8-2019

COOPERATIVE LEARNING FOR THE CONSENSUS OF MULTI-AGENT SYSTEMS

Qishuai Liu
University of Nebraska - Lincoln, qishuai.liu@huskers.unl.edu

COOPERATIVE LEARNING FOR THE CONSENSUS OF MULTI-AGENT SYSTEMS

by

Qishuai Liu

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Doctor of Philosophy

Major: Electrical Engineering

Under the Supervision of Professor Qing Hui

Lincoln, Nebraska

August, 2019

COOPERATIVE LEARNING FOR THE CONSENSUS OF MULTI-AGENT SYSTEMS

Qishuai Liu, Ph.D.

University of Nebraska, 2019

Adviser: Qing Hui

Due to a lot of attention for the multi-agent system in recent years, the consensus algorithm gained immense popularity for building fault-tolerant systems in system and control theory. Generally, the consensus algorithm drives the swarm of agents to work as a coherent group that can reach an agreement regarding a certain quantity of interest, which depends on the state of all agents themselves. The most common consensus algorithm is the average consensus, the final consensus value of which is equal to the average of the initial values. If we want the agents to find the best area of the particular resources, the average consensus will be failure. Thus the algorithm is restricted due to its incapacity to solve some optimization problems.

In this dissertation, we want the agents to become more intelligent so that they can handle different optimization problems. Based on this idea, we first design a new consensus algorithm which modifies the general bat algorithm. Since bat algorithm is a swarm intelligence method and is proven to be suitable for solving the optimization problems, this modification is pretty straightforward. The optimization problem suggests the convergence direction. Also, in order to accelerate the convergence speed, we incorporate a term related to flux function, which serves as an energy/mass exchange rate in compartmental modeling or a heat transfer rate in thermodynamics. This term is inspired by the speed-up and speed-down strategy from biological swarms. We prove the stability of the proposed consensus algorithm for both linear and nonlinear flux functions in detail by the matrix paracontraction tool and the Lyapunov-based method, respectively.

Another direction we are trying is to use the deep reinforcement learning to train the agent to reach the consensus state. Let the agent learn the input command by this method, they can become more intelligent without human intervention. By this method, we totally ignore the complex mathematical model in designing the protocol for the general consensus problem. The deep deterministic policy gradient algorithm is used to plan the command of the agent in the continuous domain. The moving robots systems are considered to be used to verify the effectiveness of the algorithm.

Table of Contents

List of Figures

List of Tables

NOMENCLATURE

$\cap$	intersectioin
$\cup$	union
δ_{ij}	1 if $i = j$, 0 if $i \neq j$
$\in$	is an element of
$\lambda_i(A)$	ith eigenvalue of $A \in \mathbb{R}^{n \times m}$
$\lambda_{\min}(A), \lambda_{\max}(A)$	minimum, maximum eigenvalues of the matrix A
$\limsup_{n \to \infty} f(x)$	limit superior of $f(x)$
$\log$	logarithm with base e
$\|A\|_\infty$	$\max_{i,j} \lvert A_{i,j} \rvert$
$\lvert x \rvert_1$	L_1 norm
$\|x\|_2$	Euclidean norm of x
$\|\cdot\|$	vector or matrix norm
$\overline{\mathbb{Z}}_+$	set of nonnegative integers
$\mathbb{C}^n$	$\mathbb{C}^{n \times 1}$ complex column vectors

$\mathbb{E}$	expectation
$\mathbb{N}$	nonnegative integers
$\mathbb{P}$	probability taken over the coin-flips
$\mathbb{R}$	real number
$\mathbb{R}^n$	$\mathbb{R}^{n\times 1}$ real column vectors
$\mathbb{R}^{n\times m}$	$n \times m$ matrics
$\mathbb{Z}$	integers
$\mathcal{N}$	set of neighbors of the node
$\min, \max$	minimum, maximum
∇	gradient
$\notin$	is not an element of
$\otimes$	Kronecker product
$\prod$	product operator
$\rho(P)$	specturm radius of A
$\rightrightarrows$	uniform convergence
$\sigma_i(A)$	ith largest singular value of $A \in \mathbb{R}^n$
$\subset$	is a proper subset of
$\subseteq$	is a subset of
$\sum$	summation operator

$\text{col}_i(A)$	ith column of A
$\det A$	determinant of A
$\text{diag}(a_1, \ldots, a_n)$	a diagonal matrix whose ith diagonal entry is a_i
exp	exponential function
$\ker(A)$	kernel of the matrix A
$\ker(A)^\perp$	orthogonal complement of $\ker(A)$
$\text{rank} A$	rank of A
$\text{ran}(A)$	range space of A
$\text{row}_i(A)$	ith row of A
sgn	signum function
$\text{span}(A)$	span of the matrix A
$\text{tr} A$	trace of A
$\triangleq$	equals by definition
$\{\}$	set
$A \geq 0$	nonnegative definite matrix
A^*	the complex conjugate of A
A^{T}	transpose of A
$A_{i,j}$	(i, j) entry of A
$E_{(i,j)}$	elementary matrix with unity in the (i, j)

$f : x \longmapsto y$	f is a function with domain x and codomain y
I_n, I	$n \times n$ identity matrix
Lap	Laplace distribution
$x \times y$	cross product of vectors $x, y \in \mathbb{R}^n$
x_i	ith component of $x \in \mathbb{R}^n$
mod	modulo operation

CHAPTER 1

INTRODUCTION

1.1 Motivation

The multi-agent systems (MAS) generally refers to a group of autonomous agents which can operate in a networked environment. A MAS is a system composed of multiple interacting intelligent agents which can communicate with each other, such that they can finish a general work together. MAS usually is used to solve some difficult problems which are not easy or sometimes impossible for a single agent or a monolithic system to solve. Intelligence generally includes some methodical, functional, procedural or algorithmic procedure to find the processing approaches which can be used to make an agent to finish its work more efficiently. In this area, some research methods related to the MAS are performed, an on-line trading method was studied in [1], disaster response was studied in [2], and modeling social structures was studied in [3].

The definition of the agent can be found in [4], where the concept of multi-agent systems is presented as well as its application. According to the authors, some of the characteristics of the agent can be defined as:

- Reactivity capabilitity: An agent has the ability to satisfy its own goal.
- Autonomy: An agent is at least partially autonomous.
- Perception: An agent can perceive its environment.

- Local view: An agent should not have the global view of the whole system, or the system is too complex such that the agent can only use partial understand of the whole knowledge.

- Communication capabilitity: An agent can be communicated with other agents in the same system.

A MAS often is composed of an environment, agents, and relationship between them. The general objective is to finish certain tasks through the agent performing operations to the environment. As a result, the environment is changed and the goal is finally achieved. MAS often is used to address the problem in automatic control, computer science, distributed computation, game theory, and social science. In these cases, a multi-agent system may include computer's agents, human teams and agent-human teams.

The MAS problems are studied widely in automatic control, specifically the system is consisted with multiple agents. These agents are supposed to equip with several sensors and actuators to perform a coordinated task. This is an important and challenging research area, which is motivated by a large number of applications. These applications include surveillance, collaborative search and rescue, environmental monitoring, and distributed reconfigurable sensor networks. In order to make these applications perform well, different cooperative control methods have been proposed and analyzed, such as formation control, rendezvous, attitude alignment, flocking, congestion control for different connected networks, air traffic control, coverage and cooperative search. In most cases, a multi-agent system can be seen as a group of nodes which denote different entities such as vehicles, sensors, and plants, etc. These entities can exchange their own information in a communication network in order to achieve the final goal. From this view, the MAS can be represented by a network of nodes which are connected through a communication topology. This communication topology can be represented by the graph theory.

Some of the classical objectives of the MAS include but are not limited to:

- Consensus and average consensus: this often refers to an agreement to which the value of the state can be reached by the multi-agent system. For the average consensus, the final state value of all the agents is the average of the initial state value if they follow the agreement protocol.

- Synchronization: If the state of all the agents reach the same point asymptotically, then the MAS reached synchronization. This is similar to the consensus algorithm, however, synchronization is generally applied in the mainfolds with particular symmetries.

- Formation control: The agents in the network are designed according to a particular configuration, where the objectives of all the agents are to achieve to a common goal. There are several formation control strategies in order to make the agents converge to the designed configuration or maintain the inter-agent distance.

- Exploration and coverage: By exploring the environment, all the agents can collect the information around the area of interest. Especially, the coverage is a collaborative task for all the agents to finish. They may reach the optimal state status in order to maximize their own interested area.

For all of these topics, consensus algorithms receive wide attention. Among those, average consensus is a popular distributed algorithm which can be used to compute the arithmetic mean of the state value of all the agents $\{x_i\}_{i=1}^N$, where the state value for agent i is x_i for all of the agents N. The average consensus is used to compute the average state value of all the agents, $\frac{1}{N}\sum_{i=1}^N x_i$, in the distributed way.

The consensus problems of multi-agent systems have gained a lot of attention from various scientific communities in recent years. With the development of the artificial intelligence in recent years, it is desired that each agent can handle different situations for the general algorithm. That means the agents can have a better cooperation and collaboration among them, thus the whole group can be driven to find a better state. In this process, information sharing, ideas generating, and decision making are better communicated between these agents and the whole group will reach a better decision when compared with previous methods. Thus the multi-agent learning algorithms are studied to demonstrate the need for a comprehensive understanding of the dynamic environment. The learning ability in mulit-agent system is not only related with the field of artificial intelligence but also studied extensively in game theory and economics as well.

1.2 Overview

This dissertation focuses on three aspects of designing consensus algorithm. First is to design the consensus algorithm based on the bat inspired swarm algorithm, which makes the proposed protocol more intelligence, meanwhile the flux function is incorporated in the protocol to accelerate the convergence rate. The second is to explore the privacy preserving for the proposed protocol. The last is to use deep reinforcement learning based method to reach the consensus.

In Chapter 2, a comprehensive review of the consensus and its application are provided.

In Chapter 3, a bat-inspired consensus algorithm is proposed based on the bat optimization. The bat optimization is proven to be suitable for solving the optimization problem and thus the proposed consensus algorithm can get this ability while reaching the consensus. In order to accelerate the convergence rate, a linear vector-valued flux function is considered to be incorporated into the algorithm. The proposed protocol uses a double-check technique to ensure that the minimum value can be found in each iteration. After that, the stability of the algorithm is proven by using the matrix paracontraction technique, which is a nonexpansive property and generally used for studying the convergence of linear iterations.

The stability of the proposed protocol can be regarded as the linear matrix problem such that the paracontraction can be used to solve it.

In Chapter 4, the nonlinear flux function is incorporated into the proposed consensus algorithm. In this chapter, since the flux function is nonlinear, we cannot transform the protocol in the matrix form. Under this condition, the Lyapunov-based method is used to analyze the stability of the proposed consensus algorithm. We also consider the performance of the consensus algorithm with some small disturbances. In this case, the stability of the protocol is studied. In order to show the effectiveness of the proposed protocol, two different optimization problems are presented and solved by this protocol. Meanwhile, all the agents can reach the consensus state at the same time. The final consensus state is the solution of the optimization problem.

In Chapter 5, the privacy characteristics of the proposed protocol is investigated. For this case, in each iteration, each agent may not want to share its own full state to others outside of the network. Instead, they can transmit its own state information encrypted. In this chapter, we consider to add noise to the state information transmitted for each agent. The ϵ-*differential privacy* of the proposed consensus algorithm is defined. With the help of d-accurate, we prove that the proposed consensus algorithm is ϵ-differential privacy. Finally, the convergence of the consensus algorithm is analyzed. The simulation result is presented to show that the proposed differential privacy consensus algorithm is effective.

In Chapter 6, we focus on another way to make the agent more intelligent. The deep reinforcement learning method is used to train the agent to reach the consensus state. By using this method, the agent can become more intelligent. They can learn different strategies to handle different problems. The deep deterministic policy gradient (DDPG) algorithm is used to train the agent since it can handle the continuous domain case for the input command of the agent. We use the mobile robots to test the proposed algorithm.

Finally, the conclusion is given in Chapter 7. The contribution of this research is also shown. Moreover, the comments for future research work are discussed.

CHAPTER 2

LITERATION REVIEW

This chapter provides a comprehensive literature review of the state-of-the-art consensus protocol. The applications of the consensus protocol are also reviewed. After that, the privacy protection methods for the network are then reviewed. Finally, the deep reinforcement learning methods for consensus are summarized.

2.1 Consensus Protocol

One of the origins for cooperative learning in multiagent systems is from biomimicry of animal swarm behaviors, such as bird flocking and fish schooling [5, 6]. There is a long history of people being fascinated by these stunning behaviors demonstrated by many creatures when they aggregate together to achieve a common goal. Biologists [7, 8], physicists [6, 9], and mathematicians [10] have constructed mathematical models to simulate these bio-inspired collective behaviors. These models always consist of constant local interaction and information exchange among individual subsystems to form a collective system. Although simulation of the collective system reveals quite a few intrinsic, surprising phenomena that are not exhibited by individual subsystems, no clear interpretation or little rigorous analysis of these phenomena was given at the inception of the models. See a recent survey related to this topic in [11]. Later, many control researchers took on this issue and developed several rigorous control-theoretic frameworks, using various mathematical tools, such as nonnegative matrix analysis [12], algebraic graph theory [13], convex analysis [14], and Lyapunov-based stability tools [15, 16], to explain why these interconnected systems can exhibit such bizarre group behaviors. All of these results ended with the same problem named after the consensus or agreement problem. It turns out that this problem has appeared in different contexts or fields, for example, stochastic algorithms [17], random processes [18], game theory [19], load balancing [20], etc.

The summary of recent progress about the consensus of multiagent systems can be found in [21, 22]. Moreover, the authors in [23] develop a mean field game to study the consensus behavior of agents, where the initial states of the agents are not necessarily Gaussain distribution. Finite-time consensus for agents having the integrator-like continuous dynamic model is proposed in [16,24,25], where semistability theory is introduced in [16,24] to guarantee its convergence while the communication network in [25] exists directional link failure. In [26], a Nussbaum-type function is used to design the control law such that the agents can seek the unknown control direction, therefore these agents can achieve the consensus cooperatively. The consensus behavior of agents, whose dynamics are modeled by diffusion partial differential equations, is studied in [27], where the agents dynamics are corrupted by additive persistent disturbances. In this case, a sliding mode based consensus is proposed. The author in [28] studies the consensus protocol among agents with antagonistic interactions, where the necessary and sufficient conditions are proposed to guarantee the consensus. This result is extended by authors in [29], where they study the opinion dynamics in social groups with ubiquitous competition and distrust between some pairs of agents.

Parallel with this advance in control research communities, researchers in computational intelligence also used similar biomimetic inspiration to develop highly successful swarm-intelligence-based optimization approaches during the last two decades. Among them, the most celebrated approaches involve ant colony optimization [30], particle swarm optimization [31], and differential evolution [32]. Although many of them are heuristic, they appear to be very successful in solving complicated optimization problems in critical infrastructures, such as power distribution management [33]. Due to the simple update formula of those algorithms, swarm-intelligence-based optimization algorithms can handle complex optimization algorithms with higher efficiency [33], which is hard to achieve by use of conventional optimization methods. Moreover, since usually no derivative or gradient operation is involved in the swarm-intelligence-based algorithms, they are able to solve more general optimization problems, such as mixed-integer, discontinuous optimization problems [34].

Of these swarm-intelligence-based optimization algorithms, the *bat searching* (BS) algorithm [35], based on the echolocation behavior of bats, gains our attention due to its striking analogue to the consensus problem for multiagent systems in control systems engineering. More specifically, BS uses a frequency-tuning technique to increase the diversity of the solutions in the population when solving an unconstrained optimization problem. Automatic zooming is used to balance exploration and exploitation during the search process for an optimal solution by mimicking the variations of pulse emission rates and loudness of bats. The BS algorithm has shown significant improvement compared to other swarm intelligence algorithms, such as particle swarm optimization, when solving unconstrained optimization problems whose objective functions are given by some standard test functions [35].

2.2 Application of the Consensus Protocol

Many apparently different problems that involve inter information exchange of the dynamical systems in various areas are closely related to consensus problems for MAS. Some of these problems are shown below.

- Firstly, the synchronization of coupled oscillators has attracted numerous engineers and scientists from diverse fields [36–38]. The general coupled neural oscillators system is the synchronous oscillations. The generalized Kuramoto model of coupled oscillators can be modeled as follows:

$$\dot{\theta}_i = \kappa \sum_{j \in N_i} \sin(\theta_j - \theta_i) + \omega_i$$

where θ_i and ω_i are the phase and frequency of the ith oscillator. In [39], the authors show that if the parameter κ is sufficient large, then synchronization to the aligned state of the network with all-to-all links is globally achieved for all initial states. Other references such as [40] study the synchronization of networked oscillators under variable time-delays. The spectral properties of graph Laplacians can be used to analyze the convergence of the oscillator network.

- The flocks of mobile agents equipped with sensing and communication devices is another application for the consensus protocol. These mobile robots can serve as mobile sensor networks for distributed sensing in an environment [41]. A design and analysis of flocking algorithms for mobile robots with obstacle-avoidance capabilities can be found in [42]. Generally, the consensus algorithm can make the agent in flocking to achieve velocity matching with respect to its neighbors.

- Another application of the consensus is the rendezvous activity. This is equivalent to reaching a consensus in position with an interaction topology graph in [43]. This problem is challenging under the switching of the network topology.

- Moreover, the distributed sensor fusion in sensor networks also uses the principle of the consensus protocol. Implementing some methods such as Kalman filter [44] by the consensus manner for various sensors, a distributed sensor fusion can be done.

2.3 Privacy Protection

In some applications, such as surveillance and monitoring network, the designer of the network would not want the information collected by the network to be leaked. In this scenario, the participating agents in the network would not want to release more information about its initial value than strict necessary to reach the consensus agreement. Thus, the privacy protection is important for the consensus protocol.

A privacy preserving average consensus algorithm is proposed in [45], where the algorithm can compute the exact average of the initial values. Meantime, it can ensure that the initial value of each agent cannot be perfectly inferred by other participating agents. In this method, the agent needs to design a correlated noise process to ensure that the noise does not affect the consensus result. In order to converge to the exact average, the noise is designed to be decaying. Thus, the asymptotic sum of the noise needs to be 0 to avoid affecting the results.

Another choice is to use the differential privacy technique. The concept of the differential privacy comes from the database literature [46]. After that, a popular adopted differentially private mechanism is to be used in the database query to guarantee that the data stored from a wide users will be protected from the external observer. Recently, this notation is borrowed by the dynamical system. In [47], by adding white Gaussian perturbations to the dynamical system, a differentially private filter is designed. The differentially private mechanisms randomize the responses to dataset analysis requests and guarantee that whether or not an individual chooses to contribute his data only marginally changes the distribution over the outputs. Consequently, the adversary who can acquire these outputs cannot infer much more information of the individuals after the publication of the outputs. Then, a differential private Kalman filter is proposed to release the output of the dynamical system while preserving differential privacy for the inputs. They also consider the systems processing as a single integer-valued signal describing the occurrence of events originating from different participants. The differential private version of the iterative averaging algorithm is proposed in [48], where the private consensus problem is studied. The agents need to preserve the privacy of their initial values from an adversary who can access all the messages exchanged and these agents finally reach the agreement. This algorithm can protect the initial value of the agent instead of its participation status.

2.4 Deep Reinforcement Learning

Reinforcement learning (RL) formalizes control problems generally as finding a policy π that can maximize expected future rewards. Value functions $V(s)$ is important to the RL, and they can catch the utility of any state s in achieving the agent's overall objective. Recently, value functions have also been generalized as $V(s, g)$ in order to represent the utility of state s for achieving a given goal $g \in G$ [49]. When the environment provides delayed rewards, we adopt a strategy to first learn ways to achieve intrinsically generated goals, and subsequently learn an optimal policy to chain them together. Each of the value function $V(s, g)$ can be used to generate a policy that terminates when the agent reaches the goal state g. A collection of these policies can be hierarchically arranged with temporal dynamics for learning or planning within the framework of semi-Markov decision processes [50]. In some high-dimensional problems, these value functions can be approximated by neural networks as $V(s, g : \theta)$.

Recently, the advancement in function approximation with deep neural networks has shown promise in handling high-dimensional sensory input. Deep Q-Networks and its variants have been successfully applied to various domains including Atari games [51] and Go [52].

However, for some challenging physical control problems that involve complex multi-joint movements, unstable and rich contact dynamics, the DQN is not sufficient to solve it. Thus, the actor-critic approach with insights from the DQN is developed. By doing so, the network is trained off-policy with samples from a replay buffer to minimize the correlation between samples and the network is trained with a target Q network to give consistent targets during temporal difference backups.

The nature of interaction between agents can be cooperative is considered in some DRL algorithm. Most studies stress the strategies such as Q function update [53], which assume that the actions of other agents made can improve collective reward. Another method is to indirectly arrive at cooperation via sharing of policy parameters [54], however, this method needs the requirement that the agent is homogeneous.

In [55], the authors propose a DRL framework by using policy gradient method with a centralized critic, and test their approach on a StarCraft micromanagement task. This method can learn a single centralized critic for all agents. Moreover, they combine recurrent policies with feed-forward critics.

CHAPTER 3

THE BIO-INSPIRED COOPERATIVE LEARNING CONSENSUS UNDER SUGGESTED CONVERGENCE DIRECTION: LINEAR CASE

3.1 Introduction

While the consensus problem for multiagent systems has drawn a great attention in recent years in different areas, it was until recently that some analogy between this problem and swarm intelligence algorithms, such as particle swarm optimization, has been noticed by [33,56]. This similarity has inspired us to improve the performance of swarm intelligence algorithms by modifying them using some techniques from the various consensus protocols in the literature. Such a combination from a control problem and a computational intelligence algorithm offers a brand new perspective to design efficient swarm intelligence algorithms, not just from the bio-inspired direction, but also from the control-theoretic methodology, leading to a one-way exploration from control theory to swarm optimization. Now the question lies in the other direction: *Is it possible to design consensus protocols for multiagent systems using some techniques or concepts from swarm intelligence, so that the state convergence direction of these systems can be guided, but not completely given, for flexibility?* This is the question we will address in this chapter, and an affirmative answer will be given to this question. Hence, a two-way, positive feedback of mutual exploration and interplay is unraveled between networked control theory and swarm optimization based algorithms, based on the result in this chapter and the results in [33, 56].

