(k) + W(k)), \tag{3.59}$$

where $P^s(k)$ is the system matrix and $H^s(k)$ is a diagonal matrix whose $(i,i)^{th}$ element is given by:

$$h_i^s(k) = \begin{cases} 0, & i \in \mathcal{N}_1(k) \cup \mathcal{N}_j(k),\ j \in \mathcal{N}_1(k) \\ 1, & \text{otherwise} \end{cases}. \tag{3.60}$$

Similarly, from (3.13), the first row and column of $P^s(k)$, which correspond to the leader, can be removed. Denote the system matrix after these removal by $\tilde{P}^s(k)$. Hence:

$$\tilde{\boldsymbol{e}}(k+1) = \tilde{P}^s(k)\tilde{\boldsymbol{e}}(k), \tag{3.61}$$

where:

$$\tilde{P}^s(k) = (\tilde{I} + \tilde{D}(k))^{-1}(\tilde{H}^s(k) + \tilde{W}(k)). \tag{3.62}$$

**Theorem 3.3.** *Consider a set of graphs $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_l\}$, where $\mathcal{G}_i$, $i \in \mathbb{N}_l$, is composed of a set of jointly connected graphs. The error dynamics (3.61) for $\mathcal{G}$ is globally asymptotically stable.*

*Proof.* Let $\mathcal{G}_i = \{\mathcal{G}_{i,k}, \ldots, \mathcal{G}_{i,k_0}\}$ correspond to the state transition matrix $\Phi_i(k+1, k_0) = \tilde{P}^s(k) \times \ldots \times \tilde{P}^s(k_0)$. On the other hand, from the properties of the discrete-time state transition matrix, $\Phi_i(k_0, k_0) = I$ and $\Phi_i(k+1, k_0) = \tilde{P}^s(k)\Phi_i(k, k_0)$ [71]. First, consider $\Phi_i(k_0+1, k_0) = \tilde{P}^s(k_0)$. From (3.60) and (3.62), the $i^{th}$ row sum of $\tilde{P}^s(k_0)$ can be written as:

$$\sum_{j=1}^{n-1} \tilde{p}^s_{ij}(k_0) \begin{cases} < 1, & i \in \mathcal{N}_1(k_0) \cup \mathcal{N}_j(k_0),\ j \in \mathcal{N}_1(k_0) \\ = 1, & \text{otherwise} \end{cases}. \tag{3.63}$$

So, $\Phi_i(k_0+1, k_0)\mathbf{1}_{(n-1)}$ can be considered as a vector of ones, except for the neighbors of the leader and the neighbors of the neighbors of the leader, which are less than one. At the next time step, $\Phi_i(k_0+2, k_0) = \tilde{P}^s(k_0+1)\Phi_i(k_0+1, k_0)$ and the $i^{th}$ row sum is:

$$\begin{aligned} \sum_{j=1}^{n-1} \phi_{ij}(k_0+2,k_0) = \tilde{p}^s_{i1}(k_0+1) \sum_{j=1}^{n-1} \tilde{\phi}_{1j}(k_0+1, k_0) + \ldots \\ + \tilde{p}^s_{i(n-1)}(k_0+1) \sum_{j=1}^{n-1} \tilde{\phi}_{(n-1)j}(k_0+1, k_0). \end{aligned} \tag{3.64}$$

The above row sum is less than 1 when $\sum_{j=1}^{n-1} \phi_{tj}(k_0+1, k_0) < 1$ for any $t \in \mathcal{N}_i(k_0+1)$ or when $\sum_{j=1}^{n-1} \tilde{p}^s_{ij}(k_0+1) < 1$ (or both). From the connectivity of the graph and equation (3.64), it can be concluded that the number of elements of vector $\Phi_i(k_0+2, k_0)\mathbf{1}_{(n-1)}$ which are less than one is greater than the number of elements of vector

$\Phi_i(k_0+1,k_0)\mathbf{1}_{(n-1)}$ which are less than one. Similarly, it can be concluded that as time increases, the number of elements which are less than one is growing. Since the graph is jointly connected, $k > n-1$ guarantees that all row sums are less than one, and that there exists a positive real number $\delta < 1$ such that:

$$\|\Phi_i(k,k_0)\|_\infty < \delta, \tag{3.65}$$

where the norm in the left side of (3.65) varies with $k$, and $\delta$ depends on the network topology. Let $\Phi(k_l,k_0)$ be the state transition matrix corresponding to $\mathcal{G}$; then $\Phi(k_l,k_0) = \Phi_l(k_l,k_{(l-1)}) \times \ldots \times \Phi_1(k_1,k_0)$. Note that:

$$\lim_{l\to\infty} \|\Phi(k_l,k_0)\|_\infty \leq \lim_{l\to\infty} \delta^l = 0. \tag{3.66}$$

Therefore, the system is globally asymptotically stable [11, 72]. ■

**Example 3.3.** Consider a network of 10 mobile robots. The topology of the network and its weight matrix are assumed to be time-varying. The weights are randomly selected between 0 and 1 at each time step. The generalized algebraic connectivity [73] of the network at different times is depicted in Figure 3.9. Note that the maximum possible generalized algebraic connectivity for this network is equal to one. Figures 3.10−3.12 show the states of the agents with the standard protocol, with control command from the leader (3.12), and with control command from the leader and subleaders (3.44). These figures demonstrate the effectiveness of the proposed control strategies, analogously to

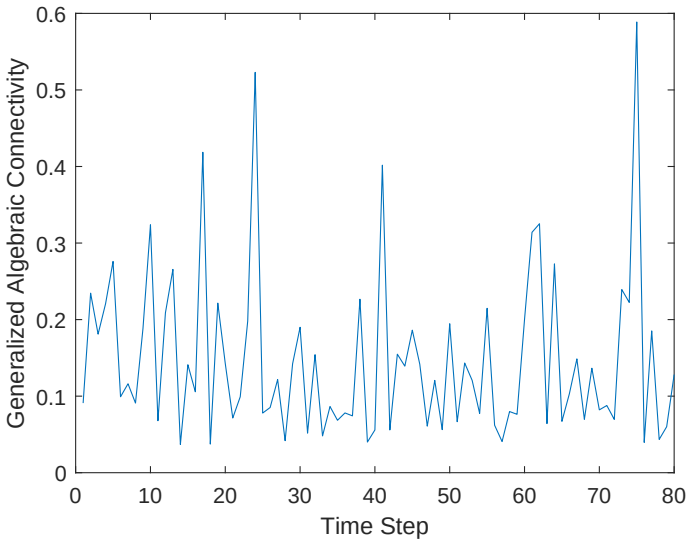


Figure 3.9: Generalized algebraic connectivity of the network of Example 3.3.

the case of fixed topology, in asymptotic convergence of the heading angles of the agents to that of the leader. Figure 3.13, on the other hand, shows the average of the states of the agents under the above-mentioned control actions. It can be observed from this figure that the system with the leader and subleaders' control command converges faster than the other two.

## 3.6 Leader Centric Connectivity

The effectiveness of the control command (3.44) strongly depends on the size of the neighbor set of the leader and its neighbors; if the leader and its neighbors have a relatively large number of neighbors compared to the total number of agents, then the control command (3.44) is more effective, otherwise, the convergence time under the

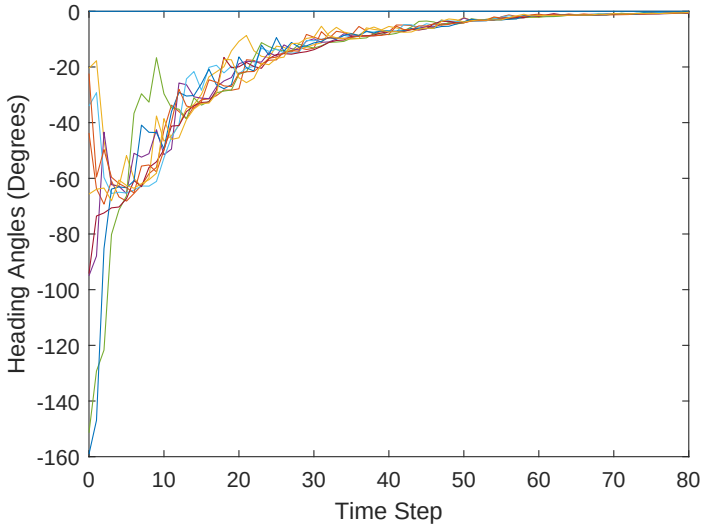


Figure 3.10: States of the agents in the time-varying network of Example 3.3 with the standard protocol.

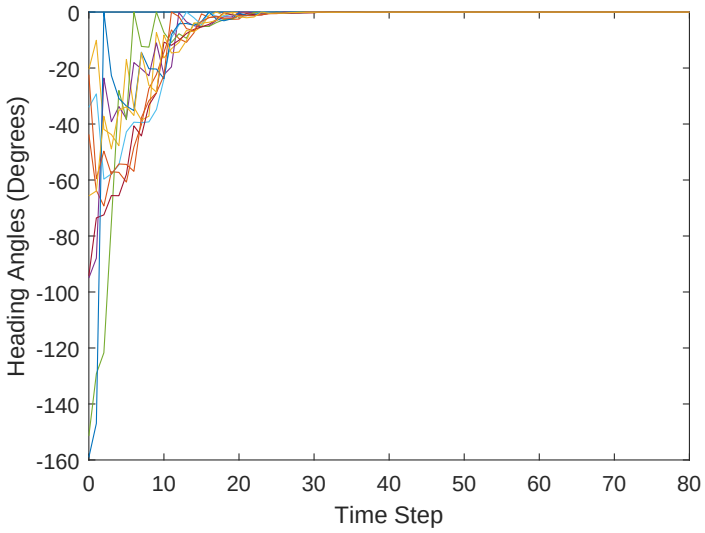


Figure 3.11: States of the agents in the time-varying network of Example 3.3 with the proposed control command from the leader.