We will address the above question by designing new consensus protocols with two additional attributes of agents being "smart" to data transmission and their state convergence direction being totally guidable but not totally controllable. In this chapter, motivated by the multiagent coordination optimization (MCO) algorithm [33, 56] and the bat searching algorithm [35], a new bat-inspired consensus protocol is proposed. More specifically, by incorporating a separate, unrelated optimization problem into the protocol, our new consensus algorithm can fully guide its state convergence direction leaning toward the best solution (i.e., the optimal solution among the population of candidate solutions) to this separate, unrelated optimization problem. At the same time, although the optimal solution to this optimization problem may always exist (e.g., convex optimization), its best solution form may not be precisely calculated or numerically found. Hence, such an issue actually creates an uncertainty for exactly predicting the final state convergence direction, which turns out to be a good merit for protecting multiagent systems from adversaries. Moreover, the proposed consensus algorithm further takes advantage of the mechanism behind the BS algorithm to enhance agents' data transmission capability so that they become "smart" enough to not only process the neighboring and their own data, but also relay the processed data among agents in a multi-hop way.

Thus, the most notable feature of the proposed cooperative learning consensus protocols is their ability to simultaneously solve an optimization problem and a consensus problem altogether. The embedded optimization problem serves as a suggested convergence direction for the consensus problem. When the proposed cooperative learning consensus protocols run in a convergence way, both problems obtain their solutions accordingly. This feature is particularly appealing in high performance computing in which multitask jobs are quite common in many parallel computing problems. Hence, the proposed method paves a way to develop corresponding algorithms for parallel solutions to multitask computing and optimization problems. A detailed convergence analysis of the proposed cooperative learning consensus protocols will be presented in this chapter. In the end, the numerical comparison between the proposed cooperative learning consensus protocols and the average consensus [12, 13] is provided to show the features of the proposed ones. Specifically, not only do the proposed consensus protocols converge under certain conditions like the average consensus [12, 13], but also the proposed consensus protocols optimize a direction function during the convergence process.

3.1.1 Swarm-Intelligence-Inspired Consensus

To begin with, we define some time-dependent, algebraic, graph-related notations to describe our cooperative learning consensus protocols. Specifically, let $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ denote a *dynamic directed graph* (or *dynamic digraph*) with the set of vertices $\mathcal{V} = \{v_1, v_2, \ldots, v_q\}$ and $\mathcal{E}(t) \subseteq \mathcal{V} \times \mathcal{V}$ representing the set of edges, where $t \in \overline{\mathbb{Z}}_+ = \{0, 1, 2, \ldots\}$. The time-varying matrix $A(t) \in \mathbb{R}^{q\times q}$ with nonnegative adjacency elements $a_{i,j}(t)$ associated with $\mathcal{E}(t)$ serves as the *adjacency matrix* of $\mathcal{G}(t)$, where $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ denotes the set of n-dimensional real column vectors, and $\mathbb{R}^{n\times m}$ denotes the set of n-by-m real matrices. The node index of $\mathcal{G}(t)$ is denoted as a finite index set $\mathcal{N} = \{1, 2, \ldots, q\}$. An edge of $\mathcal{G}(t)$ is denoted by $e_{i,j}(t) = (v_i, v_j)$, and the adjacency elements associated with the edges are positive. We assume $e_{i,j}(t) \in \mathcal{E}(t)$ if and only if $a_{i,j}(t) = 1$, $e_{i,j}(t) \notin \mathcal{E}(t)$ if and only if $a_{i,j}(t) = 0$, and $a_{i,i}(t) = 0$ for all $i \in \mathcal{N}$. The set of neighbors of the node v_i is denoted by $\mathcal{N}^i(t) = \{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}(t), j = 1, 2, \ldots, |\mathcal{N}|, j \neq i\}$, where $|\mathcal{N}|$ denotes the cardinality of $\mathcal{N}$. In many cases, for brevity, we simply take $\mathcal{V} = \mathcal{N}$. The *degree matrix* of a dynamic digraph $\mathcal{G}(t)$ is defined as $D(t) = [d_{i,j}(t)]_{i,j=1,2,\ldots,|\mathcal{N}|}$, where $d_{i,j}(t) = \begin{cases} \sum_{j=1}^{|\mathcal{N}|} a_{i,j}(t), & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$ The *Laplacian matrix* of the dynamic digraph $\mathcal{G}(t)$ is defined by $L(t) = D(t) - A(t)$. If $L(t) = L^{\mathrm{T}}(t)$, where $(\cdot)^{\mathrm{T}}$ denotes the transpose operation, then $\mathcal{G}(t)$ is called a *dynamic undirected graph* (or simply *dynamic graph*). If there is a path from any node to any other node in a dynamic digraph, then the dynamic digraph is called *strongly connected*. From now on, we use short notations, $A_t, D_t, L_t, \mathcal{G}_t, \mathcal{N}_t^i$, to denote $A(t), D(t), L(t), \mathcal{G}(t), \mathcal{N}^i(t)$, respectively.

Consider a group of q bats (agents) who have directional communications via a communi cation digraph topology $\mathcal{G}_t$ at each time instant t. Each node k in $\mathcal{G}_t$ corresponds to a labeled bat k, $k = 1, \ldots, q$. Throughout this chapter, we make the following two standing assumptions. The first one is about the connectivity of $\mathcal{G}_t$.

Assumption 1. The communication digraph $\mathcal{G}_t$ is strongly connected.

The second one is about a separate optimization problem embedded in the proposed consensus protocol.

Assumption 2. The minimization problem $\min_{\boldsymbol{x} \in \mathbb{R}^n} F(\boldsymbol{x})$ has a solution, where $F : \mathbb{R}^n \to \mathbb{R}$.

The original BS algorithm was based on the echolocation or bio-sonar characteristics of microbats [35]. More specifically, the bats can update their position information by following the below position-velocity rules to find their "prey", which tends to be the best solution to an optimization problem $\min_{\boldsymbol{x} \in \mathbb{R}^n} F(\boldsymbol{x})$:

$$f_i(t) = f_{\min} + (f_{\max} - f_{\min})\beta_i, \tag{3.1}$$

$$v_i(t+1) = v_i(t) + [x_i(t) - p(t)]f_i(t), \tag{3.2}$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad i = 1, \ldots, q \tag{3.3}$$

where $x_i(t) \in \mathbb{R}^n$ and $v_i(t) \in \mathbb{R}^n$ are the position and velocity of Bat i at each time instant t, respectively, $f_i(t) \in \mathbb{R}$ is the frequency information for Bat i at time instant t, $f_{\min}$ and $f_{\max}$ are the lower bound and upper bound of the frequency for Bat i, respectively, $\beta_i \in [0, 1]$ is a random coefficient drawn from a uniform distribution, and $p(t) \in \mathbb{R}^n$ is the current best global solution at time instant t, i.e., $p(t) = \arg\min_{1\leq i\leq q, 0\leq s\leq t} F(x_i(s))$.

Based on this algorithm, we propose a cooperative learning consensus protocol for this group of q bats. The scenario we are considering here is that all of the bats have the same constant speed, but with different heading angles. From the control-theoretic perspective, a consensus protocol for heading angles of the bats is a semi-distributed control algorithm used to asymptotically achieve a common heading angle among all of the bats. Using this concept, the proposed cooperative learning consensus protocol for heading angles of the bats, under a given minimization problem $\min_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x})$ as its suggested convergence direction, is given as follows:

$$\boldsymbol{\theta}_{1,\min}(t) = \boldsymbol{\theta}_1(t), \tag{3.4}$$

$$\boldsymbol{\theta}_{k+1,\min}(t) = \arg\min\{F(\boldsymbol{\theta}_{k,\min}(t)), F(\boldsymbol{\theta}_{k+1}(t))\}, \quad k = 1, \ldots, 2q-2 \tag{3.5}$$

$$\boldsymbol{\theta}_{2q,\min}(t) = \boldsymbol{\theta}_{2q-1,\min}(t), \tag{3.6}$$

$$\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) + f_i(t)\left\{ \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^n} \sum_{j\in\mathcal{N}_t^i} [\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}]^{\mathrm{T}}\Phi_{i,j}(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}) - \boldsymbol{\theta}_i(t)\right\}$$
$$+ f_i(t)\mu_i(t)[\boldsymbol{\theta}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)], \tag{3.7}$$

$$f_i(t) = f_{\min} + \beta_i(t)(f_{\max} - f_{\min}), \quad i = 1, \ldots, q \tag{3.8}$$

where $t \in \overline{\mathbb{Z}}_+$, $\boldsymbol{\theta}_i(t) = \boldsymbol{\theta}_{q+i}(t) \in \mathbb{R}^n$ denotes the heading angle vector of Bat i at iteration t, respectively, $f_i(t) > 0$ is the frequency of Bat i, $f_{\min} > 0$ is a given lowerbound of the frequency, $f_{\max} > 0$ is a given upperbound of the frequency, $\Phi_{i,j} : \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued *flux function* satisfying $\Phi_{i,j}(\boldsymbol{x}) = 0$ if and only if $\boldsymbol{x} = 0$ and $\boldsymbol{x}^{\mathrm{T}}\Phi_{i,j}(\boldsymbol{x}) \geq 0$ for every $\boldsymbol{x} \in \mathbb{R}^n$ and every $i, j = 1, \ldots, q$, $i \neq j$, $0 \leq \beta_i(t) \leq 1$ is a normalized range parameter for the frequency, $0 < \mu_{\min} \leq \mu_i(t) \leq \mu_{\max}$ is the zooming parameter for Bat i, $i = 1, \ldots, q$, $t \in \overline{\mathbb{Z}}_+$, $\boldsymbol{\theta} \in \mathbb{R}^n$ is a vector variable, and $\boldsymbol{\theta}_{q+i,\min}(t)$ is defined as

$\arg\min_{1\leq i\leq q} F(\boldsymbol{\theta}_i(t))$, which denotes the *suggested* convergence direction.

For brevity, we use the short notations β_t^i, f_t^i, and μ_t^i to denote $\beta_i(t)$, $f_i(t)$, and $\mu_i(t)$ for every $i = 1, \ldots, q$, respectively.

The flux function $\Phi_{i,j}(\cdot)$ can be interpreted as an energy/mass exchange rate in compartmental modeling [15, 57] or a heat transfer rate in thermodynamics [58]. Furthermore, $\Phi_{i,j}(\cdot)$ is *not* necessarily convex or *not* necessarily continuous. Next, the original BS algorithm does not have the interconnected term $\arg\min_{\boldsymbol{\theta}\in\mathbb{R}^n} \sum_{j\in\mathcal{N}_t^i}[\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}]^{\mathrm{T}}\Phi_{i,j}(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta})$ in (3.7). The addition of this term is motivated by the speed-up and speed-down strategy derived from biological swarms [59]. This term is calculated through a *multihop relay protocol* [60] based on the communication ring routing path. More specifically, it consists of the following two steps:

1) Bat $k+1$ can receive the information of $\boldsymbol{\theta}_{k,\min}(t)$ from Bat k at time instant t, $k = 1, \ldots, q-1$. At the same time, Bat $k+1$ determines $\boldsymbol{\theta}_{k+1,\min}(t) = \arg\min\{F(\boldsymbol{\theta}_{k,\min}(t))$ $, F(\boldsymbol{\theta}_{k+1}(t))\}$ and serves as a router to send $\boldsymbol{\theta}_{k+1,\min}(t)$ to the next bat.

2) After $\boldsymbol{\theta}_{q,\min}(t)$ is determined by Bat q, this information is passed to Bat $(q+1 \mod q)$, which is essentially Bat 1, where $\mod$ denotes the modulo operation. Bat $(k \mod q)$, $k = q+1, \ldots, 2q-1$, again determines $\boldsymbol{\theta}_{k,\min}(t) = \arg\min\{F(\boldsymbol{\theta}_{k-1,\min}(t))$, $F(\boldsymbol{\theta}_{(k \mod q)}(t))\}$ and serves as a router to send $\boldsymbol{\theta}_{k,\min}(t)$ to Bat $(k+1 \mod q)$ by sequentially following the directed communication path

$$\begin{aligned}\text{Bat}\, q \to \text{Bat}\, (q+1 \mod q) &\to \text{Bat}\, (q+2 \mod q)\\ &\to \cdots \to \text{Bat}\, (2q-1 \mod q)\end{aligned}$$

which is equivalent to

$$\text{Bat}\, q \to \text{Bat}\, 1 \to \text{Bat}\, 2 \to \cdots \to \text{Bat}\, q-1$$

.

Note that we used a "double-check" technique in these two steps to obtain $\arg\min_{1\le i\le q} F(\boldsymbol{\theta}_i(t))$. It is clear that after Step 1, $\boldsymbol{\theta}_{q,\min}(t)$ obtained by Bat q is indeed $\arg\min_{1\le i\le q} F(\boldsymbol{\theta}_i(t))$. Hence, in Step 2, $\boldsymbol{\theta}_{q+i,\min}(t)$ obtained by Bat i is identical to $\boldsymbol{\theta}_{q,\min}(t)$ for every $i = 1, \ldots, q-1$. However, we still let Bat i perform the comparison operation $\boldsymbol{\theta}_{q+i,\min}(t) = \arg\min\{F(\boldsymbol{\theta}_{q+i-1,\min}(t)), F(\boldsymbol{\theta}_i(t))\}$ in Step 2 to ensure that we end up with the correct $\arg\min_{1\le i\le q} F(\boldsymbol{\theta}_i(t))$.

Hence, the proposed cooperative learning consensus protocol is distinct from the existing consensus protocols in the literature. Moreover, the proposed cooperative learning consensus protocol is a semi-distributed, localized algorithm by determining $\boldsymbol{\theta}_{q+i,\min}(t) = \arg\min_{1\le i\le q} F(\boldsymbol{\theta}_i(t))$ locally, unlike the BS algorithm which computes $\arg\min_{1\le i\le q} F(\boldsymbol{\theta}_i(t))$ in a global manner in the sense that all of the bats' position information is shared in the group.

A fundamental question regarding the cooperative learning consensus protocol (3.4)–(3.8) is about its (absolute) convergence property. How can we guarantee the convergence of (3.4)–(3.8) for a given minimization problem $\min_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x})$ and arbitrary initial condition? Here, the precise meaning of (absolute) convergence is that $\lim_{t\to\infty} \boldsymbol{\theta}_1(t) = \cdots = \lim_{t\to\infty} \boldsymbol{\theta}_q(t)$ exists for (3.4)–(3.8) with any initial condition $\boldsymbol{\theta}_i(0) \in \mathbb{R}^n$, $i = 1, \ldots, q$. To answer this question for (3.4)–(3.8), we first need to study the form or the property of $\Phi_{i,j}(\cdot)$. The most commonly used one is the linear form

$$\Phi_{i,j}(\boldsymbol{x}) = \boldsymbol{x}. \tag{3.9}$$

Other forms include the signum form $\Phi_{i,j}(\boldsymbol{x}) = \mathrm{sgn}(\boldsymbol{x})$, where sgn denotes the signum function and $\mathrm{sgn}(\boldsymbol{x})$ denotes an elementwise operation for the vector $\boldsymbol{x}$. In this chapter, we consider both the linear form (3.9) and some nonlinear form for $\Phi_{i,j}(\cdot)$.

The basic idea of conducting convergence analysis for (3.4)–(3.8) under the linear form of $\Phi_{i,j}(\cdot)$ is to convert the proposed iterative algorithm into a discrete-time linear time-varying system and then to discuss its convergence property using some matrix analysis tools. It is motivated by some recent works done in semistable control and paracontraction analysis [61,62]. More specifically, we consider the discrete-time linear time-varying system given by the form

$$X(t+1) = W(t)X(t), \quad t \in \overline{\mathbb{Z}}_+ \tag{3.10}$$

where $X(t) = [\boldsymbol{\theta}_1^{\mathrm{T}}(t), \ldots, \boldsymbol{\theta}_q^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^n \times \cdots \times \mathbb{R}^n = \mathbb{R}^{nq}$. Then in this case the cooperative learning consensus protocol given by (3.4)–(3.8) can be rewritten as the compact form (3.10) by defining a corresponding $W(\cdot)$ appropriately. The frequency equation (3.8) stands alone with the rest of the equations, and it can be viewed as a time-dependent parameter in the cooperative learning consensus protocol. Thus, the convergence analysis of the proposed cooperative learning consensus protocol can be converted into a convergence problem of a discrete-time linear time-varying system given by the form (3.10). Here two different approaches will be used for the convergence analysis of (3.10): the matrix paracontraction approach [63] and the nonnegative matrix approach [64].

3.1.2 Mathematical Preliminaries

Paracontraction is a nonexpansive property for a class of linear operators which can be used to study convergence of linear iterations [63, 65], communication protocols [66], and biomimetic models [67]. In this chapter, we will use this idea to derive sufficient convergence conditions for (3.10). To this end, some new results on matrix paracontraction will be developed in this section. These new results play a key role to derive simple sufficient conditions to guarantee the convergence of the proposed cooperative learning consensus protocol. They also disclose some interesting properties for paracontraction that have not been discovered before and complement many existing paracontraction results in the literature [62, 63, 65, 67]. The following definition, due to [63], gives the notion of paracontracting matrices.

Definition 1 ([63]). Let $W \in \mathbb{R}^{n\times n}$. W is called *paracontracting* if for any $x \in \mathbb{R}^n$, $Wx \neq x$ is equivalent to $\|Wx\| < \|x\|$, where $\|\cdot\|$ denotes the (vector and matrix) 2-norm.

Next, we introduce the following fact needed later in the chapter.

Lemma 1. Let $W \in \mathbb{R}^{q\times q}$. Then $\|W\| \leq 1$ if and only if $W^{\mathrm{T}}W \leq I_q$, where I_q denotes the q-by-q identity matrix. Furthermore, $\|W\| \leq 1$ if and only if $WW^{\mathrm{T}} \leq I_q$.

Proof. First, it follows from Proposition 9.4.9 of [68, p. 609] that $\|W\| = \sigma_{\max}(W)$, where $\sigma_{\max}(W)$ denotes the maximum singular value of W. Next, it follows from Fact 5.11.35 of [68, p. 358] that $\sigma_{\max}(W) \leq 1$ if and only if $W^{\mathrm{T}}W \leq I_q$. The second conclusion is a direct consequence of the first one by noting that $\|W\| = \|W^{\mathrm{T}}\|$. □

The following result, motivated by Lemma 3.5 of [62], connects paracontraction with the singular value decomposition.

Lemma 2. Let $W \in \mathbb{R}^{q\times q}$ and $r = \mathrm{rank}(W)$, where $\mathrm{rank}(W)$ denotes the rank of W. Suppose that the singular value decomposition of W is given by $W = V \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} U^{\mathrm{T}}$, where $\Sigma = \mathrm{diag}(\sigma_1(W), \ldots, \sigma_r(W)) \in \mathbb{R}^{r\times r}$, $\mathrm{diag}(X)$ denotes a diagonal matrix whose ith diagonal entry is the ith component of X, $U = [u_1, \ldots, u_q] \in \mathbb{R}^{q\times q}$ and $V = [v_1, \ldots, v_q] \in \mathbb{R}^{q\times q}$ are orthogonal matrices, $u_i \in \mathbb{R}^q$, and $v_i \in \mathbb{R}^q$. Define m to be the positive integer satisfying $\|W\| = \sigma_1(W) = \cdots = \sigma_m(W) > \sigma_{m+1}(W) \geq \cdots \geq \sigma_r(W) >$

0. Then $\ker(W^{\mathrm{T}}W - \|W\|^2 I_q) = \mathrm{span}\{u_1, \ldots, u_m\}$ and $\ker(WW^{\mathrm{T}} - \|W^{\mathrm{T}}\|^2 I_q) = \mathrm{span}\{v_1, \ldots, v_m\}$, where $\ker(A)$ denotes the kernel of A and $\mathrm{span}\,\mathcal{S}$ denotes the span of $\mathcal{S}$.

Proof. For every $x \in \ker(W^{\mathrm{T}}W - \|W\|^2 I_q)$, let $x = \sum_{i=1}^{q} \alpha_i u_i$, where $\alpha_i \in \mathbb{R}$. It follows from the singular value decomposition of W that $W = \sum_{i=1}^{r} \sigma_i(W) v_i u_i^{\mathrm{T}}$. Hence, $W^{\mathrm{T}}W = \sum_{i=1}^{r} \sigma_i^2(W) u_i u_i^{\mathrm{T}}$ and $x^{\mathrm{T}}W^{\mathrm{T}}Wx = \sum_{i=1}^{r} \sigma_i^2(W)\alpha_i^2$. Note that $x^{\mathrm{T}}x = \sum_{i=1}^{q} \alpha_i^2$. Thus, $\|Wx\|^2 - \|W\|^2\|x\|^2 = 0$ if and only if $\sum_{i=1}^{r} \sigma_i^2(W)\alpha_i^2 - \sum_{i=1}^{q} \|W\|^2\alpha_i^2 = 0$. Note that $0 = \sum_{i=1}^{r} \sigma_i^2(W)\alpha_i^2 - \sum_{i=1}^{q} \|W\|^2\alpha_i^2 = \sum_{i=1}^{r}[\sigma_i^2(W) - \|W\|^2]\alpha_i^2 - \sum_{j=r+1}^{q} \|W\|^2\alpha_j^2 = \sum_{i=m+1}^{r}[\sigma_i^2(W) - \|W\|^2]\alpha_i^2 - \sum_{j=r+1}^{q} \|W\|^2\alpha_j^2$. Since $\sigma_i(W) < \|W\|$ for every $i = m+1, \ldots, r$, this equality holds if and only if $\alpha_s = 0$ for every $s = m+1, \ldots, q$. This implies that $x = \sum_{i=1}^{m} \alpha_i u_i$ for arbitrary $\alpha_i \in \mathbb{R}$, $i = 1, \ldots, m$. Hence, $\ker(W^{\mathrm{T}}W - \|W\|^2 I_q) = \mathrm{span}\{u_1, \ldots, u_m\}$.

Analogously, for every $y \in \ker(WW^{\mathrm{T}} - I_q)$, let $y = \sum_{i=1}^{q} \beta_i v_i$, where $\beta_i \in \mathbb{R}$. Using $W = \sum_{i=1}^{r} \sigma_i(W) v_i u_i^{\mathrm{T}}$, $WW^{\mathrm{T}} = \sum_{i=1}^{r} \sigma_i^2(W) v_i v_i^{\mathrm{T}}$, $\|W^{\mathrm{T}}\| = \|W\|$, and the similar arguments as above, it follows that $\ker(WW^{\mathrm{T}} - \|W^{\mathrm{T}}\|^2 I_q) = \mathrm{span}\{v_1, \ldots, v_m\}$. □

The next result is due to Proposition 3.2 of [62].

Lemma 3. Let $W \in \mathbb{R}^{q\times q}$ and $r = \mathrm{rank}(W)$. Suppose that the singular value decomposition of W is given by $W = V \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} U^{\mathrm{T}}$, where $\Sigma = \mathrm{diag}(\sigma_1(W), \ldots, \sigma_r(W)) \in \mathbb{R}^{r\times r}$, and $U \in \mathbb{R}^{q\times q}$ and $V \in \mathbb{R}^{q\times q}$ are orthogonal matrices. Define m to be the positive integer satisfying $\|W\| = \sigma_1(W) = \cdots = \sigma_m(W) > \sigma_{m+1}(W) \geq \cdots \geq \sigma_r(W) > 0$. Then $\|W\| \leq 1$ and $\ker(W^{\mathrm{T}}W - I_q) = \ker(WW^{\mathrm{T}} - I_q)$ if and only if W is paracontracting.

Proof. It follows from Proposition 3.2 of [62] that W is paracontracting if and only if $\|W\| \leq 1$ and $u_i = v_i$, $i = 1, \ldots, m$, where u_i, v_i, and m are defined in Lemma 2. If $\|W\| = \|W^{\mathrm{T}}\| = 1$, then it follows from Lemma 2 that $\ker(W^{\mathrm{T}}W - I_q) = \mathrm{span}\{u_1, \ldots, u_m\}$ and $\ker(WW^{\mathrm{T}} - I_q) = \mathrm{span}\{v_1, \ldots, v_m\}$. Hence, the conclusion holds. Otherwise, if

$\|W\| = \|W^{\mathrm{T}}\| < 1$, then 1 is not an eigenvalue of W and W^{T}. Hence, $\ker(W^{\mathrm{T}}W - I_q) = \{0\} = \ker(WW^{\mathrm{T}} - I_q)$. □

For a matrix $W \in \mathbb{R}^{q\times q}$ whose 2-norm is less than or equal to 1, the following result says that its complement $W - I_q$ is range symmetric.

Lemma 4. Let $W \in \mathbb{R}^{q\times q}$ and $\|W\| \leq 1$. Then $\ker(W - I_q) = \ker(W^{\mathrm{T}} - I_q)$.

Proof. If 1 is not an eigenvalue of W, then both $W - I_q$ and $W^{\mathrm{T}} - I_q$ are nonsingular. In this case, $\ker(W - I_q) = \{0\} = \ker(W^{\mathrm{T}} - I_q)$. Now, we assume that 1 is an eigenvalue of W. Note that for nonzero $x \in \ker(W - I_q)$, we have $x = Wx$. Now it follows from the Cauchy-Schwarz inequality and $\|W^{\mathrm{T}}\| = \|W\| \leq 1$ that $\|x\|^2 = x^{\mathrm{T}}x = (Wx)^{\mathrm{T}}x = x^{\mathrm{T}}W^{\mathrm{T}}x \leq \|x\|\|W^{\mathrm{T}}x\| \leq \|x\|\|W^{\mathrm{T}}\|\|x\| = \|W\|\|x\|^2 \leq \|x\|^2$, which implies that $x^{\mathrm{T}}W^{\mathrm{T}}x = \|x\|\|W^{\mathrm{T}}x\|$. This equality holds if and only if $W^{\mathrm{T}}x = \lambda x$ for some $\lambda \in \mathbb{C}$, where $\mathbb{C}$ denotes the set of complex numbers. Since $\|W^{\mathrm{T}}\| = \|W\| \leq 1$, it follows that $|\lambda| \leq 1$. Next, it follows from $W^{\mathrm{T}}x = \lambda x$ and $x = Wx$ that $\lambda^* x^{\mathrm{T}}x = (W^{\mathrm{T}}x)^{\mathrm{T}}x = x^{\mathrm{T}}Wx = x^{\mathrm{T}}x$, which implies that $\lambda = \lambda^* = 1$, where λ^* denotes the complex conjugate of λ. Hence, $W^{\mathrm{T}}x = x$, which indicates that $x \in \ker(W^{\mathrm{T}} - I_q)$, leading to $\ker(W - I_q) \subseteq \ker(W^{\mathrm{T}} - I_q)$. Similarly, for nonzero $x \in \ker(W^{\mathrm{T}} - I_q)$, it follows from the similar arguments as above that $Wx = x$, and hence, $x \in \ker(W - I_q)$, leading to $\ker(W^{\mathrm{T}} - I_q) \subseteq \ker(W - I_q)$. Thus, $\ker(W^{\mathrm{T}} - I_q) = \ker(W - I_q)$. □

Remark 1. An alternative, equivalent statement of Lemma 4 is that if $\ker(W - I_q) \neq \ker(W^{\mathrm{T}} - I_q)$ for some square matrix $W \in \mathbb{R}^{q\times q}$, then $\|W\| > 1$. ♦

Now we have a series of necessary and sufficient conditions for paracontraction based on Lemmas 1–4.

Theorem 1. Let $W \in \mathbb{R}^{q\times q}$. Then the following statements are equivalent:

$i)$ W is paracontracting.

$ii)$ W^{T} is paracontracting.

$iii)$ $\|W\| \leq 1$ and $\ker(W^{\mathrm{T}}W - I_q) = \ker(W - I_q)$.

$iv)$ $\|W\| \leq 1$ and $\ker(WW^{\mathrm{T}} - I_q) = \ker(W^{\mathrm{T}} - I_q)$.

$v)$ $W^{\mathrm{T}}W \leq I_q$ and $\ker(W^{\mathrm{T}}W - I_q) = \ker(WW^{\mathrm{T}} - I_q)$.

$vi)$ $WW^{\mathrm{T}} \leq I_q$ and $\ker(WW^{\mathrm{T}} - I_q) = \ker(W - I_q)$.

Proof. The equivalence between $i)$ and $iii)$ follows directly from Lemma 3.2 of [62]. Likewise, the equivalence between $ii)$ and $iv)$ follows from Lemma 3.2 of [62] as well. Next, it follows from Lemma 3 that $i)$ is equivalent to $\|W\| \leq 1$ and $\ker(W^{\mathrm{T}}W - I_q) = \ker(WW^{\mathrm{T}} - I_q)$. By Lemma 1, $\|W\| \leq 1$ is equivalent to $W^{\mathrm{T}}W \leq I_q$. Hence, $i)$ is equivalent to $v)$.

To show the equivalence between $i)$ and $iv)$, note that we have shown the equivalence between $i)$ and $iii)$, the equivalence between $ii)$ and $iv)$, and the equivalence between $i)$ and $v)$. Hence, if $i)$ holds, then by $v)$, $W^{\mathrm{T}}W \leq I_q$ and $\ker(W^{\mathrm{T}}W - I_q) = \ker(WW^{\mathrm{T}} - I_q)$. On the other hand, if $i)$ holds, it follows from $iii)$ and Lemma 3 that $\ker(W^{\mathrm{T}}W - I_q) = \ker(W - I_q) = \ker(WW^{\mathrm{T}} - I_q)$. In this case, by Lemma 4, $\ker(W^{\mathrm{T}}W - I_q) = \ker(W - I_q) = \ker(WW^{\mathrm{T}} - I_q) = \ker(W^{\mathrm{T}} - I_q)$. Hence, if $i)$ holds, then $iv)$ holds.

Alternatively, if $iv)$ holds, then it follows from the equivalence between $ii)$ and $iv)$ that $ii)$ holds. Now it follows from the equivalence between $i)$ and $v)$ that if $ii)$ holds, then $WW^{\mathrm{T}} \leq I_q$ and $\ker(WW^{\mathrm{T}} - I_q) = \ker(W^{\mathrm{T}}W - I_q)$. By Lemma 1, $WW^{\mathrm{T}} \leq I_q$ is equivalent to $\|W\| \leq 1$. In this case, it follows from Lemma 4 that $\ker(W^{\mathrm{T}} - I_q) = \ker(W - I_q)$. Thus, if $iv)$ holds, then $\|W\| \leq 1$ and $\ker(WW^{\mathrm{T}} - I_q) = \ker(W^{\mathrm{T}} - I_q) = \ker(W^{\mathrm{T}}W - I_q) = \ker(W - I_q)$. Finally, it follows from the equivalence between $i)$ and $iii)$ that if $\|W\| \leq 1$ and $\ker(W^{\mathrm{T}}W - I_q) = \ker(W - I_q)$, then $i)$ holds.

Finally, the equivalence between $iv)$ and $vi)$ is immediate from Lemmas 1 and 4. $\square$

Next, we present a matrix analysis result to connect null spaces of two matrices with their ranks.

Lemma 5. Let $A \in \mathbb{R}^{m\times n}$ and $C \in \mathbb{R}^{l\times n}$. Then $\ker(A) = \ker(C)$ if and only if $\mathrm{rank}(A) = \mathrm{rank}(C) = \mathrm{rank}\begin{bmatrix} A \\ C \end{bmatrix}$.

Proof. Note that both $\ker(A)$ and $\ker(C)$ are subspaces. Then it follows from Lemma 6 in [20] that $\ker(A) = \ker(C)$ if and only if $\ker(A)^{\perp} = \ker(C)^{\perp}$, where $\ker(A)^{\perp}$ denotes the orthogonal complement of $\ker(A)$. Next, it follows from Theorem 2.4.3 of [68, p. 103] that $\ker(A)^{\perp} = \mathrm{ran}(A^{\mathrm{T}})$ and $\ker(C)^{\perp} = \mathrm{ran}(C^{\mathrm{T}})$, where $\mathrm{ran}(A)$ denotes the range space of A. Hence, $\ker(A) = \ker(C)$ if and only if $\mathrm{ran}(A^{\mathrm{T}}) = \mathrm{ran}(C^{\mathrm{T}})$. Now it follows from Fact 2.11.5 of [68, p. 131] that $\mathrm{ran}(A^{\mathrm{T}}) = \mathrm{ran}(C^{\mathrm{T}})$ if and only if $\mathrm{rank}(A^{\mathrm{T}}) = \mathrm{rank}(C^{\mathrm{T}}) = \mathrm{rank}\begin{bmatrix} A^{\mathrm{T}} & C^{\mathrm{T}} \end{bmatrix}$. Since $\mathrm{rank}(K) = \mathrm{rank}(K^{\mathrm{T}})$ for any matrix K, the conclusion follows immediately. □

Motivated by Theorem 1 and Lemma 5, we have the following rank results for paracontraction.

Theorem 2. Let $W \in \mathbb{R}^{q\times q}$. Then the following statements are equivalent:

$i)$ W is paracontracting.

$ii)$ $\|W\| \le 1$ and

$$\mathrm{rank}(W^{\mathrm{T}}W - I_q) = \mathrm{rank}(W - I_q) = \mathrm{rank}\begin{bmatrix} W^{\mathrm{T}}W - I_q \\ W - I_q \end{bmatrix}. \tag{3.11}$$

$iii)$ $\|W\| \le 1$ and

$$\mathrm{rank}(WW^{\mathrm{T}} - I_q) = \mathrm{rank}(W - I_q) = \mathrm{rank}\begin{bmatrix} WW^{\mathrm{T}} - I_q & W - I_q \end{bmatrix}. \tag{3.12}$$

Proof. It follows from Lemma 5 that (3.11) holds if and only if $\ker(W^{\mathrm{T}}W - I_q) = \ker(W - I_q)$. Likewise, (3.12) holds if and only if $\ker(WW^{\mathrm{T}} - I_q) = \ker(W^{\mathrm{T}} - I_q)$. Now the equivalence between i) and ii) in this result follows from the equivalence between i) and iii) in Theorem 1. Also, the equivalence between i) and iii) in this result follows from the equivalence between i) and iv) in Theorem 1 by noting that $\mathrm{rank}(W^{\mathrm{T}} - I_q) = \mathrm{rank}(W - I_q)$ and $\mathrm{rank}\begin{bmatrix} WW^{\mathrm{T}} - I_q \\ W^{\mathrm{T}} - I_q \end{bmatrix} = \mathrm{rank}\begin{bmatrix} WW^{\mathrm{T}} - I_q & W - I_q \end{bmatrix}$. □

Based on Theorem 2 above and the proof of Theorem 5 in [67], we have the following convergence result for sequences of paracontracting matrix functions.

Theorem 3. Let Σ be a finite index set and $\mathscr{D} \subseteq \mathbb{R}^n$ be open and nonempty. Consider a matrix function $W : \Sigma \times \mathscr{D} \to \mathbb{R}^{q \times q}$. Assume that $W(p, \cdot)$ is continuous for every $p \in \Sigma$. Furthermore, assume that for every $z \in \Sigma \times \mathscr{D}$, $\|W(z)\| \leq 1$ and $\mathrm{rank}(W(z)W^{\mathrm{T}}(z) - I_q) = \mathrm{rank}(W(z) - I_q) = \mathrm{rank}\begin{bmatrix} W(z)W^{\mathrm{T}}(z) - I_q & W(z) - I_q \end{bmatrix}$. For any compact subset $\mathscr{M} \subset \Sigma \times \mathscr{D}$ and any sequence $\{z_k\}_{k=0}^{\infty} \subseteq \mathscr{M}$, if there exists a constant matrix $W_\infty \in \mathbb{R}^{q \times q}$ such that $\lim_{k \to \infty} \mathrm{rank}(W(z_k) - I_q) = \mathrm{rank}(W_\infty - I_q) = \lim_{k \to \infty} \mathrm{rank}\begin{bmatrix} W(z_k) - I_q \\ W_\infty - I_q \end{bmatrix}$, then the sequence $\{x_k\}_{k=0}^{\infty}$ defined by $x_{k+1} = W(z_k)x_k$ has a limit as $k \to \infty$.

Proof. First we claim that for every $z \in \mathscr{M}$, $W(z)$ is paracontracting. In fact, Theorem 2 has shown that for every $z \in \mathscr{M}$, $W(z)$ is paracontracting if and only if $\|W(z)\| \leq 1$ and $\mathrm{rank}(W(z)W^{\mathrm{T}}(z) - I_q) = \mathrm{rank}(W(z) - I_q) = \mathrm{rank}\begin{bmatrix} W(z)W^{\mathrm{T}}(z) - I_q & W(z) - I_q \end{bmatrix}$ for every $z \in \mathscr{M}$. Next, we claim that if $x \notin \ker(W(z) - I_q)$ for any $z \in \mathscr{M}$, then $\max_{z \in \mathscr{M}} \|W(z)x\| < \|x\|$. To see this, note that $\mathscr{M}$ is compact, $W(z)$ is paracontracting for every $z \in \mathscr{M}$, and $\|W(z)y\|$ is upper semicontinuous at $z \in \mathscr{M}$ for any $y \in \mathbb{R}^q$. Then it follows from Theorem 2.12 in [69, p. 44] that there exists $z^* \in \mathscr{M}$ such that $\max_{z \in \mathscr{M}} \|W(z)x\| = \|W(z^*)x\| < \|x\|$ provided that $W(z^*)x \neq x$.

Note that $x_{k+1} = (\prod_{i=0}^{k} W(z_{k-i}))x_0$ for every $k \geq 0$. To show that $\lim_{k\to\infty} x_k$ exists, it suffices to show that $\lim_{k\to\infty} \prod_{i=0}^{k} W(z_{k-i})$ exists. First, we claim that if $\lim_{k\to\infty} \mathrm{rank}(W(z_k)-I_q) = \mathrm{rank}(W_\infty - I_q) = \lim_{k\to\infty} \mathrm{rank} \begin{bmatrix} W(z_k) - I_q \\ W_\infty - I_q \end{bmatrix}$, then there exists a positive integer N such that $\ker(W(z_k) - I_q) = \ker(W_\infty - I_q)$ for all $k \geq N$. To see this assertion, it follows from the limit definition that for any given $\varepsilon \in (0, 1)$, there exists a positive integer $N_1 = N_1(\varepsilon)$ such that $|\mathrm{rank}(W(z_k) - I_q) - \mathrm{rank}(W_\infty - I_q)| < \varepsilon < 1$ for every $k \geq N_1$. Since $\mathrm{rank}(W(z_k) - I_q)$ and $\mathrm{rank}(W_\infty - I_q)$ are both nonnegative integers, the inequality $|\mathrm{rank}(W(z_k) - I_q) - \mathrm{rank}(W_\infty - I_q)| < 1$ holds if and only if $\mathrm{rank}(W(z_k) - I_q) = \mathrm{rank}(W_\infty - I_q)$ for every $k \geq N_1$. Similarly, there exists a positive integer $N_2 = N_2(\varepsilon)$ such that $\mathrm{rank} \begin{bmatrix} W(z_k) - I_q \\ W_\infty - I_q \end{bmatrix} = \mathrm{rank}(W_\infty - I_q)$ for every $k \geq N_2$. Let $N = \max\{N_1, N_2\}$. Then it follows that $\mathrm{rank}(W(z_k) - I_q) = \mathrm{rank}(W_\infty - I_q) = \mathrm{rank} \begin{bmatrix} W(z_k) - I_q \\ W_\infty - I_q \end{bmatrix}$ for every $k \geq N$. Now it follows from Lemma 5 that $\ker(W(z_k) - I_q) = \ker(W_\infty - I_q)$ for all $k \geq N$. Thus, $\ker(I_q - W(z_N)) = \ker(I_q - W(z_m))$ for every $m \geq N$. Note that it follows from paracontraction of $W(\cdot)$ that $\ker(I_q - W(z_N)) \neq \{\mathbf{0}_{q\times 1}\}$. Let $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_r\}$ be an orthonormal basis for $\ker(I_q - W(z_N))$ and let $R = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_r] \in \mathbb{R}^{q\times r}$. By definition of R, we have $(I_q - W(z_m))R = \mathbf{0}_{q\times r}$ for every $m \geq N$.

Next, it follows from Lemma 4 that $\ker(I_q - W^{\mathrm{T}}(z_m)) = \ker(I_q - W(z_m))$ for every $m \geq N$. Now $(I_q - W^{\mathrm{T}}(z_m))R = \mathbf{0}_{q\times r}$ for every $m \geq N$. Hence, $R^{\mathrm{T}}W(z_m) = (W^{\mathrm{T}}(z_m)R)^{\mathrm{T}} = R^{\mathrm{T}}$. Therefore, for every $i \geq N$, we have $(W(z_{i+1}) - RR^{\mathrm{T}})(W(z_i) -$

$RR^{\mathrm{T}}) = W(z_{i+1})W(z_i) - RR^{\mathrm{T}}$. By induction,

$$\begin{aligned}
&\prod_{i=0}^{m} W(z_{m-i}) - RR^{\mathrm{T}} \prod_{i=0}^{N-1} W(z_{N-1-i}) \\
&= \left(\prod_{i=0}^{m-N} W(z_{m-i}) - RR^{\mathrm{T}}\right) \prod_{i=0}^{N-1} W(z_{N-1-i}) \\
&= \left(\prod_{i=0}^{m-N} \Big(W(z_{m-i}) - RR^{\mathrm{T}}\Big)\right) \prod_{i=0}^{N-1} W(z_{N-1-i})
\end{aligned}$$

We show that $\lim_{m\to\infty} \prod_{i=0}^{m} W(z_{m-i}) = RR^{\mathrm{T}} \prod_{i=0}^{N-1} W(z_{N-1-i})$. To this end, it suffices to show that for any $x \in \mathbb{R}^q$, $\lim_{m\to\infty}(\prod_{i=0}^{m-N}(W(z_{m-i}) - RR^{\mathrm{T}}))(\prod_{i=0}^{N-1} W(z_{N-1-i}))x = \mathbf{0}_{q\times 1}$. Suppose, conversely, that there exists $\mathbf{x} \in \mathbb{R}^q$ such that $(\prod_{i=0}^{m-N}(W(z_{m-i}) - RR^{\mathrm{T}}))$ $(\prod_{i=0}^{N-1} W(z_{N-1-i}))\mathbf{x}$ does not converge to $\mathbf{0}_{q\times 1}$ as $m \to \infty$. First we claim that $W(z_k) - RR^{\mathrm{T}}$ is paracontracting for every $k \geq N$. To see this, let $\mathrm{span}\{\mathbf{e}_{r+1}, \mathbf{e}_{r+2}, \ldots \mathbf{e}_q\} = \mathrm{span}\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_r\}^{\perp}$ and let $Q = [\mathbf{e}_{r+1}, \mathbf{e}_{r+2}, \ldots \mathbf{e}_q] \in \mathbb{R}^{q\times(q-r)}$, where $S^{\perp}$ denotes the orthogonal complement of S. Next, let $x \notin \ker(W(z_k) - RR^{\mathrm{T}} - I_q)$, $k \geq N$. Then $x = Ry_1 + Qy_2$, where $y_1 \in \mathbb{R}^{r\times 1}$ and $y_2 \in \mathbb{R}^{(q-r)\times 1}$. Note that $W(z_k)R = R$ for all $k \geq N$. Then it follows that $(W(z_k) - RR^{\mathrm{T}})x = W(z_k)Qy_2$. Hence, $x \notin \ker(W(z_k) - RR^{\mathrm{T}} - I_q)$ if and only if $W(z_k)Qy_2 \neq Ry_1 + Qy_2$, or, equivalently $(W(z_k) - I_q)Qy_2 \neq Ry_1$. Since $W(z_k)$ is paracontracting, it follows from Proposition 3.2 of [65] that $W(z_k)$ is discrete-time semistable. Then it follows from $vi)$ of Proposition 11.10.2 of [68, p. 735] that $I_q - W(z_k)$ is group invertible. Hence, by Corollary 3.5.8 of [68, p. 191], $\mathrm{ran}(I_q - W(z_k)) \cap \ker(I_q - W(z_k)) = \{\mathbf{0}_{q\times 1}\}$. Since $(W(z_k) - I_q)Qy_2 \in \mathrm{ran}(I_q - W(z_k))$ and $Ry_1 \in \ker(I_q - W(z_k))$, it follows that $(W(z_k) - I_q)Qy_2 \neq Ry_1$ if and only if either $(W(z_k) - I_q)Qy_2 \neq \mathbf{0}_{q\times 1}$ or $Ry_1 \neq \mathbf{0}_{q\times 1}$.