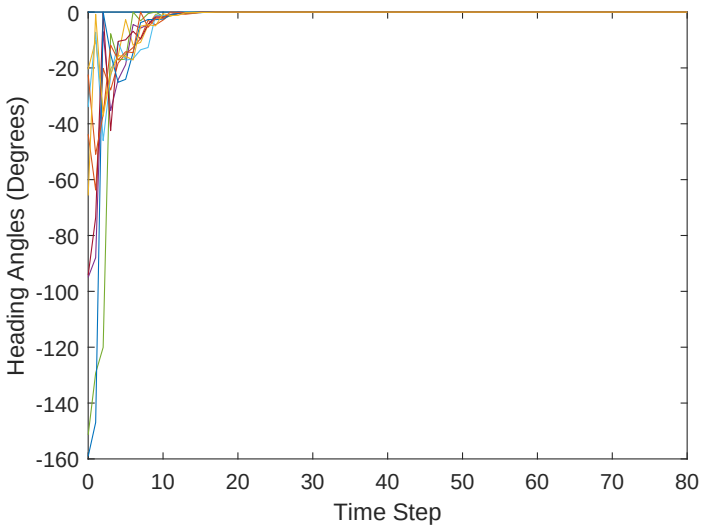


Figure 3.12: States of the agents in the time-varying network of Example 3.3 with the proposed control command from the leader and subleaders.

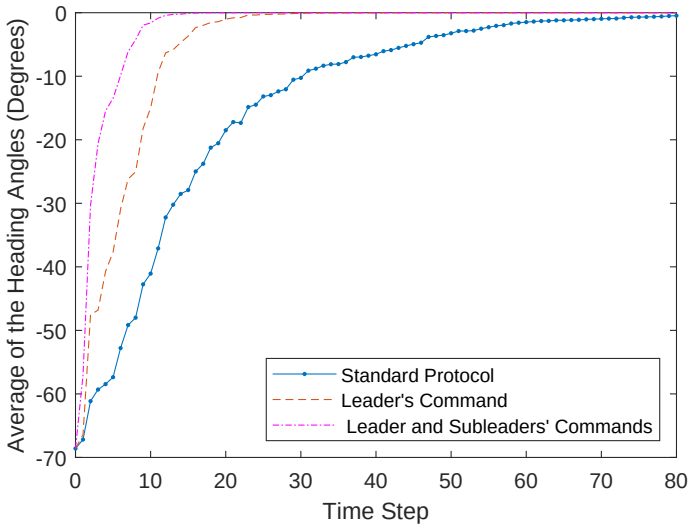


Figure 3.13: Comparison between the time-varying network of Example 3.3 with the standard protocol, with leader's control command, and with leader and subleaders' control commands.

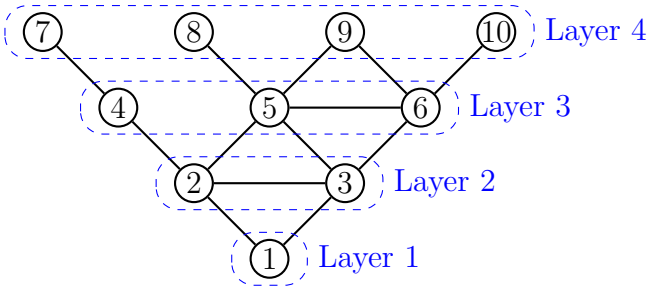


Figure 3.14: An example a network with multiple layers of hierarchy with respect to the leader.

control command (3.44) will also be relatively large. In this section, a decentralized control algorithm is proposed to address this problem.

Assume that all agents have the ability to send and receive control commands (act both as a leader and/or follower). Since it is desired to have a distributed control structure, the control command should be based on the relative state information. To clarify the concept, consider a network of 10 agents with the topology shown in Figure 3.14. This network has different layers of hierarchy, with the first one corresponding to the leader, the second one the neighbors of the leader and so on. In other words, the agents of each layer have the same number of links in their shortest path to the leader. For example, the shortest path from the agents of layer 3 to the leader is two and this number for the agents of layer 4 is three. Note that the agents of higher layers receive the information of the leader with more delay. Since the agents of each layer have access to the information of both upper and lower layers, they can share the information of the agents of the lower layer with the agents of the higher layer.

**Assumption 3.4.** The agents of each layer can only send commands to their neighbors

**Algorithm 1:** Control command for agent $i$

**if** $u_i(k-1) \neq 0$ **then**
  $\bar{x} = \frac{\sum_{r_i} x_{r_i(k)}}{c_i(k)}, \quad r_i \in \mathcal{N}_i(k) \cup \mathcal{L}_{(f_i-1)}(k)$
  $u_{ji}(k) = \bar{x} - x_j(k), \quad j \in \mathcal{N}_i(k) \cup \mathcal{L}_{(f_i+1)}(k)$
**else**
  $u_{ji}(k) = 0, \quad \forall j \in \mathcal{N}_i(k)$
**end**

in the higher layer.

**Assumption 3.5.** If an agent receives commands from different agents of the lower layer, then the average of the received commands is used.

Assume that $u_{ij}$ is the command that agent $i$ receives from agent $j$. Then, the control input is:

$$u_j(k) = \frac{\sum_i u_{ji}(k)}{c_j(k)}, \quad i \in \mathcal{N}_j, \tag{3.67}$$

where $c_j(k)$ is the total number of commands that agent $j$ receives at time $k$. Now, let $f_i$ denote the layer number corresponding to agent $i$ and $\mathcal{L}_{f_i}$ be the set of all agents in layer $f_i$. Then, the control commands that agent $i$ sends to its neighbors are obtained by using Algorithm 1. Agent $i$ can send commands to its higher layer neighbors only if it receives commands at the previous time step. Furthermore, if agent $i$ is a neighbor of multiple agents at the lower layer, then the average of their states is used as the leader state to compute the control command for the next layer. Note that the main leader still uses (3.12) as the control command.

**Remark 3.9.** Using Algorithm 1, the same control rule is applied to all agents except for the leader. This simplifies the implementation of the algorithm.

Consider $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_{\sigma(k)}, W(k))$, defined in Section 3.5, and let the agent dynamics be described by equation (3.54). To write the network equations based on the error dynamics, the control commands given by Algorithm 1 should be expressed in terms of the error vector. Adding $x_1 - x_1$ to the control command yields:

$$u_{ji}(k) = \frac{\sum_{r_i} x_{r_i}(k)}{c_i(k)} - x_1 + x_1 - x_j(k), \tag{3.68}$$

where $i$, $j$, $r_i$ are the same indices used in Algorithm 1. Considering the following error dynamics:

$$u_{ji}(k) = e_j(k) - \frac{\sum_{r_i} e_{r_i}(k)}{c_i(k)}. \tag{3.69}$$

Substituting (3.69) in (3.67) results in:

$$u_j(k) = e_j(k) - \sum_i \frac{\sum_{r_i} e_{r_i}(k)}{c_i(k)c_j(k)}. \tag{3.70}$$

Note that the second term in the right side of the above equation is a summation over the average error of the agents that act as leaders for layer $f_i$.

**Remark 3.10.** If agent $i$ is in layer 2, then $e_{r_i}(k) = e_1 = 0$. One can easily verify that in this case using the proposed algorithm for the leader's neighbor will result in the same command as in Section 3.4.

For stability analysis, let the control input $\boldsymbol{u}(.)$ be divided into two vectors. The first vector $\boldsymbol{u}_1(.)$ is generated based on the state of the leader and the neighbors of the leader. Thus, (3.70) and (3.44) give the same value for $\boldsymbol{u}_1(.)$. The second vector $\boldsymbol{u}_2(.)$ is generated based on the states off all other agents. Note that those elements of $\boldsymbol{u}_2(.)$ which correspond to the leader and its neighbors are zero. Hence, (3.57) can be rewritten as:

$$\boldsymbol{e}(k+1) = P^s(k)\boldsymbol{e}(k) - B(k)\boldsymbol{u}_2(k). \tag{3.71}$$

By excluding the leader from the above equation, it can be written as:

$$\tilde{\boldsymbol{e}}(k+1) = \tilde{P}^s(k)\tilde{\boldsymbol{e}}(k) - \tilde{B}(k)\tilde{\boldsymbol{u}}_2(k). \tag{3.72}$$

where $\tilde{P}^s(.)$ was defined earlier, $\tilde{B}(.)$ is obtained by removing the first row and column of $B(.)$, and $\tilde{\boldsymbol{u}}_2(.)$ is obtained by removing the first element of $\boldsymbol{u}_2(.)$. Note that $B(.)$ is a diagonal matrix whose first diagonal element $b_{11}$ corresponds to the leader. From (3.3), the leader has a fixed dynamics; therefore, $b_{11}$ is 0, which makes the first row and column of $B(.)$ equal to zero.