If $(W(z_k) - I_q)Qy_2 \neq \mathbf{0}_{q\times 1}$, then it follows from the paracontraction of $W(z_k)$ that $\|W(z_k)Qy_2\|^2 < \|Qy_2\|^2 \leq \|Ry_1\|^2 + \|Qy_2\|^2 = \|Ry_1 + Qy_2\|^2$, i.e. $\|(W(z_k) - RR^{\mathrm{T}})x\| <$

$\|x\|$. Hence, $W(z_k)-RR^{\mathrm{T}}$ is paracontracting for every $k \geq N$. Alternatively, if $Ry_1 \neq \mathbf{0}_{q\times 1}$, then $\|Ry_1\| > 0$. Since $\|W(z_k)\| \leq 1$, it follows that $\|W(z_k)Qy_2\|^2 \leq \|Qy_2\|^2 < \|Ry_1\|^2 + \|Qy_2\|^2 = \|Ry_1 + Qy_2\|^2$, i.e., $\|(W(z_k) - RR^{\mathrm{T}})x\| < \|x\|$. Hence, $W(z_k) - RR^{\mathrm{T}}$ is paracontracting for every $k \geq N$.

Now we have

$$\begin{aligned} &\left\|\Big(\prod_{i=0}^{m-N}\Big(W(z_{m-i}) - RR^{\mathrm{T}}\Big)\Big)\Big(\prod_{i=0}^{N-1} W(z_{N-1-i})\Big)\mathbf{x}\right\| \\ &\leq \Big(\prod_{i=0}^{m-N}\|W(z_{m-i}) - RR^{\mathrm{T}}\|\Big)\Big(\prod_{i=0}^{N-1}\|W(z_{N-1-i})\|\Big)\|\mathbf{x}\| \leq \|\mathbf{x}\| \end{aligned}$$

Thus, $(\prod_{i=0}^{m-N}(W(z_{m-i}) - RR^{\mathrm{T}}))(\prod_{i=0}^{N-1} W(z_{N-1-i}))\mathbf{x}$ is a bounded sequence on $\mathbb{R}^q$ for every $m \geq N$. It follows from Bolzano-Weierstrass Theorem (Theorem 2.3 of [69, p. 27]) that there exists a subsequence $\{m_k\}_{k=0}^{\infty} \subseteq \{N, N+1, N+2, \ldots\}$ such that

$$\lim_{k\to\infty}\Big(\prod_{i=0}^{m_k-N}(W(z_{m_k-i}) - RR^{\mathrm{T}})\Big)\Big(\prod_{i=0}^{N-1} W(z_{N-1-i})\Big)\mathbf{x} = \mathbf{w} \neq \mathbf{0}_{q\times 1}.$$

Let $(\prod_{i=0}^{m_k-N}(W(z_{m_k-i})-RR^{\mathrm{T}}))(\prod_{i=0}^{N-1} W(z_{N-1-i}))\mathbf{x} = s_k\mathbf{w}+\mathbf{d}_k$, where $\mathbf{d}_k \in \mathrm{span}\{\mathbf{w}\}^{\perp}$. Note that $\mathbf{w}^{\mathrm{T}}(\prod_{i=0}^{m_k-N}(W(z_{m_k-i})-RR^{\mathrm{T}}))(\prod_{i=0}^{N-1} W(z_{N-1-i}))\mathbf{x} = s_k\|\mathbf{w}\|^2$ and $\mathbf{d}_k^{\mathrm{T}}(\prod_{i=0}^{m_k-N}$ $(W(z_{m_k-i}) - RR^{\mathrm{T}}))(\prod_{i=0}^{N-1} W(z_{N-1-i}))\mathbf{x} = \|\mathbf{d}_k\|^2$. Taking the limit on both sides of these two equations yields $\lim_{k\to\infty} s_k = 0$ and $\lim_{k\to\infty} \mathbf{d}_k = \mathbf{0}_{q\times 1}$.

Next, we claim that $\ker(I_q - W(z_i) + RR^{\mathrm{T}}) = \{\mathbf{0}_{q\times 1}\}$ for every $i \geq m_k$ and every $k = 0, 1, 2, \ldots$. If $x \in \ker(I_q - W(z_i) + RR^{\mathrm{T}})$ for every $i \geq m_k$, then $W(z_i)x = (I_q + RR^{\mathrm{T}})x$ for every $i \geq m_k$. Since $\|W(z_i)x\|^2 = \|x\|^2 + 2\|R^{\mathrm{T}}x\|^2 + \|RR^{\mathrm{T}}x\|^2$ and $W(z_i)$ is paracontracting, it follows that $R^{\mathrm{T}}x = \mathbf{0}_{r\times 1}$. Hence, $W(z_i)x = x$, which implies that $x \in \ker(I_q - W(z_i))$ for every $i \geq m_k$. Since $m_k \geq N$ and the column vectors of R form an orthonormal basis for $\ker(I_q - W(z_N)) = \ker(I_q - W(z_i))$ for every $i \geq m_k$, it follows that there exists $y \in \mathbb{R}^r$ such that $x = Ry$. Since $R^{\mathrm{T}}x = \mathbf{0}_{r\times 1}$, it follows that $R^{\mathrm{T}}Ry = \mathbf{0}_{r\times 1}$, and hence, $Ry = \mathbf{0}_{q\times 1}$. Finally, $x = Ry = \mathbf{0}_{q\times 1}$.

Since $\mathbf{w} \neq \mathbf{0}_{q\times 1}$, it follows that $\mathbf{w} \notin \ker(I_q - W(z_i) + RR^{\mathrm{T}})$ for every $i \geq m_k$. Hence, for every $k = 0, 1, 2, \ldots$, there exists $j > m_k$ such that $\mathbf{w} \notin \ker(I_q - W(z_j) + RR^{\mathrm{T}})$. Let $n_k = \min\{j : j > m_k, \mathbf{w} \notin \ker(I_q - W(z_j) + RR^{\mathrm{T}})\}$. Then it follows that

$$\begin{aligned}
&\left(\prod_{i=0}^{n_k-N}\left(W(z_{n_k-i}) - RR^{\mathrm{T}}\right)\right)\left(\prod_{i=0}^{N-1} W(z_{N-1-i})\right)\mathbf{x} \\
&= \left(\prod_{i=0}^{n_k-m_k-1}\left(W(z_{n_k-i}) - RR^{\mathrm{T}}\right)\right)\left(\prod_{i=0}^{m_k-N}\left(W(z_{m_k-i}) - RR^{\mathrm{T}}\right)\right) \\
&\times \left(\prod_{i=0}^{N-1} W(z_{N-1-i})\right)\mathbf{x} \\
&= s_k\left(W(z_{n_k}) - RR^{\mathrm{T}}\right)\mathbf{w} + \left(\prod_{i=0}^{n_k-m_k-1}\left(W(z_{n_k-i}) - RR^{\mathrm{T}}\right)\right)\mathbf{d}_k.
\end{aligned}$$

Taking the norm on both sides of this equation yields $\|(\prod_{i=0}^{n_k-N}(W(z_{n_k-i}) - RR^{\mathrm{T}}))(\prod_{i=0}^{N-1} W(z_{N-1-i}))\mathbf{x}\| \leq |s_k| \max_{z_{n_k}\in\mathscr{M}} \|(W(z_{n_k}) - RR^{\mathrm{T}})\mathbf{w}\| + \|\mathbf{d}_k\|$.

We claim that if $x \notin \ker(W(z_{n_k}) - RR^{\mathrm{T}} - I_q)$, then $\max_{z_{n_k}\in\mathscr{M}} \|(W(z_{n_k}) - RR^{\mathrm{T}})x\| < \|x\|$. Note that we have shown that $x \notin \ker(W(z_{n_k}) - RR^{\mathrm{T}} - I_q)$ if and only if $(W(z_{n_k}) - I_q)Qy_2 \neq \mathbf{0}_{q\times 1}$ or $Ry_1 \neq \mathbf{0}_{q\times 1}$, where $x = Ry_1 + Qy_2$. If $(W(z_{n_k}) - I_q)Qy_2 \neq \mathbf{0}_{q\times 1}$, then it follows from the paracontraction of $W(z_k)$ and compactness of $\mathscr{M}$ that $\max_{n_k\in\mathscr{M}} \|W(z_{n_k})Qy_2\| \leq \max_{z\in\mathscr{M}} \|W(z)Qy_2\| = \|W(z^*)Qy_2\| < \|Qy_2\| \leq (\|Ry_1\|^2 + \|Qy_2\|^2)^{1/2} = \|Ry_1 + Qy_2\|$ for some $z^* \in \mathscr{M}$, i.e., $\max_{n_k\in\mathscr{M}} \|(W(z_{n_k}) - RR^{\mathrm{T}})x\| < \|x\|$. Alternatively, if $Ry_1 \neq \mathbf{0}_{q\times 1}$, then $\|Ry_1\| > 0$. Note that we have shown that if $x \notin \ker(W(z) - I_q)$ for any $z \in \mathscr{M}$, then $\max_{z\in\mathscr{M}} \|W(z)x\| < \|x\|$. Thus, it follows that $\max_{z_{n_k}\in\mathscr{M}} \|W(z_{n_k})Qy_2\| \leq \|Qy_2\| < (\|Ry_1\|^2 + \|Qy_2\|^2)^{1/2} = \|Ry_1 + Qy_2\|$, i.e., $\max_{z_{n_k}\in\mathscr{M}} \|(W(z_{n_k}) - RR^{\mathrm{T}})x\| < \|x\|$.

Since $\mathbf{w} \notin \ker(I_q - W(z_{n_k}) + RR^{\mathrm{T}})$, it follows that $\max_{z_{n_k}\in\mathscr{M}} \|(W(z_{n_k}) - RR^{\mathrm{T}})\mathbf{w}\| <$

$\|\mathbf{w}\|$. Hence,

$$\begin{aligned}&\limsup_{k\to\infty}\left\|\left(\prod_{i=0}^{n_k-N}\left(W(z_{n_k-i})-RR^{\mathrm{T}}\right)\right)\left(\prod_{i=0}^{N-1}W(z_{N-1-i})\right)\mathbf{x}\right\|\\&\leq\limsup_{k\to\infty}|s_k|\max_{z_{n_k}\in\mathscr{M}}\|(W(z_{n_k})-RR^{\mathrm{T}})\mathbf{w}\|+\limsup_{k\to\infty}\|\mathbf{d}_k\|\\&\leq\max_{z_{n_k}\in\mathscr{M}}\|(W(z_{n_k})-RR^{\mathrm{T}})\mathbf{w}\|<\|\mathbf{w}\|.\end{aligned}$$

Note that $\|(\prod_{i=0}^{n_k-N}(W(z_{n_k-i})-RR^{\mathrm{T}}))(\prod_{i=0}^{N-1}W(z_{N-1-i}))\mathbf{x}\|$ is monotonically decreasing in terms of k. Then it follows that $\lim_{k\to\infty}\|(\prod_{i=0}^{n_k-N}(W(z_{n_k-i})-RR^{\mathrm{T}}))(\prod_{i=0}^{N-1}W(z_{N-1-i}))\mathbf{x}\|=\|\mathbf{w}\|$. Consequently, for any subsequence $\{s_k\}_{k=0}^{\infty}$ of $\{n_k\}_{k=0}^{\infty}$, $\lim_{k\to\infty}\|(\prod_{i=0}^{s_k-N}(W(z_{s_k-i})-RR^{\mathrm{T}}))(\prod_{i=0}^{N-1}W(z_{N-1-i}))\mathbf{x}\|=\|\mathbf{w}\|$. On the other hand, it follows from $\limsup_{k\to\infty}\|(\prod_{i=0}^{n_k-N}(W(z_{n_k-i})-RR^{\mathrm{T}}))(\prod_{i=0}^{N-1}W(z_{N-1-i}))\mathbf{x}\|<\|\mathbf{w}\|$ that there exists a subsequence $\{q_k\}_{k=0}^{\infty}$ of $\{n_k\}_{k=0}^{\infty}$ such that

$$\lim_{k\to\infty}\left\|\left(\prod_{i=0}^{q_k-N}\left(W(z_{q_k-i})-RR^{\mathrm{T}}\right)\right)\left(\prod_{i=0}^{N-1}W(z_{N-1-i})\right)\mathbf{x}\right\|<\|\mathbf{w}\|.$$

This is a contradiction. Therefore, $\lim_{m\to\infty}\prod_{i=0}^{m}W(z_{m-i})=RR^{\mathrm{T}}\prod_{i=0}^{N-1}W(z_{N-1-i})$. $\square$

3.2 Cooperative Learning Consensus

In this section, we present some theoretic results on the convergence of the cooperative learning consensus protocol (3.4)–(3.8) with (3.9) by means of matrix paracontraction techniques and nonnegative matrix tools. In particular, we view the proposed cooperative learning consensus protocol with (3.9) as a discrete-time, linear time-varying system and then use Theorems 1–3 in Section 3.1.2 and Theorem 1 in [64] to rigorously show its convergence under two different sufficient conditions.

First, note that with (3.9), we have

$$\arg\min_{\boldsymbol{\theta}\in\mathbb{R}^n}\sum_{j\in\mathcal{N}_t^k}[\boldsymbol{\theta}_j(t)-\boldsymbol{\theta}]^{\mathrm{T}}\Phi_{k,j}(\boldsymbol{\theta}_j(t)-\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^n}\sum_{j\in\mathcal{N}_t^k}\|\boldsymbol{\theta}_j(t)-\boldsymbol{\theta}\|^2$$
$$= \frac{1}{|\mathcal{N}_t^k|}\sum_{j\in\mathcal{N}_t^k}\boldsymbol{\theta}_j(t), \quad k=1,\ldots,q. \tag{3.13}$$

Hence, substituting (3.13) into (3.7) yields

$$\boldsymbol{\theta}_k(t+1) = \boldsymbol{\theta}_k(t) + \frac{f_k(t)}{|\mathcal{N}_t^k|}\sum_{j\in\mathcal{N}_t^k}[\boldsymbol{\theta}_j(t)-\boldsymbol{\theta}_k(t)] + f_k(t)\mu_k(t)[\boldsymbol{\theta}_{q,\min}(t)-\boldsymbol{\theta}_k(t)], \quad k=1,\ldots,q \tag{3.14}$$

where we used the fact that $\boldsymbol{\theta}_{q+k,\min}(t) = \boldsymbol{\theta}_{q,\min}(t) = \arg\min_{1\le i\le q} F(\boldsymbol{\theta}_i(t))$.

Next, to convert the above forms into (3.10), we define a series of matrices $R_k \in \mathbb{R}^{q\times q}$, $U_k \in \mathbb{R}^{q\times q}$, and $\boldsymbol{A}_k^{[j]} \in \mathbb{R}^{q\times q}$, $j=1,\ldots,q$, $k=0,1,2,\ldots$, throughout the chapter as follows: $R_k = \mathrm{diag}(f_k^1,\ldots,f_k^q)$, $U_k = \mathrm{diag}(\mu_k^1,\ldots,\mu_k^q)$, and $\boldsymbol{A}_k^{[j]}$ is given by

$$\boldsymbol{A}_k^{[j]} = R_kU_k(\mathbf{1}_{q\times 1}\otimes E_{1\times q}^{[j]} - I_q) - R_kD_k^{-1}L_k \tag{3.15}$$

where $\otimes$ denotes the Kronecker product, $\mathbf{1}_{m\times l}$ denotes the m-by-l matrix whose entries are all ones, and $E_{l\times lq}^{[j]} \in \mathbb{R}^{l\times lq}$ denotes a block-matrix whose jth block-column is I_l and the rest of the block-elements are all zero matrices, i.e., $E_{l\times lq}^{[j]} = [\mathbf{0}_{l\times l},\ldots,\mathbf{0}_{l\times l},I_l,\mathbf{0}_{l\times l},\ldots,\mathbf{0}_{l\times l}]$, where $\mathbf{0}_{m\times l}$ denotes the m-by-l matrix whose entries are all zeros, $j=1,\ldots,q$.

Using these notations, the cooperative learning consensus protocol (3.14) can be written as (3.10) with $W(t) = I_{nq} - (R_tD_t^{-1}L_t)\otimes I_n + ((R_tU_t)\otimes I_n)(\mathbf{1}_{q\times 1}\otimes E_{n\times nq}^{[j_t]} - I_{nq}) = I_{nq} + \boldsymbol{A}_t^{[j_t]}\otimes I_n = (I_q + \boldsymbol{A}_t^{[j_t]})\otimes I_n$, $j_t \in \{1,\ldots,q\}$, $t\in\overline{\mathbb{Z}}_+$.

3.2.1 Quadratic Monotone Convergence via Matrix Paracontraction

In this subsection, we focus on using Theorems 1–3 in Section 3.1.2 to study the monotone convergence of the proposed cooperative learning consensus protocol with respect to a quadratic function of its state. This type of the convergence is named after *quadratic monotone convergence* (QMC).

Lemma 6. Consider $\boldsymbol{A}_k^{[j]}$ defined by (3.15), $j = 1, \ldots, q$, $k = 0, 1, 2, \ldots$. Then for every $j = 1, \ldots, q$, $k = 0, 1, 2, \ldots$, $\operatorname{rank}(\boldsymbol{A}_k^{[j]}) \leq q - 1$ and $\operatorname{span}\{\mathbf{1}_{q\times 1}\} \subseteq \ker(\boldsymbol{A}_k^{[j]})$. If in addition $\operatorname{rank}(\boldsymbol{A}_k^{[j]}) = q - 1$, then $\ker(\boldsymbol{A}_k^{[j]}) = \operatorname{span}\{\mathbf{1}_{q\times 1}\}$.

Proof. First, note that $(\mathbf{1}_{q\times 1} \otimes E_{1\times q}^{[j]} - I_q)\mathbf{1}_{q\times 1} = \mathbf{0}_{q\times 1}$ and $L_k\mathbf{1}_{q\times 1} = \mathbf{0}_{q\times 1}$. Hence, $\boldsymbol{A}_k^{[j]}\mathbf{1}_{q\times 1} = \mathbf{0}_{q\times 1}$. This implies that $\operatorname{rank}(\boldsymbol{A}_k^{[j]}) \leq q - 1$ and $\operatorname{span}\{\mathbf{1}_{q\times 1}\} \subseteq \ker(\boldsymbol{A}_k^{[j]})$. Finally, if $\operatorname{rank}(\boldsymbol{A}_k^{[j]}) = q-1$, then $\dim\ker(\boldsymbol{A}_k^{[j]}) = 1$, and hence, $\ker(\boldsymbol{A}_k^{[j]}) = \operatorname{span}\{\mathbf{1}_{q\times 1}\}$, where $\dim S$ denotes the dimension of a subspace S. □

It follows from Lemma 6 that 0 is an eigenvalue of $\boldsymbol{A}_k^{[j]}$ for every $j = 1, \ldots, q$ and every $k \in \overline{\mathbb{Z}}_+$. Hence, 1 is an eigenvalue of $I_q + \boldsymbol{A}_k^{[j]}$. The next result gives the *exact* value to the 2-norm or maximum singular value for $I_q + \boldsymbol{A}_k^{[j]}$ under certain conditions.

Lemma 7. Consider the matrices $\boldsymbol{A}_k^{[j]}$ defined by (3.15), $j = 1, \ldots, q$, and $k = 0, 1, 2, \ldots$. Assume that for every $j = 1, \ldots, q$ and every $k = 0, 1, 2, \ldots$, the following linear matrix inequality holds:

$$\begin{bmatrix} -\boldsymbol{A}_k^{[j]} - (\boldsymbol{A}_k^{[j]})^{\mathrm{T}} & \boldsymbol{A}_k^{[j]} \\ (\boldsymbol{A}_k^{[j]})^{\mathrm{T}} & I_q \end{bmatrix} \geq 0. \tag{3.16}$$

Then for every $j = 1, \ldots, q$ and every $k = 0, 1, 2, \ldots$, $\|I_q + \boldsymbol{A}_k^{[j]}\| = 1$.

Proof. First, it follows from Lemma 6 that 0 is an eigenvalue of $\boldsymbol{A}_k^{[j]}$ and hence, 1 is an eigenvalue of $I_q+\boldsymbol{A}_k^{[j]}$, which implies that $\|I_q+\boldsymbol{A}_k^{[j]}\|\geq 1$. Thus, to prove $\|I_q+\boldsymbol{A}_k^{[j]}\|=1$, it suffices to show that $\|I_q+\boldsymbol{A}_k^{[j]}\|\leq 1$. Note that by Lemma 1, $\|I_q+\boldsymbol{A}_k^{[j]}\|\leq 1$ if and only if $(I_q+\boldsymbol{A}_k^{[j]})(I_q+\boldsymbol{A}_k^{[j]})^{\mathrm{T}}\leq I_q$, or equivalently, $\boldsymbol{A}_k^{[j]}(\boldsymbol{A}_k^{[j]})^{\mathrm{T}}+\boldsymbol{A}_k^{[j]}+(\boldsymbol{A}_k^{[j]})^{\mathrm{T}}$ is negative-semidefinite, which is equivalent to (3.16) by the Schur complement. □

Remark 2. Using the Schur complement, it can be seen that (3.16) is also equivalent to

$$\begin{aligned}&[R_kU_k(\mathbf{1}_{q\times 1}\otimes E_{1\times q}^{[j]}-I_q)-R_kD_k^{-1}L_k][R_kU_k(\mathbf{1}_{q\times 1}\otimes E_{1\times q}^{[j]}-I_q)-R_kD_k^{-1}L_k]^{\mathrm{T}}\\&\leq R_kD_k^{-1}L_k+L_k^{\mathrm{T}}D_k^{-1}R_k+R_kU_k(I_q-\mathbf{1}_{q\times 1}\otimes E_{1\times q}^{[j]})+(I_q-\mathbf{1}_{q\times 1}\otimes E_{1\times q}^{[j]})^{\mathrm{T}}U_kR_k.\end{aligned}$$

♦

Remark 3. One could also use the matrix $(\boldsymbol{A}_k^{[j]})^{\mathrm{T}}\boldsymbol{A}_k^{[j]}+(\boldsymbol{A}_k^{[j]})^{\mathrm{T}}+\boldsymbol{A}_k^{[j]}$ to derive a similar condition as (3.16) to guarantee its negative-semidefiniteness. ♦

Now we have the QMC result for the proposed cooperative learning consensus protocol.

Theorem 4. Consider (3.4)–(3.8) with (3.9). Assume that Assumptions 1 and 2 hold. Furthermore, assume that for every $k\in\overline{\mathbb{Z}}_+$ and every $j=1,\ldots,q$, the following conditions hold:

Q1) (3.16) holds.

Q2) $\mathrm{rank}(\boldsymbol{A}_k^{[j]})=\mathrm{rank}[\boldsymbol{A}_k^{[j]}(\boldsymbol{A}_k^{[j]})^{\mathrm{T}}+\boldsymbol{A}_k^{[j]}+(\boldsymbol{A}_k^{[j]})^{\mathrm{T}}]$
$=\mathrm{rank}\begin{bmatrix}\boldsymbol{A}_k^{[j]}(\boldsymbol{A}_k^{[j]})^{\mathrm{T}}+(\boldsymbol{A}_k^{[j]})^{\mathrm{T}} & \boldsymbol{A}_k^{[j]}\end{bmatrix}=q-1.$

Then $\lim_{k\to\infty}\boldsymbol{\theta}_i(k)=\boldsymbol{\theta}^{\dagger}$ for every $\boldsymbol{\theta}_i(0)\in\mathbb{R}^n$ and every $i=1,\ldots,q$, where $\boldsymbol{\theta}^{\dagger}=\lim_{k\to\infty}\arg\min_{1\leq i\leq q}F(\boldsymbol{\theta}_i(k))$. Furthermore, this convergence is QMC, that is, $V(\boldsymbol{\theta}_1(t+1),\ldots,\boldsymbol{\theta}_q(t+1))\leq V(\boldsymbol{\theta}_1(t),\ldots,\boldsymbol{\theta}_q(t))$ for all $k\in\overline{\mathbb{Z}}_+$, where $V(\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_q)=\sum_{i=1}^{q}\|\boldsymbol{\theta}_i\|^2$.

Proof. Let $X(t) = [\boldsymbol{\theta}_1^{\mathrm{T}}(t), \ldots, \boldsymbol{\theta}_q^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{nq}$. Note that (3.4)–(3.8) with (3.9) can be rewritten as the compact form (3.10), where $W(t) = I_{nq} + \boldsymbol{A}_t^{[j_t]} \otimes I_n = (I_q + \boldsymbol{A}_t^{[j_t]}) \otimes I_n$, $j_t \in \{1, \ldots, q\}$, $t \in \overline{\mathbb{Z}}_+$. Then it follows from Assumption Q1) and Lemma 7 that $\|I_{nq} + \boldsymbol{A}_k^{[j]} \otimes I_n\| = 1$. Note that it follows from the matrix column elementary operation that $\mathrm{rank}\begin{bmatrix}\boldsymbol{A}_k^{[j]}(\boldsymbol{A}_k^{[j]})^{\mathrm{T}} + (\boldsymbol{A}_k^{[j]})^{\mathrm{T}} + \boldsymbol{A}_k^{[j]} & \boldsymbol{A}_k^{[j]}\end{bmatrix} = \mathrm{rank}\begin{bmatrix}\boldsymbol{A}_k^{[j]}(\boldsymbol{A}_k^{[j]})^{\mathrm{T}} + (\boldsymbol{A}_k^{[j]})^{\mathrm{T}} & \boldsymbol{A}_k^{[j]}\end{bmatrix}$. Hence, it follows from Assumption Q2) that $\mathrm{rank}[\boldsymbol{A}_k^{[j]}(\boldsymbol{A}_k^{[j]})^{\mathrm{T}} + \boldsymbol{A}_k^{[j]} + (\boldsymbol{A}_k^{[j]})^{\mathrm{T}}]$ $= \mathrm{rank}\begin{bmatrix}\boldsymbol{A}_k^{[j]}(\boldsymbol{A}_k^{[j]})^{\mathrm{T}} + (\boldsymbol{A}_k^{[j]})^{\mathrm{T}} + \boldsymbol{A}_k^{[j]} & \boldsymbol{A}_k^{[j]}\end{bmatrix} = \mathrm{rank}(\boldsymbol{A}_k^{[j]})$, that is,

$$\begin{aligned}
&\mathrm{rank}[(\boldsymbol{A}_k^{[j]}(\boldsymbol{A}_k^{[j]})^{\mathrm{T}} + \boldsymbol{A}_k^{[j]} + (\boldsymbol{A}_k^{[j]})^{\mathrm{T}}) \otimes I_n] \\
&\quad = \mathrm{rank}\begin{bmatrix}(\boldsymbol{A}_k^{[j]}(\boldsymbol{A}_k^{[j]})^{\mathrm{T}} + (\boldsymbol{A}_k^{[j]})^{\mathrm{T}} + \boldsymbol{A}_k^{[j]}) \otimes I_n & \boldsymbol{A}_k^{[j]} \otimes I_n\end{bmatrix} \\
&\quad = \mathrm{rank}(\boldsymbol{A}_k^{[j]} \otimes I_n).
\end{aligned}$$

By $iii)$ of Theorem 2, $I_{nq} + \boldsymbol{A}_k^{[j]} \otimes I_n$ is paracontracting. Next, it follows from Assumption Q2) and Lemma 6 that $\ker(\boldsymbol{A}_k^{[j]}) = \mathrm{span}\{\mathbf{1}_{q\times 1}\}$ for any $j = 1, \ldots, q$ and every $k \geq 0$. Hence, $\ker(\boldsymbol{A}_k^{[j]}) = \ker(\boldsymbol{A}_0^{[1]})$ for any $j = 1, \ldots, q$ and every $k \geq 0$. By Lemma 5, we have $\mathrm{rank}(\boldsymbol{A}_k^{[j]}) = \mathrm{rank}(\boldsymbol{A}_0^{[1]}) = \mathrm{rank}\begin{bmatrix}\boldsymbol{A}_k^{[j]} \\ \boldsymbol{A}_0^{[1]}\end{bmatrix}$ for any $j = 1, \ldots, q$ and every $k \geq 0$. Let $W_\infty = I_{nq} + \boldsymbol{A}_0^{[1]} \otimes I_n$. Then it follows that $\mathrm{rank}(W(t) - I_{nq}) = \mathrm{rank}(W_\infty - I_{nq}) = \mathrm{rank}\begin{bmatrix}W(t) - I_{nq} \\ W_\infty - I_{nq}\end{bmatrix}$ for every $t \in \overline{\mathbb{Z}}_+$. Now, it follows from Theorem 3 that $\lim_{t\to\infty} X(t)$ exists. The consensus of the limiting state follows directly from Assumption Q2) and Lemma 6 on $\ker(\boldsymbol{A}_k^{[j]})$. Finally, since $\lim_{k\to\infty} \boldsymbol{\theta}_i(k) = \boldsymbol{\theta}^\dagger$, it follows from (3.14) that $\lim_{k\to\infty} \boldsymbol{\theta}_{q,\min}(k) = \lim_{k\to\infty} \boldsymbol{\theta}_i(k) = \boldsymbol{\theta}^\dagger$, which means that $\boldsymbol{\theta}^\dagger = \lim_{k\to\infty} \arg\min_{1\leq i\leq q} F(\boldsymbol{\theta}_i(k))$. The monotone property $\sum_{i=1}^q \|\boldsymbol{\theta}_i(k+1)\|^2 \leq \sum_{i=1}^q \|\boldsymbol{\theta}_i(k)\|^2$ follows from the fact that $\|W(t)\| = 1$ and $\|X(t+1)\| = \|W(t)X(t)\| \leq \|W(t)\|\|X(t)\| = \|X(t)\|$. □

3.2.2 Orthant Invariant Convergence via Nonnegative Matrices

In this subsection, we use Theorem 1 in [64] to study the invariant convergence of the cooperative learning consensus protocol (3.14) within the nonnegative or nonpositive orthant. This type of the convergence is named after *orthant invariant convergence* (OIC). Recall from [57] that a matrix $A \in \mathbb{R}^{n\times n}$ is called *nonnegative* if all the entries of A are nonnegative.

Lemma 8. Consider $\boldsymbol{A}_k^{[j]}$ defined by (3.15), $j = 1, \ldots, q$, $k = 0, 1, 2, \ldots$. If for every $i = 1, \ldots, q$ and $k = 0, 1, 2, \ldots$, $f_k^i(1 + \mu_k^i) \leq 1$, then $I_q + \boldsymbol{A}_k^{[j]}$ is nonnegative for every $j = 1, \ldots, q$ and $k = 0, 1, 2, \ldots$.

Proof. It follows from (3.14) that

$$\boldsymbol{\theta}_k(t+1) = [1 - f_k(t) - f_k(t)\mu_k(t)]\boldsymbol{\theta}_k(t) + \frac{f_k(t)}{|\mathcal{N}_t^k|}\sum_{j\in\mathcal{N}_t^k}\boldsymbol{\theta}_j(t) + f_k(t)\mu_k(t)\boldsymbol{\theta}_{q+k,\min}(t). \tag{3.17}$$

Hence, if $1 - f_k(t) - f_k(t)\mu_k(t) \geq 0$, then $I_q + \boldsymbol{A}_k^{[j]}$ is nonnegative for every $j = 1, \ldots, q$ and $k = 0, 1, 2, \ldots$. □

The following result is due to Theorem 1 of [64].

Lemma 9 ([64]). Consider (3.10) where $X(t) \in \mathbb{R}^m$ and $W(t) \in \mathbb{R}^{m\times m}$, $t \in \overline{\mathbb{Z}}_+$. Assume that $W(t)$ is nonnegative and $W(t)\mathbf{1}_{m\times 1} = \mathbf{1}_{m\times 1}$ for every $t \in \overline{\mathbb{Z}}_+$. Define $\mathscr{E}(t)$ as the set of ordered pairs (j, i) such that the (i, j)th element of $W(t)$ is positive, $i, j = 1, \ldots, m$, $i \neq j$. Let $\mathscr{E}$ be the set of (i, j) such that $(i, j) \in \mathscr{E}(t)$ for infinitely many $t \in \overline{\mathbb{Z}}_+$, $i, j = 1, \ldots, m$, $i \neq j$. Furthermore, the following additional assumptions hold:

i) The set $\mathscr{B} \subseteq \mathscr{N} = \{1, \ldots, m\}$ is nonempty.

ii) There exists $c > 0$ such that if $(j, i) \in \mathscr{E}(t)$, then the (i, j)th element of $W(t)$ is bounded below by c.

iii) There exists a positive integer T such that for every $t \in \overline{\mathbb{Z}}_+$, $\bigcup_{i=1}^{T} \mathscr{E}(t+i) = \mathscr{E}$.

iv) The digraph formed by $(\mathscr{N}, \mathscr{E})$ contains a directed communication path from every $i \in \mathscr{B}$ to every $j \in \mathscr{N}$.

Then there exist $\phi_i \geq 0$, $i = 1, \ldots, m$, such that $\lim_{t\to\infty} X(t) = (\mathbf{1}_{m\times 1}[\phi_1, \ldots, \phi_m])X(0)$. Furthermore, if $i \in \mathscr{B}$, then $\phi_i > 0$. Finally, this convergence is OIC, that is, $X(t) \in \overline{\mathbb{R}}_+^m = \{[x_1, \ldots, x_m]^{\mathrm{T}} \in \mathbb{R}^m : x_i \geq 0, i = 1, \ldots, m\}$ if $X(0) \in \overline{\mathbb{R}}_+^m$ or $X(t) \in \overline{\mathbb{R}}_-^m = \{[x_1, \ldots, x_m]^{\mathrm{T}} \in \mathbb{R}^m : x_i \leq 0, i = 1, \ldots, m\}$ if $X(0) \in \overline{\mathbb{R}}_-^m$.

Using the above result, we have the following OIC result for the proposed cooperative learning consensus protocol.

Theorem 5. Consider (3.4)–(3.8) with (3.9). Assume that Assumptions 1 and 2 hold with the digraph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$, where $\mathcal{V} = \{1, \ldots, q\}$. Let $\mathcal{E}$ be the set of (i, j) such that $(i, j) \in \mathcal{E}_t$ for infinitely many $t \in \overline{\mathbb{Z}}_+$, $i, j = 1, \ldots, q$, $i \neq j$. Furthermore, assume that for every $k \in \overline{\mathbb{Z}}_+$ and every $j = 1, \ldots, q$, the following conditions hold:

O1) $f_k^i(1 + \mu_k^i) \leq 1$.

O2) There exists a positive integer T such that for every $k \in \overline{\mathbb{Z}}_+$, $\bigcup_{i=1}^{T} \mathcal{E}_{k+i} = \mathcal{E}$.

Then $\lim_{k\to\infty} \boldsymbol{\theta}_i(k) = \boldsymbol{\theta}^\dagger$ for every $\boldsymbol{\theta}_i(0) \in \mathbb{R}^n$ and every $i = 1, \ldots, q$, where $\boldsymbol{\theta}^\dagger = \lim_{k\to\infty} \arg\min_{1\leq i\leq q} F(\boldsymbol{\theta}_i(k))$. Furthermore, this convergence is OIC.

Proof. To prove the conclusion, it suffices to verify all of the assumptions in Lemma 9. First, it follows from Assumption O1) and Lemma 8 that $I_q + \boldsymbol{A}_k^{[j]}$ is nonnegative for every $j = 1, \ldots, q$ and $k = 0, 1, 2, \ldots$. Furthermore, it follows from Lemma 6 that $(I_q + \boldsymbol{A}_k^{[j]})\mathbf{1}_{q\times 1} = \mathbf{1}_{q\times 1}$.

Next, by Assumption 1, *i*) and *iv*) of Lemma 9 hold with $\mathscr{B} = \mathscr{N} = \{1, \ldots, q\} = \mathcal{V}$. For $\boldsymbol{A}_k^{[j]}$ given by (3.15), it follows from (3.17) that all of the off-diagonal elements of

$I_q + \boldsymbol{A}_k^{[j]}$ are chosen among $f_k^i/|\mathcal{N}_k^i|$, $f_k^i/|\mathcal{N}_k^i| + f_k^i\mu_k^i$, and $f_k^i\mu_k^i$. Since $|\mathcal{N}_k^i| \leq q-1$, $f_k^i \geq f_{\min} > 0$, and $\mu_k^i \geq \mu_{\min} > 0$, it follows that $f_k^i/|\mathcal{N}_k^i| \geq f_{\min}/(q-1)$, $f_k^i/|\mathcal{N}_k^i| + f_k^i\mu_k^i \geq f_{\min}/(q-1) + f_{\min}\mu_{\min}$, and $f_k^i\mu_k^i \geq f_{\min}\mu_{\min}$. Then ii) of Lemma 9 holds with $\mathscr{E}(t) = \mathcal{E}_t$ and $c = \min\{f_{\min}/(q-1), f_{\min}\mu_{\min}\} > 0$. Finally, iii) of Lemma 9 follows directly from Assumption O2) with $\mathscr{E} = \mathcal{E}$. Note that for (3.10) with $X(t) = [\boldsymbol{\theta}_1^{\mathrm{T}}(t), \ldots, \boldsymbol{\theta}_q^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{nq}$, $X(t) = W(t)W(t-1)\cdots W(0)X(0) = [(I_q + \boldsymbol{A}_t^{[j_t]}) \otimes I_n][(I_q + \boldsymbol{A}_{t-1}^{[j_{t-1}]}) \otimes I_n] \cdots [(I_q + \boldsymbol{A}_0^{[j_0]}) \otimes I_n]X(0) = ([(I_q + \boldsymbol{A}_t^{[j_t]})(I_q + \boldsymbol{A}_{t-1}^{[j_{t-1}]}) \cdots (I_q + \boldsymbol{A}_0^{[j_0]})] \otimes I_n)X(0)$. Now it follows from Lemma 9 that there exists $\phi_i > 0$, $i = 1, \ldots, q$, such that $\lim_{t\to\infty} X(t) = ((\mathbf{1}_{q\times 1}[\phi_1, \ldots, \phi_q]) \otimes I_n)X(0)$, and hence, $\lim_{k\to\infty} \boldsymbol{\theta}_i(k) = \boldsymbol{\theta}^\dagger$ for every $\boldsymbol{\theta}_i(0) \in \mathbb{R}^n$ and every $i = 1, \ldots, q$, where $\boldsymbol{\theta}^\dagger = \lim_{k\to\infty} \arg\min_{1\leq i\leq q} F(\boldsymbol{\theta}_i(k))$, and the convergence is OIC. □

Remark 4. Both Theorems 4 and 5 give sufficient conditions to guarantee the convergence of (3.4)–(3.8) with (3.9). While Theorem 4 does not require $W(t)$ to be nonnegative, it involves a critical norm condition (Q1) and a rank condition (Q2). In contrast, Theorem 5 does not have the norm and rank conditions; but it requires $W(t)$ to have the nonnegativity property (O1) and joint connectivity property (O2). ♦

3.3 Simulation

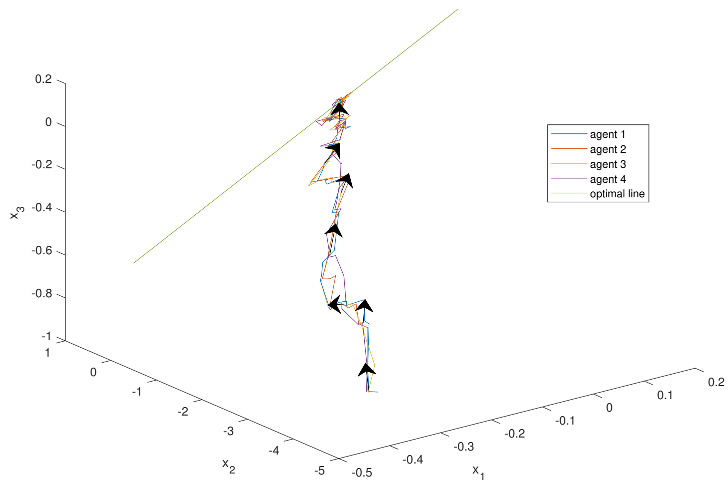
3.3.1 Verification

To illustrate the convergence property of the proposed cooperative learning consensus protocol under different suggested convergence directions for the linear case, we consider two cost functions $F_1(\boldsymbol{x}) = |x_1 - x_2| + |x_3 - x_1|$ and $F_2(\boldsymbol{x}) = \max\{x_1, x_2, x_3\} - \min\{x_1, x_2, x_3\}$ for $\boldsymbol{x} = [x_1, x_2, x_3]^{\mathrm{T}} \in \mathbb{R}^3$. The function $\Phi_{ij}(\cdot)$ is chosen as $\Phi_{ij}(x_j(t) - x_i(t)) = x_j(t) - x_i(t)$. The number of bats used in the cooperative learning consensus protocol is 4. The Laplacian matrix L_t of its graph is given by

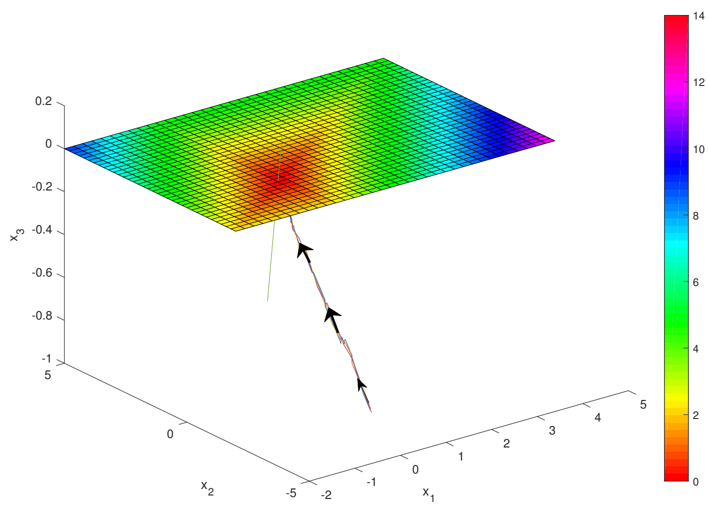
$$L_t = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

and hence the graph contains a ring loop. The frequency $f_i(t)$ is drawn from $[0.2, 0.7]$ randomly and the zooming number $\mu_i(t)$ is drawn from $[0.2, 0.5]$ randomly for every $i = 1, 2, 3, 4$. In the simulation we verified (3.16) and the rank condition Q2) in Theorem 4 at every time instant in order to proceed with the algorithm running. Figure 3.1 shows the convergence of the cooperative learning consensus protocol when minimizing $F_1(\boldsymbol{x})$ while Figure 3.2 shows the convergence of the cooperative learning consensus protocol when minimizing $F_2(\boldsymbol{x}) = \max\{x_1, x_2, x_3\} - \min\{x_1, x_2, x_3\}$.

Next, we consider Theorem 5. Again, the number of bats used in the cooperative learning consensus protocol is 4. The zooming number $\mu_i(t)$ is drawn from $[0.2, 0.7]$ randomly and the frequency $f_i(t)$ is drawn from $[0.2, 0.5]$ randomly, and thus the first condition O1) is satisfied. The Laplacian matrix L_t of its graph is given as follows: $L_{3s+1} = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$, $L_{3s+2} = \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$, and $L_{3s+3} = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$, where $s \in \overline{\mathbb{Z}}_+$. Thus, $\bigcup_{i=1}^{3} \mathcal{E}_{k+i} = \mathcal{E}$ for every $k \in \overline{\mathbb{Z}}_+$, and hence, the second condition O2) is satisfied. Figure 3.3 shows the convergence of the cooperative learning consensus protocol when minimizing $F_1(\boldsymbol{x}) = |x_1 - x_2| + |x_3 - x_1|$ while Figure 3.4 shows the convergence of the cooperative learning consensus protocol when minimizing $F_2(\boldsymbol{x}) = \max\{x_1, x_2, x_3\} - \min\{x_1, x_2, x_3\}$.