**Theorem 3.4.** *Consider a network represented by a set of graphs $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_l\}$ where the subgraph $\mathcal{G}_i$ is composed of a set of jointly connected graphs. The error dynamics of the network states (3.72) is globally asymptotically stable.*

*Proof.* To prove the theorem, a procedure similar to the proof of Theorem 3.3 is used

here. Assume that $\hat{P}(k)$ is the closed-loop system matrix corresponding to (3.72), and let $j \in \mathcal{N}_1 \cup \mathcal{N}_i$, $\forall i \in \mathcal{N}_1$. Then, $\hat{p}_{jt}(k) = \tilde{p}^s_{jt}(k)$ and $\sum_t \hat{p}_{jt(k)} < 1$, for all $t \in \mathbb{N}_{n-1}$. On the other hand, for $j \notin \mathcal{N}_1 \cup \mathcal{N}_i$, $\forall i \in \mathcal{N}_1$, $\tilde{e}_j(k+1)$ from (3.72) can be expanded as follows:

$$\begin{aligned}\tilde{e}_j(k+1) =& \tilde{p}^s_{j1}(k)\tilde{e}_1(k) + \ldots + \tilde{p}^s_{jj}(k)\tilde{e}_j(k) + \ldots + \tilde{p}^s_{j(n-1)}(k)\tilde{e}_{n-1}(k) \\ & - \frac{1}{1+d_j(k)}\tilde{e}_j(k) + \frac{1}{1+d_j(k)} \times \frac{1}{m}\left(\tilde{e}_{r_{i1}}(k) + \ldots + \tilde{e}_{r_{im}}(k)\right),\end{aligned} \tag{3.73}$$

where $m = c_i(k)c_j(k)$. From (3.5), it results that $\tilde{p}^s_{jj}(k) = \frac{1}{1+d_j(k)}$, which means that the coefficient of $\tilde{e}_j(k)$ in the above equation is equal to zero (i.e. $\hat{p}_{jj}(k) = 0$). Note that the sum of the coefficients of $\tilde{e}_t(k)$ over all $t \in \mathbb{N}_{n-1}$ in the control command (3.70) is $1 - \frac{1}{m} \times m = 0$, i.e. $\sum_{t=1}^{n-1} \hat{p}_{jt}(k) = \sum_{t=1}^{n-1} \tilde{p}^s_{jt}(k) = 1$. Therefore, $\hat{P}(k)$ is a non-negative matrix with the following properties:

$$\sum_{t=1}^{n-1} \hat{p}_{jt}(k) \begin{cases} < 1, & j \in \mathcal{N}_1(k) \cup \mathcal{N}_i(k),\ i \in \mathcal{N}_1(k) \\ = 1, & \text{otherwise} \end{cases}, \tag{3.74}$$

which is the same as (3.63). Furthermore, $\hat{P}(k)$ includes all the connections of $\tilde{P}^s(k)$ as well as the new links added by the control input. Thus, similar to the proof of Theorem 3.3, $k > n-1$ guarantees that the state transition matrix $\Phi_i(k, k_0)$ corresponding to $\mathcal{G}_i$ has an infinity norm less than $\delta$, where $\delta$ is a positive real scalar less than one. Hence, $\lim_{l\to\infty} \|\Phi(k_l, k_0)\|_\infty$ converges to zero and this completes the proof. ■

**Example 3.4.** Consider a network of 100 mobile robots. The topology is updated every 10 steps and the weight matrix is updated at every step (the weights are randomly selected between 0 and 1). The generalized algebraic connectivity at each step is depicted in Figure 3.15 (note that the maximum possible generalized algebraic connectivity for the network is equal to one). Figures 3.16−3.19 show the states of the network with no control command, with leader's control command, with leader's and subleaders' control commands, and with the control command of any agent acting as the leader, respectively. In all four figures, the agents preserve their consensus in moving towards the leader. Figure 3.20 shows the average of all states for four different control strategies. Under the control strategy that all agents can act as the leader, the followers converge to the leader faster than any other strategy. Although the network with leader's and subleaders' control commands does not have the fastest convergence, it has satisfactory performance.

**Example 3.5.** In this example, a comparison is provided between some existing methods and the proposed strategies. The methods considered for this comparison are as follows:

1. The method in [11], which is the basis of the strategies proposed in this thesis, and can be formulated as follows:

$$x_i(k+1) = \frac{1}{1+d_i+b_i}\left(x_i(k) + \sum_{j\in\mathcal{N}_i} x_j(k) + b_i x_0\right) \tag{3.75}$$

where $x_0$ is the leader's state, and $b_i$ is equal to 1 if agent $i$ is a neighbor of the

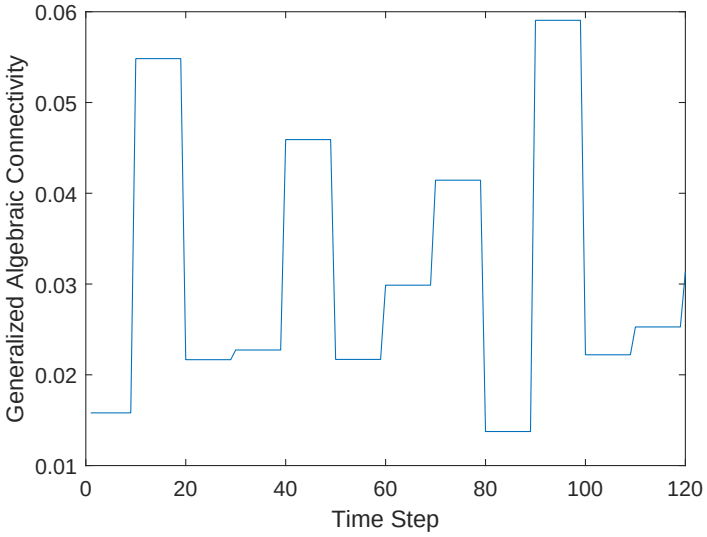


Figure 3.15: Generalized algebraic connectivity of the system in Example 3.4.

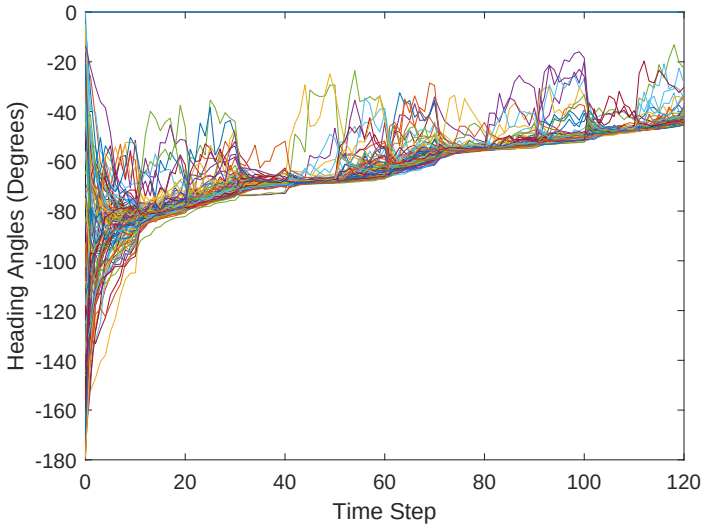


Figure 3.16: States of the agents in the time-varying network of Example 3.4 with 100 agents under the standard protocol.

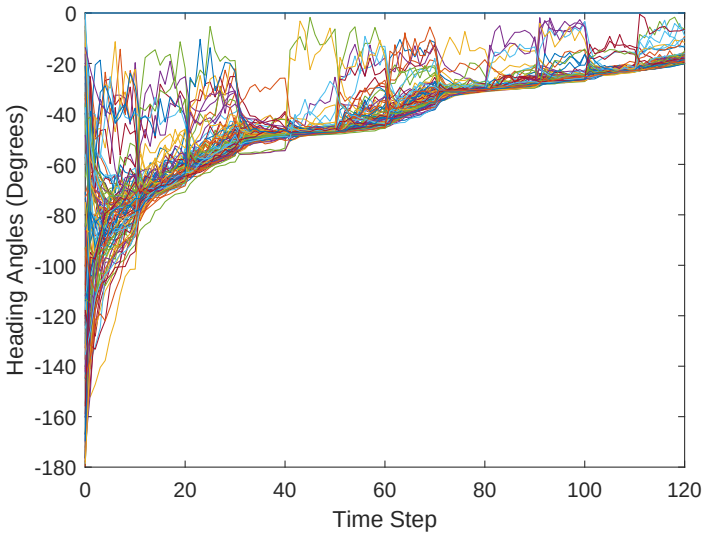


Figure 3.17: States of the agents in the time-varying network of Example 3.4 with 100 agents under leader's control command.

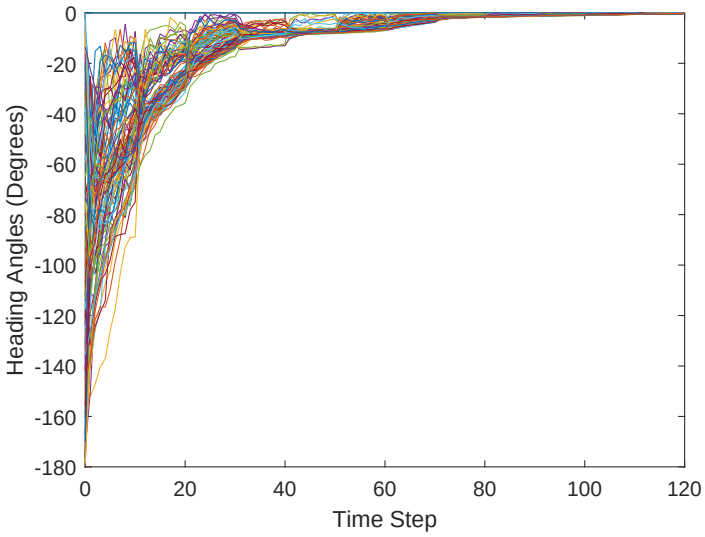


Figure 3.18: States of the agents in the time-varying network of Example 3.4 with 100 agents under leader's and subleaders' control commands.

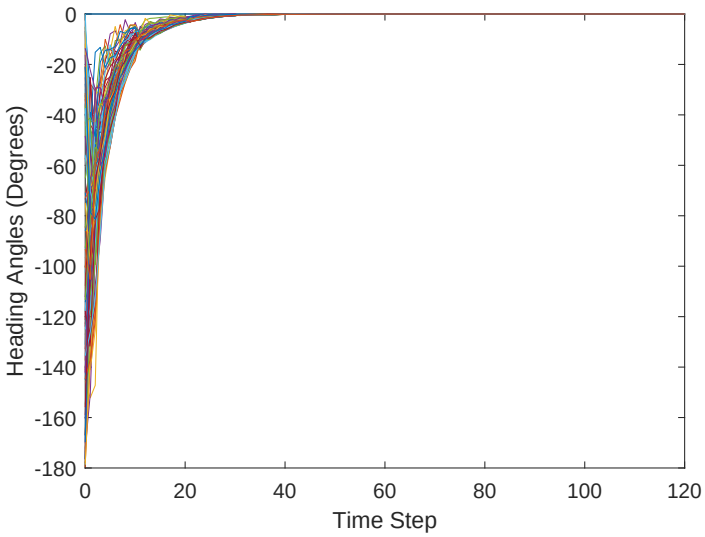


Figure 3.19: States of the agents in the time-varying network of Example 3.4 with 100 agents when they can all act as leader and/or follower.

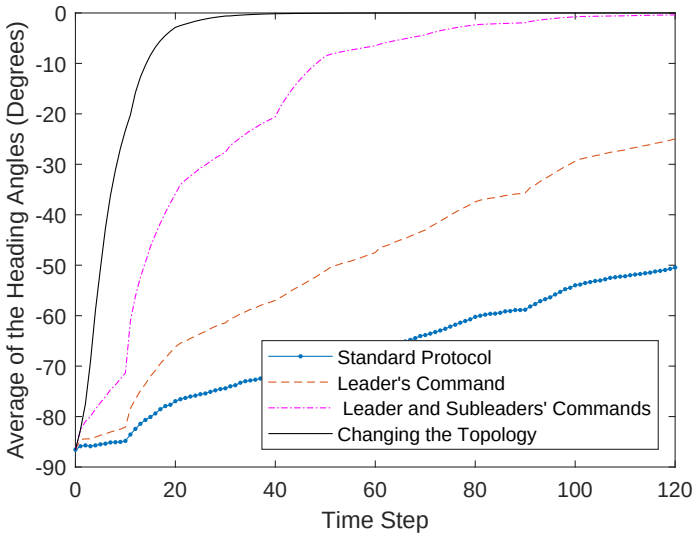


Figure 3.20: Comparison between different control strategies for the network of Example 3.4 with switching topology and time-varying weights.

leader and is equal to zero otherwise.

2. The method in [25], in which:

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in \mathcal{N}_i} a_{ij}\Big(x_j(k) - x_i(k)\Big) \tag{3.76}$$

where $\epsilon$ is the step size bounded as follows:

$$0 < \epsilon < \frac{1}{\delta} \tag{3.77}$$

$$\delta = \max_i \sum_{j \neq i} (a_{ij}) \tag{3.78}$$

To increase the convergence rate in this comparison, $\epsilon$ is chosen equal to $\frac{1}{\delta} - 0.001$.

3. The method in [74], wherein a time-varying weight algorithm is used to achieve fast convergence as follows::

$$x_i(k+1) = w_{ii}(k)x_i(k) + \sum_{j \in \mathcal{N}_i} w_{ij}(k)x_j(k) \tag{3.79}$$

where $w_{ij}(k)$ is the weight, and is obtained as:

$$w_{ij}(k) = \begin{cases} 1/\Big(1 + \max(d_i, d_j)\Big) & (i,j) \in \mathcal{E} \\ 1 - \sum_{l \in \mathcal{N}_i} w_{il} & i = j \\ 0 & \text{otherwise.} \end{cases} \tag{3.80}$$

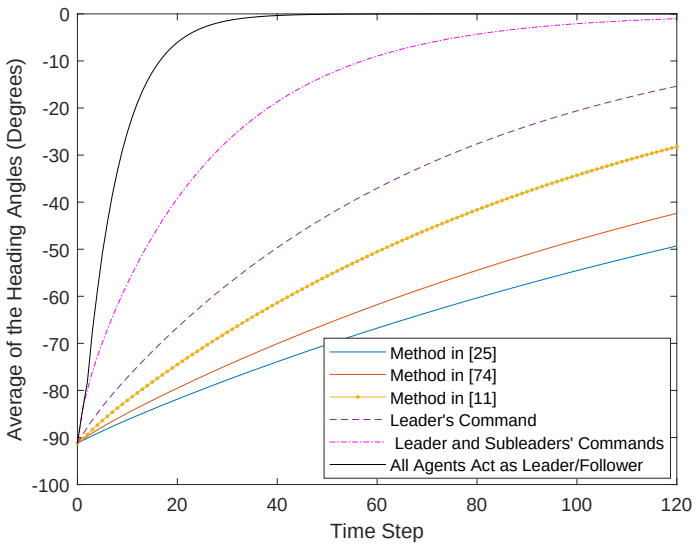


Figure 3.21: A comparison between the proposed control strategies and three existing methods.

To compare all the methods, a network of 50 mobile robots with fixed topology is considered. The weights of all links for all methods are assumed to be equal to one, except the method in [74] whose corresponding weights are obtained by (3.80). Figure 3.21 shows the average of the heading angles for the network under the proposed control strategies and that obtained by using the three methods noted earlier. The results confirm that the proposed strategies yield faster convergence compared to all three methods.

## 3.7 Network with Input Saturation

In this section, it is desired to incorporate input saturation into the dynamic equation (3.52). The stability of the system under leader's and subleader's commands (3.44) is

investigated.

To consider the input saturation in (3.52), the equation is rewritten as follows:

$$\begin{aligned} x_i(k+1) =& x_i(k) - x_i(k) + \frac{1}{1+d_i(k)} \\ & \left( x_i(k) + \sum_{j \in \mathcal{N}_i(k)} w_{ij}(k) x_j(k) + u_i(k) \right) \\ =& x_i(k) + \frac{1}{1+d_i(k)} \\ & \left( -d_i(k) x_i(k) + \sum_{j \in \mathcal{N}_i(k)} w_{ij}(k) x_j(k) + u_i(k) \right), \end{aligned} \qquad i \in \mathbb{N}_n - \{1\},\ k \in \mathbb{Z}. \tag{3.81}$$

The above equation can be expressed as:

$$x_i(k+1) = x_i(k) + \boldsymbol{\ell}_i(\boldsymbol{x}(k), u_i(k), k), \tag{3.82}$$

where $\boldsymbol{\ell}_i(\boldsymbol{x}(k), u_i(k), k)$ is a scalar function associated with agent $i$, defined as:

$$\begin{aligned} \boldsymbol{\ell}_i(\boldsymbol{x}(k), u_i(k), k) = & \frac{1}{1+d_i(k)} \\ & \left( \sum_{j \in \mathcal{N}_i(k)} w_{ij}(k) \left( x_j(k) - x_i(k) \right) + u_i(k) \right) \end{aligned} \tag{3.83}$$

(note that the above function contains only relative information of agent $i$ with respect to its neighbors). On the other hand, input saturation can be defined for (3.82) as:

$$x_i(k+1) = x_i(k) + sat\Big(\boldsymbol{\ell}_i(\boldsymbol{x}(k), u_i(k), k)\Big), \tag{3.84}$$

where $sat(.)$ is the saturation function described below:

$$sat(x) = \begin{cases} sign(x)\bar{u}_i, & |x| \geq \bar{u}_i \\ x, & |x| < \bar{u}_i \end{cases}. \tag{3.85}$$

Moreover, $sign(.)$ is the sign function and $\bar{u}_i > 0$ is the upper bound of the input saturation for agent $i$. Equation (3.84) can be rewritten in the following form:

$$x_i(k+1) = x_i(k) + \alpha_i\Big(\mathscr{f}_i(\boldsymbol{x}(k), u_i(k), k)\Big)\mathscr{f}_i(\boldsymbol{x}(k), u_i(k), k), \tag{3.86}$$

where $\alpha_i(.)$ is a scalar function given by:

$$\alpha_i(x) = \begin{cases} \dfrac{\bar{u}_i}{|x|}, & |x| \geq \bar{u}_i \\ 1, & |x| < \bar{u}_i \end{cases}, \tag{3.87}$$

and bounded as:

$$0 < \alpha_i(.) \leq 1. \tag{3.88}$$

Equation (3.86) can then be expanded as follows:

$$\begin{aligned} x_i(k+1) =& \frac{1}{1+d_i(k)} \\ & \Bigg( \Big(1 + d_i(k) - d_i(k)\alpha_i\Big(\boldsymbol{\ell}_i(\boldsymbol{x}(k), u_i(k), k)\Big)\Big) x_i(k) \\ & + \alpha_i\Big(\boldsymbol{\ell}_i(\boldsymbol{x}(k), u_i(k), k)\Big) \sum_{j \in \mathcal{N}_i(k)} w_{ij}(k) x_j(k) \\ & + \alpha_i\Big(\boldsymbol{\ell}_i(\boldsymbol{x}(k), u_i(k), k)\Big) u_i(k) \Bigg). \end{aligned} \tag{3.89}$$