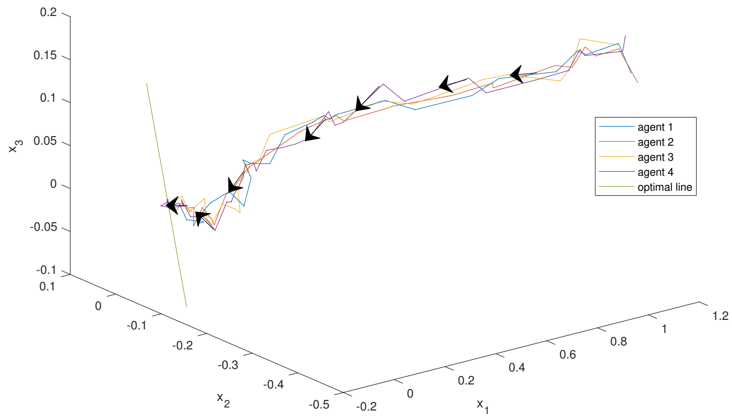


(a) Convergence of cooperative learning consensus.

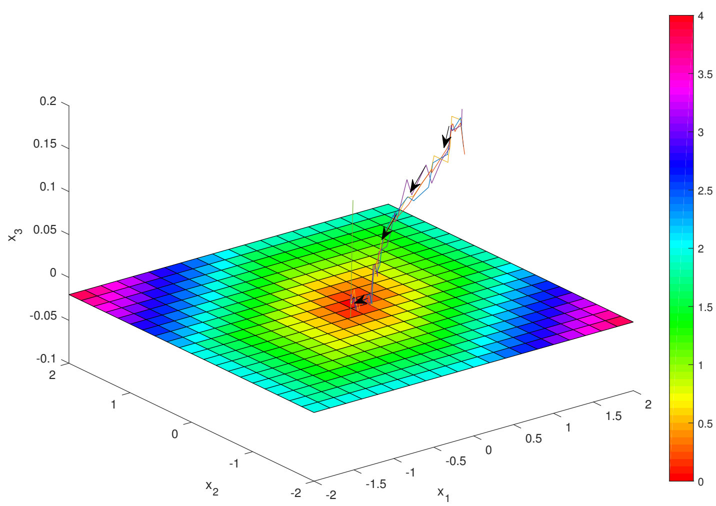


(b) Function value map

Figure 3.1: Convergence of cooperative learning consensus when minimizing $F_1(\boldsymbol{x}) = |x_1 - x_2| + |x_3 - x_1|$: Theorem 4.

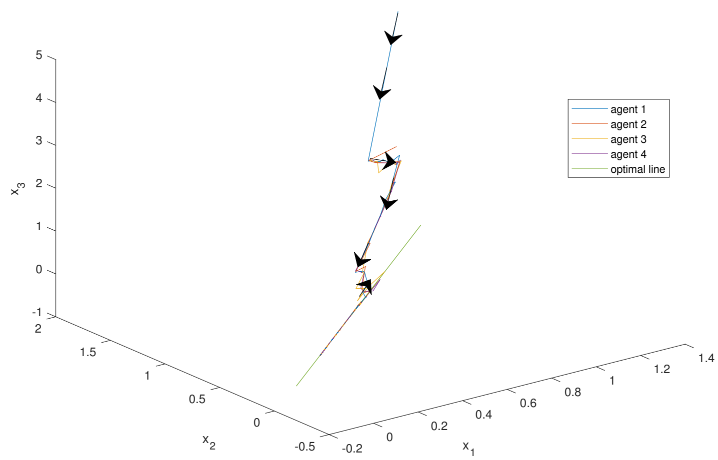


(a) Convergence of cooperative learning consensus.

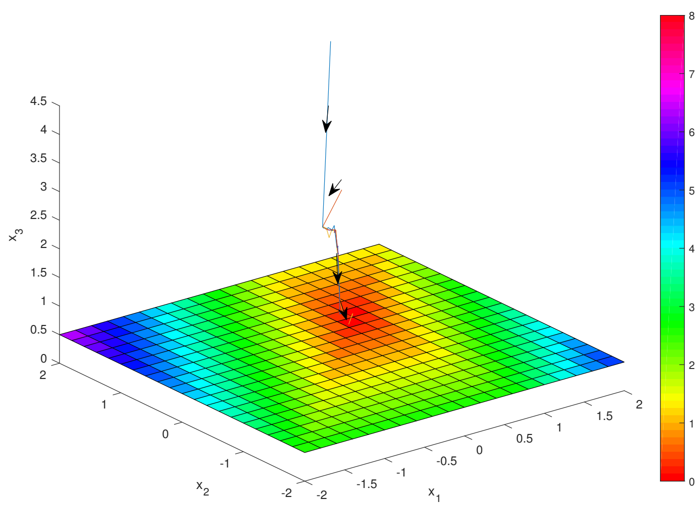


(b) Function value map

Figure 3.2: Convergence of cooperative learning consensus when minimizing $F_2(\boldsymbol{x}) = \max\{x_1, x_2, x_3\} - \min\{x_1, x_2, x_3\}$: Theorem 4.

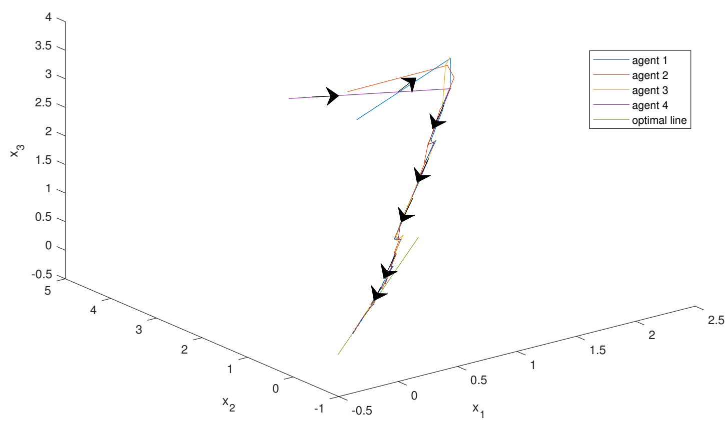


(a) Convergence of cooperative learning consensus.

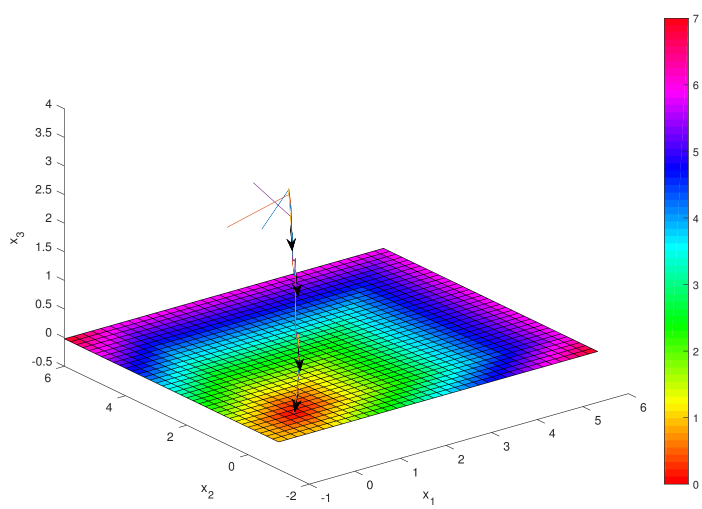


(b) Function value map.

Figure 3.3: Convergence of the cooperative learning consensus protocol when minimizing $F_1(\boldsymbol{x}) = |x_1 - x_2| + |x_3 - x_1|$: Theorem 5.



(a) Convergence of cooperative learning consensus.



(b) Function value map.

Figure 3.4: Convergence of the cooperative learning consensus protocol when minimizing $F_2(\boldsymbol{x}) = \max\{x_1, x_2, x_3\} - \min\{x_1, x_2, x_3\}$: Theorem 5.

3.3.2 Comparison

In this part, we will conduct a comparison simulation to demonstrate the key difference–the ability to solve a separate, unrelated optimization problem–between the proposed cooperative learning consensus protocol and the existing, most dominant average consensens protocols in the literature. Here we use two benchmark test functions from [33] serving as the cost function $F(\boldsymbol{x})$ of the separate optimization problem $\min_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x})$. The first one is Rosenbrock's Valley function: $F_1(\boldsymbol{x}) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$ and the second one is Zakharov function: $F_2(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2 + (0.5ix_i)^2 + (0.5ix_i)^4$. The global minimum for both of them is 0 at $x_i = 1$ and at $x_i = 0$, respectively, $i = 1, \ldots, n$.

In the simulation, we take $\boldsymbol{x} = [x_1, x_2, x_3, x_4]^{\mathrm{T}}$. For both $F_1(\boldsymbol{x})$ and $F_2(\boldsymbol{x})$, we run 30 times for both the average consensus (AC) and the proposed cooperative learning consensus (CLC). For each run, the initial value for both consensus protocols are the same. Finally, we take the average of their state information in 30 runs as the ultimate value for $\boldsymbol{x}$, and use it to compute $F_1(\boldsymbol{x})$ and $F_2(\boldsymbol{x})$ for comparison of both consensus protocols. The numerical comparisons are provided in Tables 3.1 and 3.2. According to these results, the CLC protocol can achieve much better statistical performance than the AC protocol.

Table 3.1: Comparison of the proposed cooperative learning consensus and average consensus for $F_1(\boldsymbol{x})$ after 30 run times

Algorithm	Min	Max	Median	Average	SD
Cooperative Learning Consensus	2.00E-03	6.07E+01	2.98E+00	5.11E+00	1.04E+01
Average Consensus	1.47E+00	4.57E+03	1.65E+01	2.77E+02	8.32E+02

Figure 3.5 shows the convergence of the average consensus and the proposed cooperative learning consensus under the benchmark function $F_1(\boldsymbol{x})$, respectively. The proposed

Table 3.2: Comparison of the proposed cooperative learning consensus and average consensus for $F_2(\boldsymbol{x})$ after 30 run times

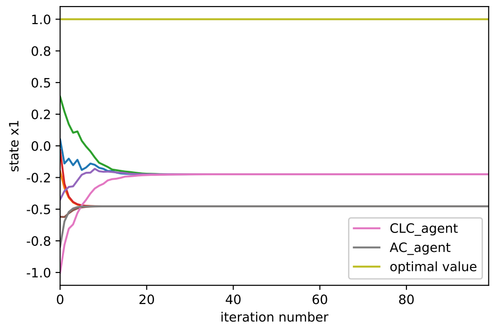
Algorithm	Min	Max	Median	Average	SD
Cooperative Learning Consensus	1.33E-10	2.37E-01	2.197E-05	1.30E-02	4.30E-02
Average Consensus	1.10E-02	2.87E+02	2.03E+00	1.80E+01	5.19E+01

cooperative learning consensus approaches the optimal solution **1** closer than the average consensus. Moreover, the proposed cooperative learning consensus has the steady-state result 15.9276 for $F_1(\boldsymbol{x})$ while the average consensus result for $F_1(\boldsymbol{x})$ is 72.2378. Clearly the proposed cooperative learning consensus outperforms the average consensus by a large margin in this case when solving a minimization problem of $F_1(\boldsymbol{x})$.

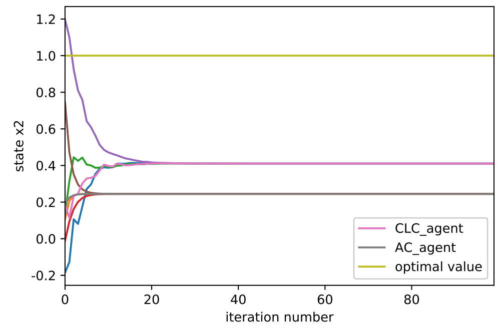
Figure 3.6 shows the convergence of the average consensus and the proposed cooperative learning consensus under the benchmark function $F_2(\boldsymbol{x})$, respectively. The proposed cooperative learning consensus approaches the optimal solution **0** while the average consensus has a bigger deviation from the optimal solution **0**. Moreover, the proposed cooperative learning consensus has the steady-state result 0.0075 for $F_2(\boldsymbol{x})$ while the average consensus result for $F_2(\boldsymbol{x})$ is 1.2405. Clearly the proposed cooperative learning consensus is still far superior to the average consensus in this case when solving a minimization problem of $F_2(\boldsymbol{x})$.

3.4 Conclusion

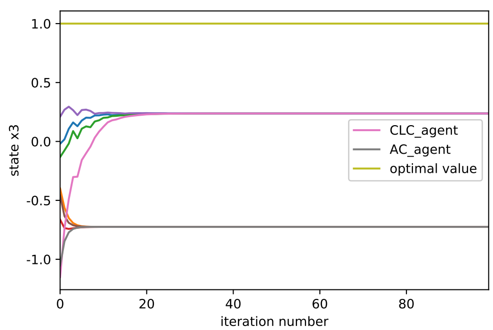
In this chapter, motivated by the bat searching algorithm in swarm intelligence, a new class of cooperative learning consensus protocols were proposed. This new consensus protocol can simultaneously fulfill two tasks in one framework: an optimization problem and a consensus achievement problem. The onvergene analysis of the proposed cooperative learning consensus protocol was discussed in details for the linear case by means of nonnegative and paracontracting matrix analysis.



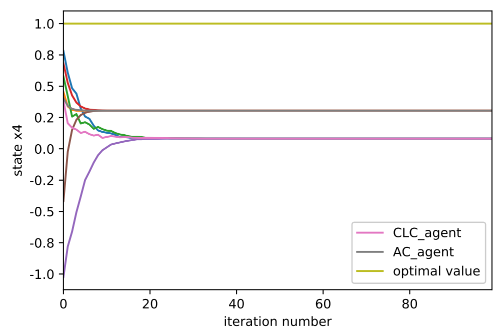
(a) x_1 versus iteration number.



(b) x_2 versus iteration number.



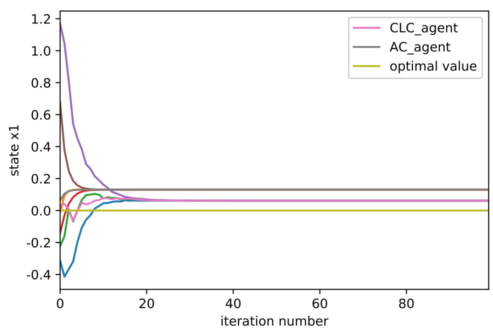
(c) x_3 versus iteration number.



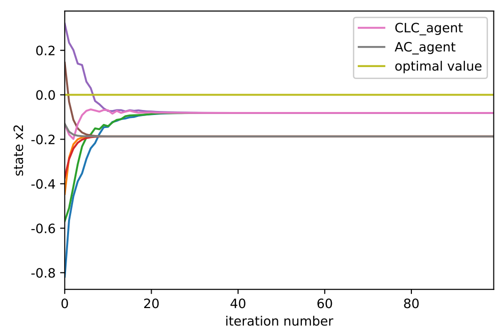
(d) x_4 versus iteration number.

Figure 3.5: Convergence of average consensus for Rosenbrock's Valley function.

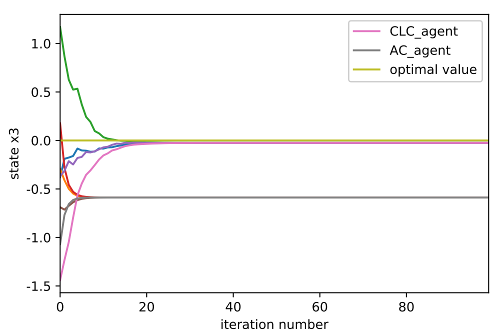
One important remaining problem for the proposed cooperative learning consensus protocol is the time delay issue for the multihop relay protocol part when the two-step information exchange is going among these bats. when the two-step information exchange is going among these bats. In this chapter, we just assumed that this two-step information exchange occurs at every time instant. It would be interesting to see what will happen to the proposed cooperative learning consensus protocol if there is a time delay or latency in this information exchange process. Note that some time delay issues have been discussed for continuous-time consensus protocols under a general framework of semistability theory for nonlinear dynamical systems [70]. In the next chapter, we will explore the performance of this algorithm for the nonliear case of the flux function.



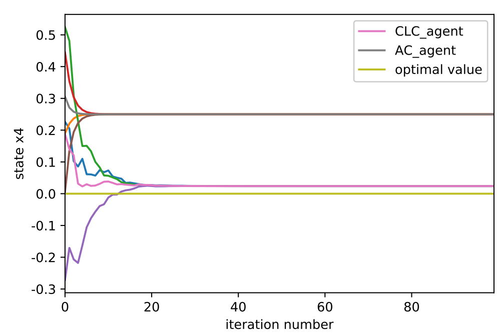
(a) x_1 versus iteration number.



(b) x_2 versus iteration number.



(c) x_3 versus iteration number.



(d) x_4 versus iteration number.

Figure 3.6: Convergence of cooperative learning consensus for minimizing Rosenbrock's Valley function.

CHAPTER 4

THE BIO-INSPIRED COOPERATIVE LEARNING CONSENSUS UNDER SUGGESTED CONVERGENCE DIRECTION: NONLINEAR CASE

4.1 Introduction

Distributed network coordination of dynamic multi-agent systems has attracted a lot of attention in recent years from many aspects such as the application of multi-agent systems in formation control [71–73], flocking and fish schooling [74, 75], and distributed sensor networks [76]. The foundation of designing such distributed network coordinated algorithms is the *consensus protocols* derived from corresponding consensus problems. The consensus problems have been studied considerably in many different fields. In [77], the authors provide a general framework of the consensus problems to study multi-agent systems with fixed and switching topologies. A distributed algorithm is proposed to reach the global consensus in finite time in [78]. In [79], the distributed H_∞ consensus problems are explored for multi-agent systems subject to external disturbances and uncertainties. The authors in [80] and [81] propose the consensus algorithms for a group of agents with limited communication data rate. A consensus tracking algorithm is proposed in [82] for multi-agents with fixed and switching topologies to ensure that the state of the agents follows a reference of the leader. In [83], to deal with the group coordination problem with undirected communication graphs, a passivity-based design framework is proposed.

Among these research topics, some of the consensus problems are motivated by swarm intelligence. Many researchers have constructed mathematical models to simulate their collective behaviors in biological systems [84], [85]. Later, various mathematical techniques are used by control researchers to develop a rigorous control-theoretic framework to conduct theoretical analysis of these behaviors. In [86], a nonnegative matrix analysis technique is used to investigate the coordination behavior of multi-agent systems. The authors in [87] show that the algebraic graph theory can be used to analyze the consensus protocol for multi-agent systems under dynamically changing interaction topologies. In [88], a Lyapunov-based method is used to develop a general framework for designing consensus protocols in dynamic networks for achieving multi-agent coordination in finite time. Motivated by the bat searching (BS) algorithm, we propose a new bat-inspired consensus protocol in this chapter. It combines the original bat algorithm with the idea of multi-agent coordination optimization (MCO) algorithm [89]. The proposed consensus protocol is embedded into a separate optimization problem first. This embedded optimization problem guides the convergence direction. The proposed consensus protocol has some nonlinear dynamics which differs from the consensus protocol in last chapter. Meanwhile, we consider a multi-agent system with external disturbances to design its consensus protocols. These proposed consensus protocols have a potential of solving a simultaneous optimized planning and regulating problem for agent-based autonomous systems, with applications in manufacturing, robotics, and power systems.

4.2 Bat-Inspired Consensus

The notation used in this chapter are presented here. Some time-dependent, algebraic graph related notations are used to describe the cooperative bat-inspired consensus protocols.

Specifically, $\mathbb{R}$ denotes the set of real number, $\overline{\mathbb{R}}_+$ denotes the set of nonnegative real numbers, $\mathbb{R}^n$ denotes the set of n-dimensional real column vectors, $\mathbb{R}^{n\times m}$ denotes the set of n-by-m matrices, $(\cdot)^T$ denotes transpose, $(\cdot)^{-1}$ denotes inverse, and $||\cdot||$ denotes the Euclidean norm, respectively. Next, $\otimes$ denotes the Kronecker product. diag(x) denotes a square diagonal matrix with the elements of vector x on the main diagonal. Let $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ denote a dynamic directed graph with the set of vertices $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ and $\mathcal{E}(t) \subset \mathcal{V}\times\mathcal{V}$ representing the set of edges, where $t \in \bar{\mathbb{Z}}_+ = \{0, 1, 2, \ldots\}$. A graph with the property that $(v_i, v_j) \in \mathcal{E}$ implies $(v_j, v_i) \in \mathcal{E}$ is said to be undirected. The time-varying adjacency matrix $A(t) \in \mathbb{R}^{N\times N}$ associated with the directed graph $\mathcal{G}$ is defined by nonnegative adjacency elements $a_{ij}(t)$ as $A(t) = [a_{ij}(t)]$. We assume $(v_i, v_j) \in \mathcal{E}(t)$ if and only if $a_{ij} = 1$, $(v_i, v_j) \notin \mathcal{E}(t)$ if and only if $a_{ij} = 0$, and $a_{ii} = 0$ for all $i \in \mathcal{N}$, where $\mathcal{N} = \{1, 2, \ldots, N\}$ denotes the node index of $\mathcal{G}(t)$. The set of neighbors of the node v_i is denoted by $\mathcal{N}^i(t) = \{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}(t), j = 1, 2, \ldots, |\mathcal{N}|, j \neq i\}$, where $|\mathcal{N}|$ denotes the cardinality of $\mathcal{N}$. The degree matrix of a dynamic graph $\mathcal{G}(t)$ is defined by $\delta(t) = [d_{ij}(t)]$, where $i, j \in \{1, 2, \ldots, |\mathcal{N}|\}$ and

$$d_{ij}(t) = \begin{cases} \sum_{j=1}^{|\mathcal{N}|} a_{ij}(t), & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

The Laplacian matrix of the dynamic graph $\mathcal{G}$ is defined by $L(t) = \delta(t) - A(t)$. If there is a path from any node to any other node in a dynamic graph, the dynamic graph is called *strongly connected*.

We consider a group of N bats who can communicate with their neighboring bats via a communication graph topology $\mathcal{G}(t)$ at each time instant t. Each node k in the graph corresponds to a labeled bat k, $k = 1, 2, \ldots, N$. Next, we will make two standing assumptions throughout the chapter. The first one is about the connectivity of $\mathcal{G}(t)$.