The above equation can be written in matrix form for all agents as:

$$\boldsymbol{x}(k+1) = \mathscr{P}\Big(\boldsymbol{x}(k), \boldsymbol{u}(k), k\Big)\boldsymbol{x}(k) + \mathscr{B}\Big(\boldsymbol{x}(k), \boldsymbol{u}(k), k\Big)\boldsymbol{u}(k), \tag{3.90}$$

where $\mathscr{P}$ and $\mathscr{B}$ are matrix-valued functions defined by:

$$\begin{aligned} \mathscr{P}\Big(\boldsymbol{x}(k), \boldsymbol{u}(k), k\Big) &= (I + D(k))^{-1} \\ & \Bigg( I + D(k)\Big(I - \boldsymbol{\alpha}\Big(\boldsymbol{x}(k), \boldsymbol{u}(k), k\Big)\Big) + \boldsymbol{\alpha}\Big(\boldsymbol{x}(k), \boldsymbol{u}(k), k\Big) W(k) \Bigg) \end{aligned} \tag{3.91}$$

$$\mathscr{B}\Big(\boldsymbol{x}(k), \boldsymbol{u}(k), k\Big) = (I + D(k))^{-1} \boldsymbol{\alpha}\Big(\boldsymbol{x}(k), \boldsymbol{u}(k), k\Big), \tag{3.92}$$

Furthermore, $\boldsymbol{\alpha}\Big(\boldsymbol{x}(k), \boldsymbol{u}(k), k\Big)$ is a diagonal matrix function with $\alpha_i\Big(\boldsymbol{\ell}_i(\boldsymbol{x}(k), u_i(k), k)\Big)$ being its $i^{th}$ diagonal element, $i \in \mathbb{N}_n$.

### 3.7.1 Stability Analysis

For $\boldsymbol{u}(k) = 0$, $x_1\mathbf{1}_n$ is an equilibrium of (3.90) because:

$$\mathscr{P}\Big(x_1\mathbf{1}_n, 0, k\Big)x_1\mathbf{1}_n = x_1\mathbf{1}_n. \tag{3.93}$$

To investigate the stability of the above equilibrium point, define $\boldsymbol{x}(k) = x_1\mathbf{1}_n - \boldsymbol{e}(k)$ and substitute it into (3.86) to obtain:

$$e_i(k+1) = e_i(k) + \alpha_i\Big(\boldsymbol{\ell}_i(\boldsymbol{e}(k), -u_i(k), k)\Big)\boldsymbol{\ell}_i(\boldsymbol{e}(k), -u_i(k), k), \tag{3.94}$$

where $\boldsymbol{\ell}_i(\boldsymbol{e}(k), -u_i(k), k)$ is:

$$\begin{aligned}\boldsymbol{\ell}_i(\boldsymbol{e}(k), -u_i(k), k) = &\frac{1}{1+d_i(k)} \\ &\left(\sum_{j\in\mathcal{N}_i(k)} w_{ij}(k)\Big(e_j(k) - e_i(k)\Big) - u_i(k)\right).\end{aligned} \tag{3.95}$$

Then, the dynamic equation of the network can be written as:

$$\boldsymbol{e}(k+1) = \mathscr{P}\Big(\boldsymbol{e}(k), \boldsymbol{u}(k), k\Big)\boldsymbol{e}(k) - \mathscr{B}\Big(\boldsymbol{e}(k), \boldsymbol{u}(k), k\Big)\boldsymbol{u}(k), \tag{3.96}$$

where the functions $\mathscr{P}$ and $\mathscr{B}$ are defined in (3.91) and (3.92), with $\boldsymbol{\alpha}\Big(\boldsymbol{e}(k), -\boldsymbol{u}(k), k\Big) = \text{diag}(\Big[\alpha_1\Big(\boldsymbol{\ell}_1(\boldsymbol{e}(k), -u_1(k), k)\Big) \quad \dots \quad \alpha_n\Big(\boldsymbol{\ell}_n(\boldsymbol{e}(k), -u_n(k), k)\Big)\Big]$). Applying the control

input defined in (3.44), one arrives at:

$$\boldsymbol{e}(k+1) = \mathscr{P}^{cl}\Big(\boldsymbol{e}(k), k\Big)\boldsymbol{e}(k), \tag{3.97}$$

where:

$$\begin{aligned}\mathscr{P}^{cl}\Big(\boldsymbol{e}(k), k\Big) &= (I + D(k))^{-1} \\ &\left(\mathscr{H}\Big(\boldsymbol{e}(k), k\Big) + D(k)\Big(I - \boldsymbol{\alpha}\Big(\boldsymbol{e}(k), k\Big)\Big) + \boldsymbol{\alpha}\Big(\boldsymbol{e}(k), k\Big)W(k)\right),\end{aligned} \tag{3.98}$$

Also $\mathscr{H}\Big(\boldsymbol{e}(k), k\Big)$ is a diagonal matrix function, with $\hbar_i\Big(\boldsymbol{e}(k), k\Big)$ defined below being its $i$th diagonal element:

$$\hbar_i\Big(\boldsymbol{e}(k), k\Big) = \begin{cases} 1 - \alpha_i\Big(\ell_i(\boldsymbol{e}(k), -e_i(k), k)\Big), & i \in \mathcal{N}_1(k) \cup \mathcal{N}_j(k),\ j \in \mathcal{N}_1(k) \\ 1, & otherwise \end{cases}, \tag{3.99}$$

and $\boldsymbol{\alpha}\Big(\boldsymbol{e}(k), k\Big)$ is a diagonal matrix function with $\alpha_i\Big(\boldsymbol{e}(k), k\Big)$ defined below being its $i$th diagonal element:

$$\alpha_i\Big(\boldsymbol{e}(k), k\Big) = \begin{cases} \alpha_i\Big(\ell_i(\boldsymbol{e}(k), -e_i(k), k)\Big), & i \in \mathcal{N}_1(k) \cup \mathcal{N}_j(k),\ j \in \mathcal{N}_1(k) \\ \alpha_i\Big(\ell_i(\boldsymbol{e}(k), 0, k)\Big), & otherwise \end{cases}. \tag{3.100}$$

Since the first element of $\boldsymbol{e}(k)$, which corresponds to the leader, is always zero, the first row and column of $\mathscr{P}^{cl}$ can be removed. Thus, the new equation is written as:

$$\tilde{\boldsymbol{e}}(k+1) = \tilde{\mathscr{P}}^{cl}\Big(\tilde{\boldsymbol{e}}(k), k\Big)\tilde{\boldsymbol{e}}(k), \tag{3.101}$$

where $\tilde{\mathscr{P}}^{cl}$ is:

$$\begin{aligned}\tilde{\mathscr{P}}^{cl}\Big(\tilde{\boldsymbol{e}}(k), k\Big) = (\tilde{I} + \tilde{D}(k))^{-1} \\ \left(\tilde{\mathscr{H}}\Big(\tilde{\boldsymbol{e}}(k), k\Big) + \tilde{D}(k)\Big(\tilde{I} - \tilde{\boldsymbol{\alpha}}\Big(\tilde{\boldsymbol{e}}(k), k\Big)\Big) + \tilde{\boldsymbol{\alpha}}\Big(\tilde{\boldsymbol{e}}(k), k\Big)\tilde{W}(k)\right).\end{aligned} \tag{3.102}$$

Moreover, $\tilde{D}(k)$ :=diag($\begin{bmatrix} d_2(k) & \dots & d_n(k)\end{bmatrix}$), $\tilde{\boldsymbol{\alpha}}\Big(\tilde{\boldsymbol{e}}(k), k\Big)$ :=diag($\begin{bmatrix}\alpha_2\Big(\tilde{\boldsymbol{e}}(k), k\Big) \dots \alpha_n\Big(\tilde{\boldsymbol{e}}(k), k\Big)\end{bmatrix}$), $\tilde{\mathscr{H}}\Big(\tilde{\boldsymbol{e}}(k), k\Big)$ :=diag($\begin{bmatrix}\hbar_2\Big(\tilde{\boldsymbol{e}}(k), k\Big) & \dots & \hbar_n\Big(\tilde{\boldsymbol{e}}(k), k\Big)\end{bmatrix}$), $\tilde{W}(k)$ is obtained by removing the first row and column of $W(k)$, and $\tilde{I}$ is the identity matrix of appropriate dimension.

**Theorem 3.5.** *System* (3.101) *is globally asymptotically stable.*

*Proof.* As the first step of the proof, note that the saturation function $\alpha_i(\tilde{\boldsymbol{e}}(k), k)$ is bounded according to (3.88). Instead of the scalar function $\alpha_i(\tilde{e}(k), k)$, consider an unknown time-varying scalar $\hat{a}_i(k)$ with the same upper and lower bounds as (3.88) [75]. Thus, the function matrix $\tilde{\mathscr{P}}^{cl}\Big(\tilde{\boldsymbol{e}}(k), k\Big)$ in (3.102) is converted into a time-varying matrix

$\tilde{P}^{cl}(k)$ as follows:

$$\begin{aligned}\tilde{P}^{cl}(k) = &\Big(\tilde{I} + \tilde{D}(k)\Big)^{-1} \\ &\Big(\hat{H}(k) + \tilde{D}(k)\Big(\tilde{I} - \hat{A}(k)\Big) + \hat{A}(k)\tilde{W}(k)\Big),\end{aligned} \tag{3.103}$$

where $\hat{A}(k)$ is a diagonal matrix, with $\hat{a}_i(k)$ being its $i$th diagonal element, and $\hat{H}(k)$ is a diagonal matrix, with $\hat{h}_i(k)$ defined below being its $i$th diagonal element:

$$\hat{h}_i(k) = \begin{cases} 1 - \hat{a}_i(k), & i \in \mathcal{N}_1(k) \cup \mathcal{N}_j(k),\ j \in \mathcal{N}_1(k) \\ 1, & otherwise \end{cases}. \tag{3.104}$$

Assume that $i \in \mathcal{N}_1(k)$. Then, the $i^{th}$ row sum of $\tilde{P}^{cl}(k)$ is:

$$\begin{aligned}\sum_{j=1}^{n-1} \tilde{p}_{ij}^{cl}(k) =&\Big(1 + \tilde{d}_i(k)\Big)^{-1}\Big(1 - \hat{a}_i(k) + \tilde{d}_i(k) - \tilde{d}_i(k)\hat{a}_i(k) + \tilde{d}_i(k)\hat{\alpha}_i(k) - w_{i1}\hat{a}_i(k)\Big) \\ =&\Big(1 + \tilde{d}_i(k)\Big)^{-1}\Big(1 - \hat{a}_i(k) + \tilde{d}_i(k) - w_{i1}\hat{a}_i(k)\Big).\end{aligned} \tag{3.105}$$

Note that $\sum_{j=1}^{n-1} \tilde{w}_{ij}(k) = \tilde{d}_i(k) - w_{i1}(k)$ given that the first row of $W$ is removed to obtain $\tilde{W}$. Furthermore, $0 \leq 1 - \hat{a}_i(k) < 1$ and $\tilde{d}_i(k) > w_{i1}\hat{a}_i(k)$. Thus:

$$\Big(1 + \tilde{d}_i(k)\Big) > \Big(1 - \hat{a}_i(k) + \tilde{d}_i(k) - w_{i1}\hat{a}_i(k)\Big), \tag{3.106}$$

and

$$\sum_{j=1}^{n-1} \tilde{p}_{ij}^{cl}(k) < 1. \tag{3.107}$$

Now, assume that $i \in \mathcal{N}_j(k)$, where $j \in \mathcal{N}_1(k)$. Then, the $i^{th}$ row sum of $\tilde{P}^{cl}(k)$ is:

$$\begin{aligned}\sum_{j=1}^{n-1} \tilde{p}_{ij}^{cl}(k) =& \Big(1 + \tilde{d}_i(k)\Big)^{-1} \Big(1 - \hat{a}_i(k) + \tilde{d}_i(k) - \tilde{d}_i(k)\hat{a}_i(k) + \tilde{d}_i(k)\hat{a}_i(k)\Big) \\ <& 1.\end{aligned} \tag{3.108}$$

Finally, assume that $i \notin \mathcal{N}_1(k) \cup \mathcal{N}_j(k)$, where $j \in \mathcal{N}_1(k)$. The $i^{th}$ row sum of $\tilde{P}^{cl}(k)$ is:

$$\begin{aligned}\sum_{j=1}^{n-1} \tilde{p}_{ij}^{cl}(k) =& \Big(1 + \tilde{d}_i(k)\Big)^{-1} \Big(1 + \tilde{d}_i(k) - \tilde{d}_i(k)\hat{a}_i(k) + \tilde{d}_i(k)\hat{a}_i(k)\Big) \\ =& 1.\end{aligned} \tag{3.109}$$

The above row sum is the same as that of $\tilde{P}^s(k)$ in (3.63). Since $\tilde{I} - \hat{A}(k)$ is a diagonal non-negative matrix, $\tilde{P}^{cl}(k)$ is a non-negative matrix as well. Hence, $\tilde{P}^{cl}(k)$ and $\tilde{P}^s(k)$ have similar properties and the same network connectivity ($\tilde{W}(k)$ is the same in both (3.59) and (3.103)). Thus, following a procedure similar to the proof of Theorem 3.3, it is concluded that (3.103) is asymptotically stable. Hence, it is guaranteed that the equilibrium $\boldsymbol{e} = 0$ for the nonlinear system is locally asymptotically stable. To study the global stability, it is required to find the region of attraction $R$, defined as [76]:

$$R = \{\boldsymbol{e}(k_0) \in \mathbb{R}^{n-1} \mid \Phi(k_l, k_0) \to 0 \text{ as } l \to \infty\} \tag{3.110}$$

To obtain an estimate of $R$, the boundary of $\boldsymbol{e}$ in the nonlinear system should be derived

first [75]. From (3.88) and (3.94):

$$\mathscr{f}_i(\boldsymbol{e}(k), -u_i(k), k) \leq \frac{\bar{u}_i}{\alpha_{min}}, \tag{3.111}$$

where $\alpha_{min}$ is the minimum value of $\alpha_i$ which is zero as given in (3.88). This means that the size of the region $R$ is infinite, and hence the region of attraction of the equilibrium of the nonlinear function at the origin is the entire space. Thus, (3.101) is globally asymptotically stable. ■

**Remark 3.11.** One can use an approach similar to the proof of Theorem 3.5 to show that under the control command described in Algorithm 1 the network subject to input saturation described by (3.90) is globally asymptotically stable.

**Example 3.6.** Consider a network of 100 mobile robots with a topology which is updated at every 20 steps and a weight matrix which is updated at every step. Let all agents be subjected to input saturation with the upper bound $\bar{u}_i = 2.8$, $\forall i \in \mathbb{N}_n$. The generalized algebraic connectivity of the system is depicted in Figure 3.22. Figures 3.23-3.26 show the states of the network for the system subject to input saturation without any control command, with leader's control command, with leader's and subleaders' control commands, and with the commands of all agents that act as a leader. These figures demonstrate that under all four control strategies all agents converge to the leader. Figure 3.27 compares the average of the states of the agents under all four control strategies. It is clear that using all agents as the leader and follower can lead the system to a faster

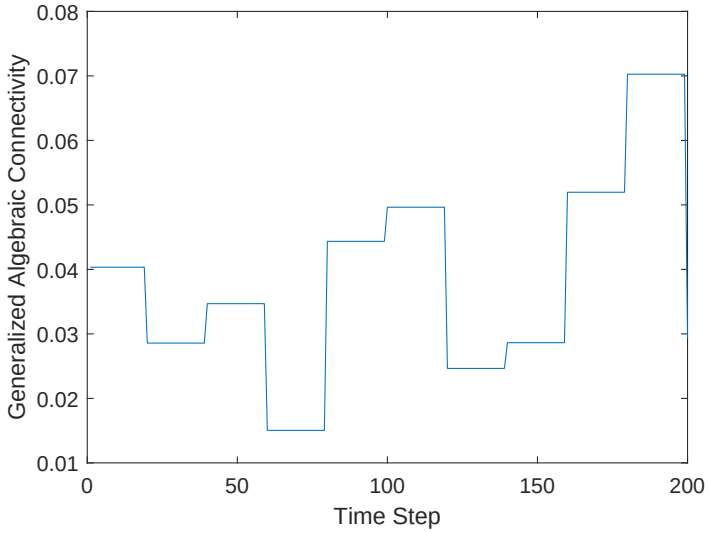


Figure 3.22: Generalized algebraic connectivity of the network in Example 3.6.

convergence even in presence of the input saturation.

## 3.8 Conclusions

Consensus control of a multi-agent network with leader-follower structure is investigated in this chapter. The control strategy uses local information of the agents, based on the nearest neighbor rule. The network can be as simple as one with a small number of agents, time-invariant weights and fixed topology, and can be as complex as one with a large number of agents, time-varying weights and switching topology. Accordingly, the control scheme can be as simple as some commands generated by the leader for all agents, commands generated by the leader and its neighbors, and commands generated

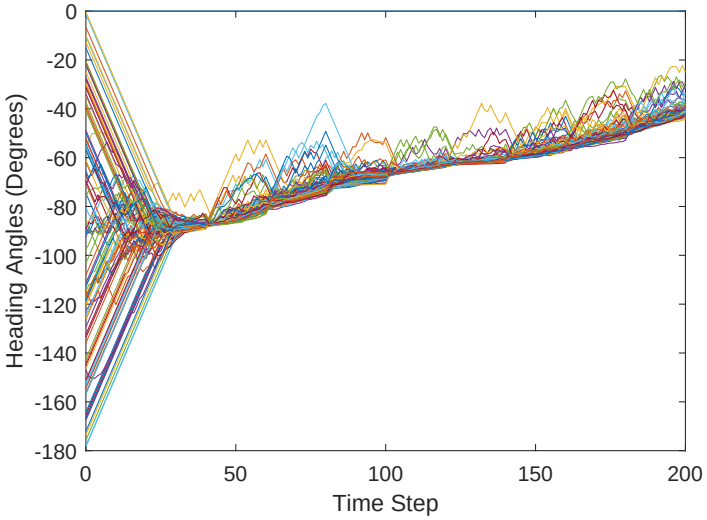


Figure 3.23: States of the agents in the network in the presence of input saturation in Example 3.6 with the standard protocol.