Assumption 3. The communication graph $\mathcal{G}(t)$ is strongly connected, i.e., without loss of generality, assuming that there exists a path in the N bats:

$$\text{Bat } 1 \leftrightarrow \text{Bat } 2 \leftrightarrow \cdots \leftrightarrow \text{Bat } N \leftrightarrow \text{Bat } 1 \tag{4.1}$$

Meanwhile, we assume that all of the bats are able to access to their neighbors' state information and serve as routers to transfer some data to their own neighbors.

The next assumption is about a separate optimization problem embedded in the proposed bat consensus protocol.

Assumption 4. The minimization problem $\min_{x\in\mathbb{R}^n} F(x)$ has a solution, where $F : \mathbb{R}^n \to \mathbb{R}$.

The original bat algorithm was based on the echolocation or bio-sonar characteristics of microbats [35]. The bats can update their position information by following certain rules to find the prey. The update rules for solving an optimization problem $\min_{x\in\mathbb{R}^n} F(x)$ are given by the following form:

$$\begin{aligned} f_i(t) &= f_{\min} + (f_{\max} - f_{\min})\beta_i(t), \\ v_i(t+1) &= v_i(t) + [x_i(t) - p(t)]f_i(t), \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \tag{4.2}$$

where $x_i(t)$ and $v_i(t)$ are the position and velocity of Bat i at each time instant t, respectively. $f_i(t)$ is the frequency information for Bat i at time instant t. $f_{\min}$ and $f_{\max}$ are the lower bound and upper bound of the frequency for Bat i, respectively. $\beta_i(t) \in [0, 1]$ is a random vector drawn from a uniform distribution, and $p(t)$ is the current best global solution at time instant t, i.e.,

$$p(t) = \arg \min_{1\leq i\leq N, 0\leq s\leq t} F(x_i(s))$$

Based on this algorithm, we propose a bat-inspired consensus protocol. We are considering here that all of the bats have the same constant speed, but with different heading angles. The proposed consensus protocol for heading angles of the bats can be used to asymptotically achieve a common heading angle among all of the bats. Thus, the consensus protocol described here for tackling heading angles of the bats can be described as follows:

$$\boldsymbol{\theta}_{1,\min}(t) = \boldsymbol{\theta}_1(t), \tag{4.3}$$

$$\begin{aligned} \boldsymbol{\theta}_{k+1,\min}(t) = &\arg\min\{F(\boldsymbol{\theta}_{k,\min}(t)), F(\boldsymbol{\theta}_{k+1}(t))\}, \\ &k = 1, \ldots, 2N-1 \end{aligned} \tag{4.4}$$

$$\boldsymbol{\theta}_{2N,\min}(t) = \boldsymbol{\theta}_{2N-1,\min}(t), \tag{4.5}$$

$$\begin{aligned} \boldsymbol{\theta}_i(t+1) = &\boldsymbol{\theta}_i(t) + f_i(t)\{\arg\min_{\boldsymbol{\theta}(t)\in\mathbb{R}^n} \sum_{j\in\mathcal{N}_t^i} \\ &[\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}(t)]^T \Phi_{ij}(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}(t)) - \boldsymbol{\theta}_i(t)\} \\ &+ f_i(t)\mu_i(t)[\boldsymbol{\theta}_{N+i,\min}(t) - \boldsymbol{\theta}_i(t)], \end{aligned} \tag{4.6}$$

$$f_i(t) = f_{\min} + \beta(t)(f_{\max} - f_{\min}), i = 1, \ldots, N \tag{4.7}$$

where $t \in \overline{\mathbb{Z}}_+, \boldsymbol{\theta}_i(t) = \boldsymbol{\theta}_{N+i} \in \mathbb{R}^m$ denotes the heading angle of Bat i at time instant t. $\beta,\ f_{\min}$ and $f_{\max}$ have the same meaning as in (4.2). $0 < \mu_i(t) < 1$ is the zooming parameter for Bat i. $\Phi_{ij} : \mathbb{R}^m \to \mathbb{R}^m$ is a vector-valued flux function which satisfies $\Phi_{ij}(x) = 0$ if and only if $x = 0$ and $x^T\Phi_{ij}x \geq 0$ for every $x \in \mathbb{R}^m$ and every $i, j = 1, \ldots, N, i \neq j$. The flux function $\Phi_{ij}(\cdot)$ is convex and can be interpreted as an energy/mass exchange rate in compartmental modeling [90], [91] or a heat transfer rate in thermodynamics [92]. It should be noted that the original bat algorithm does not have the interconnected term $\arg\min_{\boldsymbol{\theta}(t)\in\mathbb{R}^m} \sum_{j\in\mathcal{N}_t^i}[\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}(t)]^T\Phi_{ij}(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}(t))$ in (4.6). This term stems from the speed-up and speed-down strategy which was derived from biological swarms [93]. The term $\boldsymbol{\theta}_{N+i,\min}(t)$ in (4.6), which is equivalent to $\arg\min_{1\leq i\leq N} F(\boldsymbol{\theta}_i(t))$, is the suggested convergence direction. This term is calculated through the *mutihop communication protocol* (4.3)-(4.5) based on the communication loop path (4.1), which includes the following two steps:

1. Bat $k+1$ can receive the state information $\boldsymbol{\theta}_{k,\min}(t)$ from Bat k at time instant t, $k = 1, \ldots, N-1$. At the same time, Bat $k+1$ determines

$$\boldsymbol{\theta}_{k+1,\min}(t) = \arg\min\{F(\boldsymbol{\theta}_{k,\min}(t)), F(\boldsymbol{\theta}_{k+1}(t))\}$$

   and meanwhile, it serves as a router to send $\boldsymbol{\theta}_{k+1,\min}(t)$ to the next Bat in (4.1).

2. After $\boldsymbol{\theta}_{N,\min}(t)$ is determined by Bat N, this information is passed to Bat ($N+1$ mod N), which is essentially Bat 1, where mod denotes the modulo operation. Bat (k mod N), $k = N+1, \ldots, 2N-1$, again determines

$$\boldsymbol{\theta}_{k,\min}(t) = \arg\min\{F(\boldsymbol{\theta}_{k-1,\min}(t)), F(\boldsymbol{\theta}_{(k \bmod N)}(t))\}$$

   and serves as a router to send $\theta_{k,\min}(t)$ to Bat ($k+1$ mod N) by sequentially following the directed communication path

$$\begin{aligned} &\text{Bat } N \leftrightarrow \text{Bat } (N+1 \bmod N) \leftrightarrow \text{Bat } (N+2 \bmod N) \\ &\leftrightarrow \cdots \leftrightarrow \text{Bat } (2N-1 \bmod N), \end{aligned}$$

   which equals

$$\text{Bat } N \leftrightarrow \text{Bat } 1 \leftrightarrow \text{Bat } 2 \leftrightarrow \cdots \leftrightarrow \text{Bat } (N-1)$$

   thus, this implies (4.1).

It should be noted that we use a "double-check" technique in these two steps to obtain

$\arg\min_{1\leq i\leq N} F(\boldsymbol{\theta}_i(t))$. Specifically, after Step 1, we can find that $\boldsymbol{\theta}_{N,\min}(t)$ obtained by Bat N is indeed $\arg\min_{1\leq i\leq N} F(\boldsymbol{\theta}_i(t))$. Thus, in Step 2, $\boldsymbol{\theta}_{N+i,\min}(t)$ obtained by Bat i equals $\boldsymbol{\theta}_{N,\min}(t)$ for every $i = 1, \ldots, N-1$. However, we still let Bat i perform the comparison operation

$$\boldsymbol{\theta}_{N+i,\min}(t) = \arg\min\{F(\boldsymbol{\theta}_{N+i-1,\min}(t)), F(\boldsymbol{\theta}_i(t))\}$$

in Step 2. This operation can guarantee that we can acquire exact $\arg\min_{1\leq i\leq N} F(\boldsymbol{\theta}_i(t))$ without any major error.

Henceforth, this consensus protocol is distinct from existing consensus protocols described in the literature. Meanwhile, this consensus protocol can be implemented in a distributed manner by determining $\boldsymbol{\theta}_{N+i,\min}(t) = \arg\min_{1\leq i\leq N} F(\boldsymbol{\theta}_i(t))$ locally, which differs from the BS algorithm. $\arg\min_{1\leq i\leq N} F(\boldsymbol{\theta}_i(t))$ computed by this algorithm is in a global manner which requires that all of the bats' position information is known to each bat.

4.3 Convergence Analysis of Bat-Inspired Consensus

One of the major questions around the proposed bat-inspired consensus protocol is its convergence property. Given an optimization problem $\min_{x\in\mathbb{R}^n} F(x)$, how can we guarantee that the proposed consensus protocol converges asymptotically? Here convergence means that

$$\lim_{t\to\infty} \boldsymbol{\theta}_1(t) = \cdots = \lim_{t\to\infty} \boldsymbol{\theta}_N(t)$$

exists for the consensus protocol (4.3)-(4.7). To analyze the convergence of this consensus protocol, we first need to focus on the form of $\Phi_{ij}(\cdot)$. In last chapter, the linear form was proposed and subsequently, the convergence analysis can be conducted by using the matrix paracontraction technique. However, if the function $\Phi_{ij}(\cdot)$ is nonlinear, then we cannot convert it into a linear matrix form, and hence, cannot use the matrix paracontraction method to determine the convergence of this consensus protocol. Next, we will present a technique to deal with the convergence of the bat-inspired consensus protocol with a nonlinear form of $\Phi_{ij}(\cdot)$.

The bat-inspired consensus protocol can be rewritten as follows:

$$\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) + f_i(t)u_i(t) + f_i(t)\mu_i(t)[\boldsymbol{\theta}_{N+i,\min}(t) - \boldsymbol{\theta}_i(t)] \tag{4.8}$$

and $u_i(t)$ is defined as

$$u_i(t) = g_i(\boldsymbol{\theta}(t)) - \boldsymbol{\theta}_i(t)$$

where

$$g_i(\boldsymbol{\theta}(t)) = \arg\min_{\boldsymbol{\theta}(t)\in\mathbb{R}^m} \sum_{j\in\mathcal{N}_t^i} (\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}(t))\Phi_{ij}(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}(t))$$

Since $g_i(\boldsymbol{\theta}(t))$ is an argmin mapping, we can write $g_i(\boldsymbol{\theta}(t))$ as $g_i(\boldsymbol{\theta}_{ip}, \boldsymbol{\theta}(t))$, where $\boldsymbol{\theta}_{ip} = \text{col}(\boldsymbol{\theta}_j(t)), j \in \mathcal{N}^i$ is the parameter of the function $g_i(\boldsymbol{\theta}(t))$ and "col" denotes the columnization operation. Next, to perform our analysis, we will make the following assumption about the function $g_i(\boldsymbol{\theta}(t))$.

Assumption 5. $g_i(\boldsymbol{\theta})$ is semi-Lipschitz continuous, that is, there exists a constant γ such that

$$||g_i(\boldsymbol{\theta}) - g_j(\boldsymbol{\theta})|| \leq \gamma||\boldsymbol{\theta}_{ip} - \boldsymbol{\theta}_{jp}||$$

where $\boldsymbol{\theta}_{ip}$ and $\boldsymbol{\theta}_{jp}$ are the parameters of $g_i(\boldsymbol{\theta})$ and $g_j(\boldsymbol{\theta})$, respectively, and the domain of $\boldsymbol{\theta}$ is compact and convex.

We have the following theorem about the convergence of the proposed consensus protocol.

Theorem 6. Consider (4.8) with (4.3)–(4.5) and (4.7). Assume that Assumptions 1, 2, and 5 hold. Furthermore, assume that there exist three positive constants $c_1, c_2, c_3 > 0$ and a positive-definite matrix $P = P^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ such that

$$\begin{bmatrix} \Gamma_{11}(t) & \Gamma_{12}(t) & \Gamma_{13}(t) \\ * & \Gamma_{22}(t) & \Gamma_{23}(t) \\ * & * & \Gamma_{33}(t) \end{bmatrix} \leq -c_3 I_{3nq} \tag{4.9}$$

where $\Gamma_{11}(t) = (-c_2 + 2c_1)M_1(t) + 2c_1\gamma^2 q^2 \bar{q}(t)M(t)$, $\Gamma_{12}(t) = (P\otimes I_q)M_2(t)$, $\Gamma_{13}(t) = (P\otimes I_q)M_3(t)$, $\Gamma_{22}(t) = M_2^{\mathrm{T}}(t)(P\otimes I_q)M_2(t) - c_1M_1(t)$, $\Gamma_{23}(t) = M_2^{\mathrm{T}}(t)(P\otimes I_q)M_3(t)$, $\Gamma_{33}(t) = M_3^{\mathrm{T}}(t)(P\otimes I_q)M_3(t) + c_2M_1(t)$, $\bar{q}(t) = \sum_{i=1}^{q}|\mathcal{N}_t^i|$, $M(t) = \mathrm{diag}(\sum_{i=1}^{q}\mathrm{row}_i(A_t)) \otimes I_n$, $\mathrm{row}_i(A_t)$ denotes the ith row of the adjacency matrix A_t, $M_1(t) = (2qI_q - 2\mathbf{1}_{q\times q})\otimes I_n$, $M_2(t) = F_2(t)\otimes I_n$, $M_3(t) = F_3(t)\otimes I_n$, $F_2(t) = \mathrm{diag}(f(t)) - (1/q)(\mathbf{1}_{q\times 1}f^{\mathrm{T}}(t))$, $F_3(t) = \mathrm{diag}(\Delta f(t)) - (1/q)(\mathbf{1}_{q\times 1}\Delta f^{\mathrm{T}}(t))$, $f(t) = [f_1(t), \ldots, f_q(t)]^{\mathrm{T}}$, $\Delta f(t) = [f_1(t)\mu_1(t), \ldots, f_q(t)\mu_q(t)]^{\mathrm{T}}$, and "$*$" is used for the blocks induced by symmetry. Then the heading angle vectors of the q bats will asymptotically reach consensus under the proposed cooperative learning protocol (4.8) with (4.3)–(4.5) and (4.7). Moreover, this consensus reaching is uniform in t, that is, $\boldsymbol{\theta}_i(t) \rightrightarrows \boldsymbol{\theta}^{\dagger}$ as $t \to \infty$ for every $i = 1, \ldots, q$, where $\boldsymbol{\theta}^{\dagger} = \lim_{t\to\infty}\arg\min_{1\leq i\leq q} F(\boldsymbol{\theta}_i(t))$ and "$\rightrightarrows$" denotes uniform convergence.

Proof. Define $e_i(t) = \boldsymbol{\theta}_i(t) - (1/q)\sum_{j=1}^{q}\boldsymbol{\theta}_j(t) = (1/q)\sum_{j=1}^{q}(\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t))$ for every

$i = 1, \ldots, q$ and every $t \in \overline{\mathbb{Z}}_+$. Then it follows that for every $i = 1, \ldots, q$,

$$\begin{aligned}
e_i(t+1) =& \boldsymbol{\theta}_i(t+1) - (1/q)\sum_{j=1}^{q}\boldsymbol{\theta}_j(t+1)\\
=&(1/q)\sum_{j=1}^{q}[\boldsymbol{\theta}_i(t+1) - \boldsymbol{\theta}_j(t+1)]\\
=&(1/q)\sum_{j=1}^{q}[\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t)] + (1/q)\sum_{j=1}^{q}[f_i(t)\boldsymbol{u}_i(t) - f_j(t)\boldsymbol{u}_j(t)]\\
&+ (1/q)\sum_{j=1}^{q}[f_i(t)\mu_i(t)(\boldsymbol{\theta}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)) - f_j(t)\mu_j(t)(\boldsymbol{\theta}_{q+j,\min}(t) - \boldsymbol{\theta}_j(t))]\\
=&e_i(t) + (1/q)\sum_{j=1}^{q}[f_i(t)(g_t^i(\boldsymbol{\theta}(t)) - \boldsymbol{\theta}_i(t)) - f_j(t)(g_t^j(\boldsymbol{\theta}(t)) - \boldsymbol{\theta}_j(t))]\\
&+ (1/q)\sum_{j=1}^{q}[f_i(t)\mu_i(t)(\boldsymbol{\theta}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)) - f_j(t)\mu_j(t)(\boldsymbol{\theta}_{q+j,\min}(t) - \boldsymbol{\theta}_j(t))]
\end{aligned}$$

Let $\Delta G_i(t) = (1/q)\sum_{j=1}^{q}[f_i(t)(g_t^i(\boldsymbol{\theta}(t)) - \boldsymbol{\theta}_i(t)) - f_j(t)(g_t^i(\boldsymbol{\theta}(t)) - \boldsymbol{\theta}_j(t))]$ and $\Delta\xi_i(t) = (1/q)\sum_{j=1}^{q}[f_i(t)\mu_i(t)(\boldsymbol{\theta}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)) - f_j(t)\mu_j(t)(\boldsymbol{\theta}_{q+j,\min}(t) - \boldsymbol{\theta}_j(t))]$, $i = 1, \ldots, q$. Then for every $i = 1, \ldots, q$,

$$e_i(t+1) = e_i(t) + \Delta G_i(t) + \Delta\xi_i(t) \tag{4.10}$$

Now we can choose the Lyapunov function candidate $V(t) = \sum_{i=1}^{q} e_i^{\mathrm{T}}(t)Pe_i(t)$, where $P = P^{\mathrm{T}}$ is a positive-definite matrix, and it follows that

$$\begin{aligned}
\Delta V(t) &= V(t+1) - V(t)\\
&= \sum_{i=1}^{q} e_i^{\mathrm{T}}(t+1)Pe_i(t+1) - \sum_{i=1}^{q} e_i^{\mathrm{T}}(t)Pe_i(t)
\end{aligned} \tag{4.11}$$

Substituting (4.10) into (4.11) yields

$$\Delta V(t) = \sum_{i=1}^{q} \Big\{ e_i^{\mathrm{T}}(t) P \Delta G_i(t) + e_i^{\mathrm{T}}(t) P \Delta \xi_i(t) + \Delta G_i^{\mathrm{T}}(t) P e_i(t) + \Delta G_i^{\mathrm{T}}(t) P \Delta G_i(t)$$
$$+ \Delta G_i^{\mathrm{T}}(t) P \Delta \xi_i(t) + \Delta \xi_i^{\mathrm{T}}(t) P e_i(t)$$
$$+ \Delta \xi_i^{\mathrm{T}}(t) P \Delta G_i(t) + \Delta \xi_i^{\mathrm{T}}(t) P \Delta \xi_i(t) \Big\} \qquad (4.12)$$

Let $\Delta g(t) = [\Delta g_1^{\mathrm{T}}(t), \ldots, \Delta g_q^{\mathrm{T}}(t)]^{\mathrm{T}}$, where $\Delta g_i(t) = g_t^i(\boldsymbol{\theta}(t)) - \boldsymbol{\theta}_i(t)$, $i = 1, \ldots, q$. Since, by Assumption 5, $g_t^i(\boldsymbol{\theta}(t))$ is uniformly pseudo-Lipschitz continuous in terms of its parameters, it follows that for every $i, j = 1, \ldots, q$,

$$\begin{aligned} ||\Delta g_i(t) - \Delta g_j(t)|| &\leq ||g_t^i(\boldsymbol{\theta}(t)) - g_t^j(\boldsymbol{\theta}(t))|| + ||\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t)|| \\ &\leq \sum_{a \in \mathcal{N}_t^i} \sum_{b \in \mathcal{N}_t^j} \gamma ||\boldsymbol{\theta}_a(t) - \boldsymbol{\theta}_b(t)|| + ||\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t)|| \\ &= \sum_{a \in \mathcal{N}_t^i} \sum_{b \in \mathcal{N}_t^j} \gamma ||e_a(t) - e_b(t)|| + ||e_i(t) - e_j(t)|| \end{aligned}$$

where γ is the uniform pseudo-Lipschitz constant. By using the fact $||a + b||^2 \leq 2(||a||^2 + ||b||^2)$ and the Cauchy-Schwarz inequalty, for every $i, j = 1, \ldots, q$, we have

$$\begin{aligned} ||\Delta g_i(t) - \Delta g_j(t)||^2 &\leq 2 \Bigg(\sum_{a \in \mathcal{N}_t^i} \sum_{b \in \mathcal{N}_t^j} \gamma ||e_a(t) - e_b(t)|| \Bigg)^2 + 2||e_i(t) - e_j(t)||^2 \\ &\leq 2\gamma^2 (|\mathcal{N}_t^i| \cdot |\mathcal{N}_t^j|) \sum_{a \in \mathcal{N}_t^i} \sum_{b \in \mathcal{N}_t^j} ||e_a(t) - e_b(t)||^2 + 2||e_i(t) - e_j(t)||^2 \end{aligned}$$

Thus, for every $i = 1, \ldots, q$,

$$\sum_{j=1}^{q} ||\Delta g_i(t) - \Delta g_j(t)||^2 \le 2\gamma^2 \sum_{j=1}^{q} \Big((|\mathcal{N}_t^i| \cdot |\mathcal{N}_t^j|) \sum_{a\in\mathcal{N}_t^i} \sum_{b\in\mathcal{N}_t^j} ||e_a(t) - e_b(t)||^2 \Big)$$
$$+ 2\sum_{j=1}^{q} ||e_i(t) - e_j(t)||^2$$
$$\le 2\gamma^2 \sum_{j=1}^{q} \Big((|\mathcal{N}_t^i| \cdot |\mathcal{N}_t^j|) \sum_{a\in\mathcal{N}_t^i} \sum_{b\in\mathcal{N}_t^j} (||e_a(t)||^2 + ||e_b(t)||^2) \Big)$$
$$+ 2\sum_{j=1}^{q} ||e_i(t) - e_j(t)||^2$$