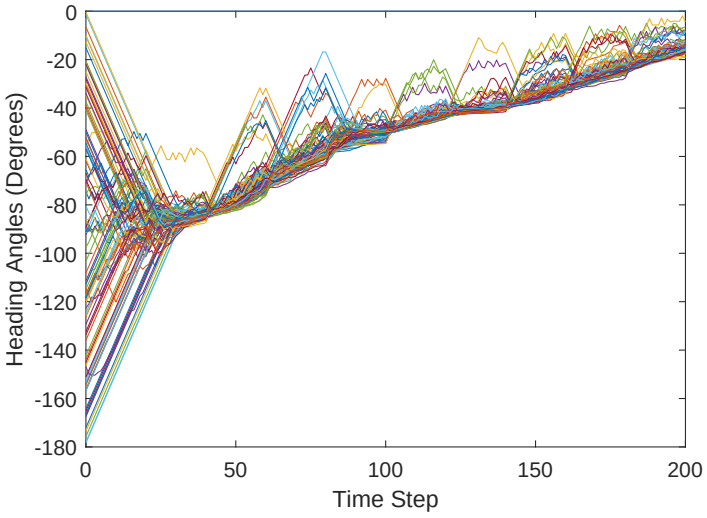


Figure 3.24: States of the agents in the network in the presence of input saturation in Example 3.6 with leader's control command.

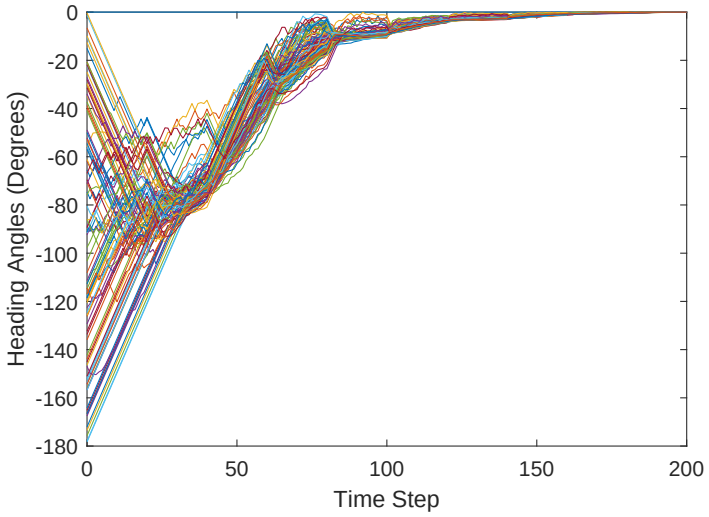


Figure 3.25: States of the agents in the network in the presence of input saturation in Example 3.6 with leader's and subleaders' control commands.

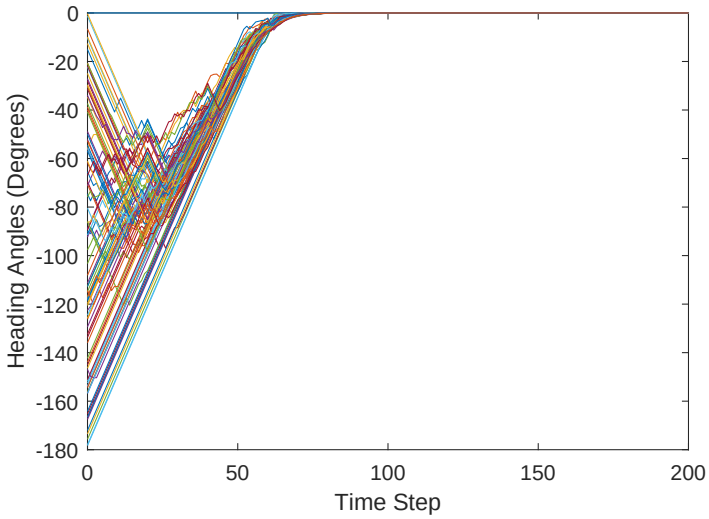


Figure 3.26: States of the agents in the network in the presence of input saturation in Example 3.6 with the commands of all agents that act as a leader.

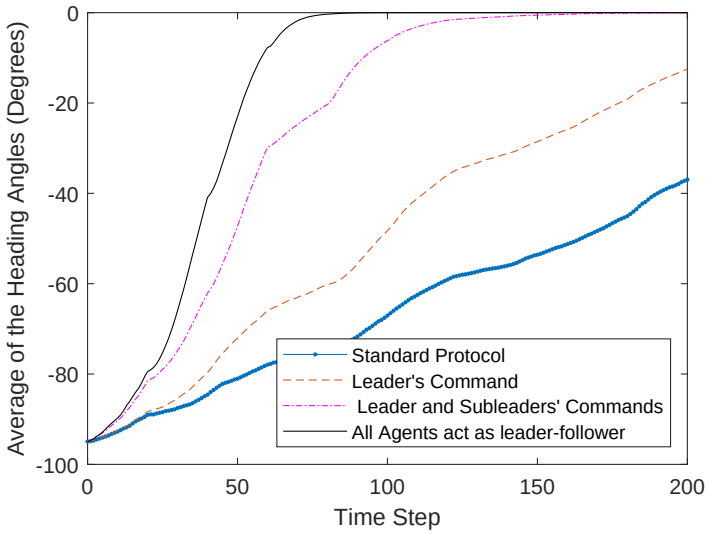


Figure 3.27: Comparison between all four control strategies for the network of Example 3.6 with switching topology, time-varying weights, and input saturation.

by all agents for their neighbors. To evaluate the convergence performance of the proposed control schemes, the location of the dominant eigenvalue of the closed-loop system for the case of time-invariant weights with fixed topology, and the composition of the state transition matrix for the case of time-varying weights with switching topology are investigated. The stability of the closed-loop system in the presence of input saturation is also analyzed. The efficacy of the results obtained is confirmed by several numerical examples.

# Chapter 4

# Conclusion and Future Work

This thesis investigates consensus control of a multi-agent network with leader-follower structure. The main contribution of this thesis is introducing a new consensus control scheme by adding the control command to the dynamic equation of the network under nearest neighbor rule. Networks with fixed and switching topologies are studied and the input saturation in agents' dynamics is also considered. The stability of the network under the proposed methods is investigated.

In Chapter 2, the equilibrium characteristics in multi-agent networks are studied. Using a Markov chain model, a simple technique for computing the steady-state matrix is provided. It is then shown how the number of links can impact the convergence time. A control rule is subsequently proposed and the eigenvalues of the resultant closed-loop system for two network topologies are investigated for convergence analysis.

The results show that the proposed method has a faster convergence rate compared to the conventional consensus control rules. Simulation results confirm the superior performance of the proposed follower-based control allocation strategy.

In Chapter 3, consensus control of a multi-agent network with leader-follower structure is investigated. The control strategy uses local information of the agents, based on the nearest neighbor rule. The network can be as simple as one with a small number of agents, time-invariant weights and fixed topology, and can be as complex as one with a large number of agents, time-varying weights and switching topology. Accordingly, the control scheme can be as simple as some commands generated by the leader for all agents, commands generated by the leader and its neighbors, and commands generated by all agents for their neighbors. To evaluate the convergence performance of the proposed control schemes, the location of the dominant eigenvalues of the closed-loop system for the case of time-invariant weights with fixed topology, and the composition of the state transition matrix for the case of time-varying weights with switching topology are investigated. The stability of the closed-loop system in the presence of input saturation is also analyzed. The efficacy of the results obtained is confirmed by several numerical examples.

## 4.1 Suggestions for Future Work

The main contribution of this thesis is to develop distributed control strategies for a multi-agent network with leader-follower structure based on the nearest neighbor rule. The following ideas are suggested for future research directions.

- In the present work it is assumed that after removing the leader and its links, the graph of the network will be divided into multiple subgraphs. Each subgraph is also assumed to be a weighted undirected graph. It would be interesting to investigate an asymmetric network with weighted links. Note that the convergence analysis in this case will be more complicated given that the graph representing an asymmetric network is directed.

- Throughout this work, the network is assumed to be homogeneous. It would be interesting to extend the results to the case of heterogeneous networks.

- The control command for each agent in this work is computed based on the relative information of the agent with respect to its neighbors. To achieve faster convergence, each agent can compute its control command based on both its local information and estimates of the states of all other agents.

- This work can be extended to the case when communications between agents is subject to perturbation, noise or data loss. Therefore, it would be interesting to develop a robust, reliable and fault-tolerant control strategy for this type of

multi-agent network.

# Bibliography

[1] J. G. Skellam, "Random dispersal in theoretical populations," *Biometrika*, vol. 38, no. 1/2, pp. 196–218, 1951.

[2] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[3] A. Ligtenberg, M. Wachowicz, A. K. Bregt, A. Beulens, and D. L. Kettenis, "A design and application of a multi-agent system for simulation of multi-actor spatial planning," *Journal of Environmental Management*, vol. 72, no. 1-2, pp. 43–55, 2004.

[4] S. D. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-agent systems for power engineering applications-part i: Concepts, approaches, and technical challenges," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1743–1752, 2007.

[5] I. Arel, C. Liu, T. Urbanik, and A. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*,

vol. 4, no. 2, pp. 128–135, 2010.

[6] H. Mahboubi, F. Sharifi, A. G. Aghdam, and Y. Zhang, "Distributed control of multi-agent systems with limited communication range in the fixed obstacle environments," *IEEE Access*, vol. 7, pp. 118 259–118 268, 2019.

[7] D. V. Dimarogonas and K. J. Kyriakopoulos, "On the rendezvous problem for multiple nonholonomic agents," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 916–922, 2007.

[8] 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.

[9] Y. Cao, W. Ren, and Z. Meng, "Decentralized finite-time sliding mode estimators and their applications in decentralized finite-time formation tracking," *Systems & Control Letters*, vol. 59, no. 9, pp. 522–529, 2010.

[10] 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.

[11] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[12] E. Semsar-Kazerooni and K. Khorasani, "An LMI approach to optimal consensus seeking in multi-agent systems," in *2009 American Control Conference*, 2009, pp. 4519–4524.

[13] M. F. Land and T. Collett, "Chasing behaviour of houseflies (fannia canicularis)," *Journal of Comparative Physiology*, vol. 89, no. 4, pp. 331–357, 1974.

[14] A. Okubo, "Dynamical aspects of animal grouping: swarms, schools, flocks, and herds," *Advances in Biophysics*, vol. 22, pp. 1–94, 1986.

[15] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[16] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, no. 6, p. 1226, 1995.

[17] 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.

[18] W. Hu, L. Liu, and G. Feng, "Consensus of linear multi-agent systems by distributed event-triggered strategy," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 148–157, 2015.

[19] Z. Li, W. Ren, X. Liu, and M. Fu, "Consensus of multi-agent systems with general linear and lipschitz nonlinear dynamics using distributed adaptive protocols," *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1786–1791, 2012.

[20] W. Ni and D. Cheng, "Leader-following consensus of multi-agent systems under fixed and switching topologies," *Systems & Control Letters*, vol. 59, no. 3-4, pp. 209–217, 2010.

[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] B. Ning, Q. Han, Z. Zuo, J. Jin, and J. Zheng, "Collective behaviors of mobile robots beyond the nearest neighbor rules with switching topology." *IEEE Transactions on Cybernetics*, vol. 48, no. 5, p. 1577, 2018.

[23] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[24] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[25] 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.

[26] R. M. Murray, "Recent research in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 571–583, 2007.

[27] J. Yu and L. Wang, "Group consensus of multi-agent systems with undirected communication graphs," in *7th Asian Control Conference*, 2009, pp. 105–110.

[28] M. Porfiri and D. J. Stilwell, "Consensus seeking over random weighted directed graphs," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1767–1773, 2007.

[29] D. Yang, W. Ren, X. Liu, and W. Chen, "Decentralized event-triggered consensus for linear multi-agent systems under general directed graphs," *Automatica*, vol. 69, pp. 242–249, 2016.

[30] K. Cai and H. Ishii, "Average consensus on arbitrary strongly connected digraphs with time-varying topologies," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 1066–1071, 2014.

[31] A. Priolo, A. Gasparri, E. Montijano, and C. Sagues, "A distributed algorithm for average consensus on strongly connected weighted digraphs," *Automatica*, vol. 50, no. 3, pp. 946–951, 2014.

[32] Y. Shang, "Continuous-time average consensus under dynamically changing topologies and multiple time-varying delays," *Applied Mathematics and Computation*, vol. 244, pp. 457–466, 2014.

[33] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90–104, 2011.

[34] D. Li, Q. Liu, X. Wang, and Z. Lin, "Consensus seeking over directed networks with limited information communication," *Automatica*, vol. 49, no. 2, pp. 610–618, 2013.

[35] T. Li, M. Fu, L. Xie, and J. F. Zhang, "Distributed consensus with limited communication data rate," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 279–292, 2010.

[36] W. Fu, J. Qin, J. Wu, W. X. Zheng, and Y. Kang, "Interval consensus over random networks," *Automatica*, vol. 111, p. 108603, 2020.

[37] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

[38] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 19–33, 2007.

[39] T. Wimalajeewa and S. K. Jayaweera, "Optimal power scheduling for correlated data fusion in wireless sensor networks via constrained pso," *IEEE Transactions on Wireless Communications*, vol. 7, no. 9, pp. 3608–3618, 2008.

[40] L. Xiao, S. Boyd, and S. J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.

[41] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[42] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[43] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed convex optimization on dynamic networks," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3545–3550, 2016.

[44] Y. Hong, J. Hu, and L. Gao, "Tracking control for multi-agent consensus with an active leader and variable topology," *Automatica*, vol. 42, no. 7, pp. 1177–1182, 2006.

[45] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.

[46] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, "Minimizing convergence error in multi-agent systems via leader selection: A supermodular optimization approach," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1480–1494, 2014.

[47] Z. Kan, J. M. Shea, and W. E. Dixon, "Leader–follower containment control over directed random graphs," *Automatica*, vol. 66, pp. 56–62, 2016.

[48] J. Shao, J. Qin, A. N. Bishop, T. Z. Huang, and W. X. Zheng, "A novel analysis on the efficiency of hierarchy among leader-following systems," *Automatica*, vol. 73, pp. 215–222, 2016.

[49] L. Moreau, "Stability of continuous-time distributed consensus algorithms," in *43rd IEEE Conference on Decision and Control*, vol. 4, 2004, pp. 3998–4003.

[50] T. Li and J. F. Zhang, "Consensus conditions of multi-agent systems with time-varying topologies and stochastic communication noises," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2043–2057, 2010.

[51] F. Xiao and L. Wang, "State consensus for multi-agent systems with switching topologies and time-varying delays," *International Journal of Control*, vol. 79, no. 10, pp. 1277–1284, 2006.

[52] Y. G. Sun and L. Wang, "Consensus of multi-agent systems in directed networks with nonuniform time-varying delays," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1607–1613, 2009.

[53] D. Meng and Y. Jia, "Scaled consensus problems on switching networks," *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1664–1669, 2015.

[54] Z. Li, Z. Duan, and F. L. Lewis, "Distributed robust consensus control of multi-agent systems with heterogeneous matching uncertainties," *Automatica*, vol. 50, no. 3, pp. 883–889, 2014.

[55] Y. Zheng and L. Wang, "Distributed consensus of heterogeneous multi-agent systems with fixed and switching topologies," *International Journal of Control*, vol. 85, no. 12, pp. 1967–1976, 2012.

[56] L. Wang, W. J. Feng, M. Z. Chen, and Q. G. Wang, "Consensus of nonlinear multi-agent systems with adaptive protocols," *IET Control Theory & Applications*, vol. 8, no. 18, pp. 2245–2252, 2014.

[57] W. Yu, G. Chen, and M. Cao, "Consensus in directed networks of agents with nonlinear dynamics," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1436–1441, 2011.

[58] D. Buzorgnia and A. G. Aghdam, "A follower-based control allocation in multi-agent networks," in *2018 American Control Conference*, 2018, pp. 43–48.

[59] L. Shi, Y. Xiao, J. Shao, and W. X. Zheng, "Containment control of asynchronous discrete-time general linear multiagent systems with arbitrary network topology," *IEEE Transactions on Cybernetics*, 2019.

[60] T. Yang, Z. Meng, D. V. Dimarogonas, and K. H. Johansson, "Global consensus for discrete-time multi-agent systems with input saturation constraints," *Automatica*, vol. 50, no. 2, pp. 499–506, 2014.

[61] H. Su, M. Z. Chen, X. Wang, and J. Lam, "Semiglobal observer-based leader-following consensus with input saturation," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2842–2850, 2013.

[62] Y. Li, J. Xiang, and W. Wei, "Consensus problems for linear time-invariant multi-agent systems with saturation constraints," *IET Control Theory & Applications*, vol. 5, no. 6, pp. 823–829, 2011.

[63] H. Geng, Z. Q. Chen, Z. X. Liu, and Q. Zhang, "Consensus of a heterogeneous multi-agent system with input saturation," *Neurocomputing*, vol. 166, pp. 382–388, 2015.

[64] D. Gamerman and H. F. Lopes, *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*, 2nd ed., ser. Chapman & Hall/CRC Texts in Statistical Science. CRC Press, 2006.

[65] W. Ren and R. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 73–82, 2004.

[66] D. A. Levin and Y. Peres, *Markov Chains and Mixing Times*. American Mathematical Soc., 2017, vol. 107.

[67] B. N. Datta, *Numerical Linear Algebra and Applications*. SIAM, 2010, vol. 116.

[68] D. S. Watkins, *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*. SIAM, 2007, vol. 101.

[69] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.

[70] P. Diaconis, D. Stroock *et al.*, "Geometric bounds for eigenvalues of markov chains," *The Annals of Applied Probability*, vol. 1, no. 1, pp. 36–61, 1991.

[71] M. Gopal, *Modern Control System Theory*. New Age International, 1993.

[72] B. Zhou and T. Zhao, "On asymptotic stability of discrete-time linear time-varying systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4274–4281, 2017.

[73] M. M. Asadi, M. Khosravi, A. G. Aghdam, and S. Blouin, "Generalized algebraic connectivity for asymmetric networks," in *2016 American Control Conference*, 2016, pp. 5531–5536.

[74] L. Xiao, S. Boyd, and S. Lall, "Distributed average consensus with time-varying metropolis weights," *Automatica*, 2006.

[75] F. Amato, *Robust Control of Linear Systems Subject to Uncertain Time-Varying Parameters*. Springer, 2006, vol. 325.

[76] H. K. Khalil, "Nonlinear systems," 1992.