Since $|\mathcal{N}_t^i| \le q$ for every $i = 1, \ldots, q$ and every $t \in \overline{\mathbb{Z}}_+$, it further follows that for every $i = 1, \ldots, q$,

$$\sum_{j=1}^{q} ||\Delta g_i(t) - \Delta g_j(t)||^2 \le 2\gamma^2 q^2 \sum_{j=1}^{q} \Big(\sum_{a\in\mathcal{N}_t^i} \sum_{b\in\mathcal{N}_t^j} (||e_a(t)||^2 + ||e_b(t)||^2) \Big)$$
$$+ 2\sum_{j=1}^{q} ||e_i(t) - e_j(t)||^2$$
$$= 2\gamma^2 q^2 \big[(|\mathcal{N}_t^1| + \cdots + |\mathcal{N}_t^q|) e^{\mathrm{T}}(t) (\mathrm{diag}(\mathrm{row}_i(A_t)) \otimes I_n) e(t)$$
$$+ |\mathcal{N}_t^i| e^{\mathrm{T}}(t) (\mathrm{diag}(\mathrm{row}_1(A_t) + \cdots + \mathrm{row}_q(A_t)) \otimes I_n) e(t) \big]$$
$$+ 2\sum_{j=1}^{q} ||e_i(t) - e_j(t)||^2$$

where $e(t) = [e_1^{\mathrm{T}}(t), \ldots, e_q^{\mathrm{T}}(t)]^{\mathrm{T}}$ and $\mathrm{row}_i(A_t)$ is the ith row of the adjacency matrix A_t. Let $\bar{q}(t) = |\mathcal{N}_t^1| + \cdots + |\mathcal{N}_t^q|$ and $M(t) = \mathrm{diag}(\mathrm{row}_1(A_t) + \cdots + \mathrm{row}_q(A_t)) \otimes I_n$, then for any positive constant $c_1 > 0$, we can obtain

$$c_1 \Delta g^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n \Delta g(t) \le 2c_1\gamma^2 q^2 \bar{q}(t) e^{\mathrm{T}}(t) M(t) e(t)$$
$$+ 2c_1 e^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n e(t) \tag{4.13}$$

Let $\Delta\xi(t) = [\Delta\xi_1^{\mathrm{T}}(t), \ldots, \Delta\xi_q^{\mathrm{T}}(t)]^{\mathrm{T}}$ and $\Delta G(t) = [\Delta G_1^{\mathrm{T}}(t), \ldots, \Delta G_q^{\mathrm{T}}(t)]^{\mathrm{T}}$, then it follows from (4.12) that

$$\begin{aligned}\Delta V(t) =& e^{\mathrm{T}}(t)(P \otimes I_q)\Delta G(t) + e^{\mathrm{T}}(t)(P \otimes I_q)\Delta\xi(t) + \Delta G^{\mathrm{T}}(t)(P \otimes I_q)e(t)\\ &+ \Delta G^{\mathrm{T}}(t)(P \otimes I_q)\Delta G(t) + \Delta G^{\mathrm{T}}(t)(P \otimes I_q)\Delta\xi(t) + \Delta\xi^{\mathrm{T}}(t)(P \otimes I_q)e(t)\\ &+ \Delta\xi^{\mathrm{T}}(t)(P \otimes I_q)\Delta G(t) + \Delta\xi(t)(P \otimes I_q)\Delta\xi(t) \end{aligned} \tag{4.14}$$

From (4.13), we have

$$\begin{aligned}&- c_1\Delta g^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n\Delta g(t)\\ &+ 2c_1\gamma^2 q^2\bar{q}(t)e^{\mathrm{T}}(t)M(t)e(t) + 2c_1 e^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n e(t) \geq 0\end{aligned} \tag{4.15}$$

Also, define $\Theta(t) = [\Theta_1^{\mathrm{T}}(t), \ldots, \Theta_q^{\mathrm{T}}(t)]^{\mathrm{T}}$, where $\Theta_i(t) = \boldsymbol{\theta}_{q+i,\min} - \boldsymbol{\theta}_i(t)$, $i = 1, \ldots, q$. Since $\boldsymbol{\theta}_{q+i,\min} = \boldsymbol{\theta}_{q+j,\min}$ for every $i, j = 1, \ldots, q$, it follows that

$$\begin{aligned}\sum_{i=1}^{q}\sum_{j=1}^{q}||\Theta_i(t) - \Theta_j(t)||^2 &= \sum_{i=1}^{q}\sum_{j=1}^{q}||(\boldsymbol{\theta}_{q+i,\min} - \boldsymbol{\theta}_i(t)) - (\boldsymbol{\theta}_{q+j,\min} - \boldsymbol{\theta}_j(t))||^2\\ &= \sum_{i=1}^{q}\sum_{j=1}^{q}||\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t)||^2\\ &= \sum_{i=1}^{q}\sum_{j=1}^{q}||e_i(t) - e_j(t)||^2\end{aligned} \tag{4.16}$$

The left hand side of (4.16) can be rewritten as $\sum_{i=1}^{q}\sum_{j=1}^{q}||\Theta_i(t)-\Theta_j(t)||^2 = \Theta^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n\Theta(t)$. For the right hand side of (4.16), $\sum_{i=1}^{q}\sum_{j=1}^{q}||\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t)||^2 = e^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n e(t)$. Hence, (4.16) can be rewritten as

$$\Theta^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n\Theta(t) = e^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n e(t)$$

For any positive constant $c_2 > 0$, it follows that

$$c_2\Theta^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n\Theta(t) - c_2e^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n e(t) = 0 \tag{4.17}$$

Since $\Delta G(t) = F_2(t) \otimes I_m \Delta g(t)$ and $\Delta\xi(t) = F_3(t) \otimes I_m\Theta(t)$, by adding (4.15) and (4.17) to the right hand side of (4.14), we have

$$\begin{aligned}
\Delta V(t) \leq & (-c_2 + 2c_1)e^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n e(t) + 2c_1\gamma^2 q^2 \bar{q}(t)e^{\mathrm{T}}(t)M(t)e(t) \\
& + e^{\mathrm{T}}(t)(P \otimes I_q)(F_2(t) \otimes I_n)\Delta g(t) + e^{\mathrm{T}}(t)(P \otimes I_q)(F_3(t) \otimes I_n)\Theta(t) \\
& + \Delta g^{\mathrm{T}}(t)(F_2(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)e(t) \\
& + \Delta g^{\mathrm{T}}(t)(F_2(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(F_2(t) \otimes I_n)\Delta g(t) \\
& - c_1\Delta g^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n\Delta g(t) \\
& + \Delta g^{\mathrm{T}}(t)(F_2(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(F_3(t) \otimes I_n)\Theta(t) \\
& + \Theta^{\mathrm{T}}(t)(F_3(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)e(t) \\
& + \Theta^{\mathrm{T}}(t)(F_3(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(F_2(t) \otimes I_n)\Delta g(t) \\
& + \Theta^{\mathrm{T}}(t)(F_3(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(F_3(t) \otimes I_n)\Theta(t) \\
& + c_2\Theta^{\mathrm{T}}(t)(2qI_q - 2\mathbf{1}_{q\times q}) \otimes I_n\Theta(t)
\end{aligned}$$

Let $\Xi_1(t) = [e^{\mathrm{T}}(t), \Delta g^{\mathrm{T}}(t), \Theta^{\mathrm{T}}(t)]^{\mathrm{T}}$, then by (4.9), we have

$$\Delta V(t) \leq \Xi_1^{\mathrm{T}}(t)\Omega_1(t)\Xi_1(t) \leq -c_3\Xi_1^{\mathrm{T}}(t)\Xi_1(t) \leq -c_3||e(t)||^2 \tag{4.18}$$

where

$$\Omega_1(t) = \begin{bmatrix} \Gamma_{11}(t) & \Gamma_{12}(t) & \Gamma_{13}(t) \\ \Gamma_{12}^{\mathrm{T}}(t) & \Gamma_{22}(t) & \Gamma_{23}(t) \\ \Gamma_{13}^{\mathrm{T}}(t) & \Gamma_{23}^{\mathrm{T}}(t) & \Gamma_{33}(t) \end{bmatrix}$$

By vii) of Theorem 13.11 in [69, p. 785], it follows that (4.18) implies that the discrete-time system (4.10) is (uniformly) geometrically stable, which means that $e(t) \rightrightarrows 0$ as $t \to \infty$. Thus the consensus of the heading angle vectors for the q bats will be reached. □

4.4 Bat-Inspired Consensus with Disturbances

In this section, we extend the previous cooperative learning consensus protocol in the nonlinear case to the scenario where a group of q bats suffer from external disturbances. Specifically, consider the cooperative learning consensus protocol given by the following form

$$\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) + f_i(t)\boldsymbol{u}_i(t) + f_i(t)\mu_i(t)[\boldsymbol{\theta}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)] + \omega_i(t), \quad i = 1, \ldots, q \tag{4.19}$$

where $\omega_i(t) \in \mathbb{R}^n$ is the external disturbance vector for Bat i at time instant t. The other parameters are defined as the same in (4.8) with (4.3)–(4.5) and (4.7).

The objective of this cooperative learning consensus protocol is to reach global consensus and to maintain a desirable disturbance rejection performance. In order to achieve this goal, we first define the variable $e_i(t) = \boldsymbol{\theta}_i(t) - (1/q)\sum_{j=1}^{q} \boldsymbol{\theta}_j(t)$, where $i = 1, 2, \ldots, q$. Next, motivated by [94], we have the following definition regarding the disturbance rejection performance for (4.19).

Definition 2. For a positive scalar γ_1, the cooperative learning consensus protocol (4.19) with (4.3)–(4.5) and (4.7) can *robustly reach consensus with the performance* γ_1 if the following two requirements hold:

1. $\lim_{t\to\infty} e(t) = 0$ for $\omega(t) \equiv 0$,
2. $\limsup_{T\to\infty} \sum_{t=0}^{T} ||e(t)||^2 \le \gamma_1 \limsup_{T\to\infty} \sum_{t=0}^{T} ||\omega(t)||^2$ for all $\omega(t) \not\equiv 0$,

where $e(t) = [e_1^{\mathrm{T}}(t), \ldots, e_q^{\mathrm{T}}(t)]^{\mathrm{T}}$ and $\omega(t) = [\omega_1^{\mathrm{T}}(t), \ldots, \omega_q^{\mathrm{T}}(t)]^{\mathrm{T}}$.

At this stage, we can state the following result that provides a sufficient condition to ensure the two requirements in Definition 2.

Theorem 7. Consider (4.19) with (4.3)–(4.5) and (4.7). Assume that Assumptions 1, 2, and 5 hold. Furthermore, assume that there exist three positive constants $c_1, c_2, c_3 > 0$ and a positive-definite matrix $P = P^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ such that $0 < P < (1 + c_3)I_n$ and

$$\begin{bmatrix} \Gamma_{11}(t) + I_{nq} & \Gamma_{12}(t) & \Gamma_{13}(t) & \Gamma_{14}(t) \\ * & \Gamma_{22}(t) & \Gamma_{23}(t) & \Gamma_{24}(t) \\ * & * & \Gamma_{33}(t) & \Gamma_{34}(t) \\ * & * & * & \Gamma_{44}(t) \end{bmatrix} \le -c_3 I_{4nq} \tag{4.20}$$

where $\Gamma_{14}(t) = (P\otimes I_q)(\mathcal{M}\otimes I_n)$, $\Gamma_{24}(t) = M_2^{\mathrm{T}}(t)(P\otimes I_q)(\mathcal{M}\otimes I_n)$, $\Gamma_{34}(t) = M_3^{\mathrm{T}}(t)(P\otimes I_q)(\mathcal{M}\otimes I_n)$, $\Gamma_{44}(t) = (\mathcal{M}\otimes I_n)^{\mathrm{T}}(P\otimes I_q)(\mathcal{M}\otimes I_n) - \gamma_1^2 I_{nq}$, $\mathcal{M} = I_q - (1/q)\mathbf{1}_{q\times q}$, $\gamma_1 > 0$ is given, and the rest of the symbols are defined as the same in Theorem 6. Then the cooperative learning consensus protocol (4.19) with (4.3)–(4.5) and (4.7) can robustly reach consensus with the performance γ_1^2/γ_2, where $\gamma_2 = \min\{1 + c_3 - \lambda_{\max}(P), \lambda_{\min}(P)\}$, and $\lambda_{\max}(P)$ and $\lambda_{\min}(P)$ denote the maximum and minimum eigenvalues of P, respectively.

Proof. Since $e_i(t) = \boldsymbol{\theta}_i(t) - (1/q)\sum_{j=1}^{q} \boldsymbol{\theta}_j(t) = (1/q)\sum_{j=1}^{q}(\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t))$ for every

$i = 1, \ldots, q$ and every $t \in \overline{\mathbb{Z}}_+$, it follows that for every $i = 1, \ldots, q$,

$$\begin{aligned} e_i(t+1) &= (1/q)\sum_{j=1}^{q}(\boldsymbol{\theta}_i(t+1) - \boldsymbol{\theta}_j(t+1)) \\ &= (1/q)\sum_{j=1}^{q}(\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t)) \\ &\quad + (1/q)\sum_{j=1}^{q}[f_i(t)(g_t^i(\boldsymbol{\theta}(t)) - \boldsymbol{\theta}_i(t)) - f_j(t)(g_t^j(\boldsymbol{\theta}(t)) - \boldsymbol{\theta}_j(t))] \\ &\quad + (1/q)\sum_{j=1}^{q}[f_i(t)\mu_i(t)(\boldsymbol{\theta}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)) - f_j(t)\mu_j(t)(\boldsymbol{\theta}_{j+q,\min}(t) - \boldsymbol{\theta}_j(t))] \\ &\quad + (1/q)\sum_{j=1}^{q}(\omega_i(t) - \omega_j(t)) \end{aligned} \tag{4.21}$$

Let $\Psi_i(t) = (1/q)\sum_{j=1}^{q}(\omega_i(t) - \omega_j(t))$, then it follows from (4.21) that for every $i = 1, \ldots, q$,

$$e_i(t+1) = e_i(t) + \Delta G_i(t) + \Delta\xi_i(t) + \Psi_i(t)$$

where $\Delta G_i(t)$ and $\Delta\xi_i(t)$ were defined in the proof of Theorem 6.

Next, we choose the Lyapunov function candidate $V(t) = \sum_{i=1}^{q} e_i^{\mathrm{T}}(t)Pe_i(t)$, where

$P = P^{\mathrm{T}}$ is a positive-definite matrix. Then it follows that

$$\begin{aligned}
\Delta V(t) =& V(t+1) - V(t) \\
=& \sum_{i=1}^{q} e_i^{\mathrm{T}}(t+1) P e_i(t+1) - \sum_{i=1}^{q} e_i^{\mathrm{T}}(t) P e_i(t) \\
=& \sum_{i=1}^{q} \Big\{ e_i^{\mathrm{T}}(t) P \Delta G_i(t) + e_i^{\mathrm{T}}(t) P \Delta \xi_i(t) + e_i^{\mathrm{T}}(t) P \Psi_i(t) + \Delta G_i^{\mathrm{T}}(t) P e_i(t) \\
& + \Delta G_i^{\mathrm{T}}(t) P \Delta G_i(t) + \Delta G_i^{\mathrm{T}}(t) P \Delta \xi_i(t) \\
& + \Delta G_i(t) P \Delta \Psi_i(t) + \Delta \xi_i^{\mathrm{T}}(t) P e_i(t) \\
& + \Delta \xi_i^{\mathrm{T}}(t) P \Delta G_i(t) + \Delta \xi_i^{\mathrm{T}}(t) P \Delta \xi_i(t) \\
& + \Delta \xi_i^{\mathrm{T}}(t) P \Psi_i(t) + \Psi_i^{\mathrm{T}}(t) P e_i(t) \\
& + \Psi_i^{\mathrm{T}}(t) P \Delta G_i(t) + \Psi_i^{\mathrm{T}}(t) P \Delta \xi(t) + \Psi_i^{\mathrm{T}}(t) P \Psi_i(t) \Big\}
\end{aligned} \tag{4.22}$$

Furthermore, since $\Delta G(t) = F_2(t) \otimes I_n \Delta g(t)$ and $\Delta \xi(t) = F_3(t) \otimes I_n \Theta(t)$, where $F_2(t)$ and $F_3(t)$ were defined in the statement of Theorem 6, and $\Delta g(t)$ and $\Theta(t)$ were defined in the proof of Theorem 6, it follows from (4.22) that

$$
\begin{aligned}
\Delta V(t) =& e^{\mathrm{T}}(t)(P \otimes I_q)(F_2 \otimes I_n)\Delta g(t) + e^{\mathrm{T}}(t)(P \otimes I_q)(F_3(t) \otimes I_n)\Theta(t) \\
&+ e^{\mathrm{T}}(t)(P \otimes I_q)(\mathcal{M} \otimes I_n)\omega(t) + \Delta g^{\mathrm{T}}(t)(F_2(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)e(t) \\
&+ \Delta g^{\mathrm{T}}(t)(F_2(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(F_2(t) \otimes I_n)\Delta g(t) \\
&+ \Delta g^{\mathrm{T}}(t)(F_2(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(F_3(t) \otimes I_n)\Theta(t) \\
&+ \Delta g^{\mathrm{T}}(t)(F_2(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(\mathcal{M} \otimes I_n)\omega(t) \\
&+ \Theta^{\mathrm{T}}(t)(F_3(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)e(t) \\
&+ \Theta^{\mathrm{T}}(t)(F_3(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(F_2(t) \otimes I_n)\Delta g(t) \\
&+ \Theta^{\mathrm{T}}(t)(F_3(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(F_3(t) \otimes I_n)\Theta(t) \\
&+ \Theta^{\mathrm{T}}(t)(F_3(t) \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(\mathcal{M} \otimes I_n)\omega(t) \\
&+ \omega^{\mathrm{T}}(t)(\mathcal{M} \otimes I_n)^{\mathrm{T}}(P \otimes I_q)e(t) \\
&+ \omega^{\mathrm{T}}(t)(\mathcal{M} \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(F_2(t) \otimes I_n)\Delta g(t) \\
&+ \omega^{\mathrm{T}}(t)(\mathcal{M} \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(F_3(t) \otimes I_n)\Theta(t) \\
&+ \omega^{\mathrm{T}}(t)(\mathcal{M} \otimes I_n)^{\mathrm{T}}(P \otimes I_q)(\mathcal{M} \otimes I_n)\omega(t)
\end{aligned}
\tag{4.23}
$$

Next, we define $W(t) = \Delta V(t) + e^{\mathrm{T}}(t)e(t) - \gamma_1^2\omega^{\mathrm{T}}(t)\omega(t)$. Adding (4.15) and (4.17) to both sides of $\Delta V(t)$, it follows from (4.23) and (4.20) that

$$
W(t) \leq \Xi_2^{\mathrm{T}}(t)\Omega_2(t)\Xi_2(t) \leq -c_3\Xi_2^{\mathrm{T}}(t)\Xi_2(t) \leq 0 \tag{4.24}
$$

where $\Xi_2(t) = [e^{\mathrm{T}}(t), \Delta g^{\mathrm{T}}(t), \Theta^{\mathrm{T}}(t), \omega^{\mathrm{T}}(t)]^{\mathrm{T}}$ and

$$\Omega_1(t) = \begin{bmatrix} \Gamma_{11}(t) + I_{nq} & \Gamma_{12}(t) & \Gamma_{13}(t) & \Gamma_{14}(t) \\ \Gamma_{12}^{\mathrm{T}}(t) & \Gamma_{22}(t) & \Gamma_{23}(t) & \Gamma_{24}(t) \\ \Gamma_{13}^{\mathrm{T}}(t) & \Gamma_{23}^{\mathrm{T}}(t) & \Gamma_{33}(t) & \Gamma_{34}(t) \\ \Gamma_{14}^{\mathrm{T}}(t) & \Gamma_{24}^{\mathrm{T}}(t) & \Gamma_{34}^{\mathrm{T}}(t) & \Gamma_{44}(t) \end{bmatrix}$$

If $\omega(t) \equiv 0$, then it follows from (4.24) that $\Delta V(t) \leq -c_3\|e(t)\|^2$, which, by $viii)$ of Theorem 13.11 in [69, p. 785], implies that $e(t) \rightrightarrows 0$ as $t \to \infty$. Hence, the first requirement of Definition 2 is satisfied.

If $\omega(t) \not\equiv 0$, then it follows from (4.24) that $W(t) \leq -c_3\|e(t)\|^2$, which further implies that for any $T \in \overline{\mathbb{Z}}_+$,

$$(1 + c_3)\sum_{t=0}^{T} ||e(t)||^2 + \sum_{t=0}^{T}(V(t+1) - V(t)) \leq \gamma_1^2 \sum_{t=0}^{T} ||\omega(t)||^2$$

or equivalently,

$$(1 + c_3)\sum_{t=0}^{T} ||e(t)||^2 + e^{\mathrm{T}}(T+1)(P \otimes I_q)e(T+1) - e^{\mathrm{T}}(0)(P \otimes I_q)e(0) \leq \gamma_1^2 \sum_{t=0}^{T} ||\omega(t)||^2 \tag{4.25}$$

Since $e^{\mathrm{T}}(T+1)(P \otimes I_q)e(T+1) \geq \lambda_{\min}(P \otimes I_q)\|e(T+1)\|^2 = \lambda_{\min}(P)\|e(T+1)\|^2$ and

$e^{\mathrm{T}}(0)(P \otimes I_q)e(0) \leq \lambda_{\max}(P)\|e(0)\|^2$, it follows from (4.25) that

$$\begin{aligned}\gamma_2 \sum_{t=0}^{T+1} \|e(t)\|^2 &\leq [1 + c_3 - \lambda_{\max}(P)]\|e(0)\|^2 + (1 + c_3) \sum_{t=1}^{T} \|e(t)\|^2 \\ &+ \lambda_{\min}(P)\|e(T+1)\|^2 \\ &\leq (1 + c_3) \sum_{t=0}^{T} ||e(t)||^2 + e^{\mathrm{T}}(T+1)(P \otimes I_q)e(T+1) \\ &- e^{\mathrm{T}}(0)(P \otimes I_q)e(0) \\ &\leq \gamma_1^2 \sum_{t=0}^{T} ||\omega(t)||^2 \\ &\leq \gamma_1^2 \sum_{t=0}^{T+1} ||\omega(t)||^2\end{aligned}$$

which implies that

$$\limsup_{T\to\infty} \sum_{t=0}^{T+1} \|e(t)\|^2 \leq (\gamma_1^2/\gamma_2) \limsup_{T\to\infty} \sum_{t=0}^{T+1} ||\omega(t)||^2$$

Hence, the second requirement of Definition 2 is satisfied. Therefore, the cooperative learning consensus protocol (4.19) with (4.3)–(4.5) and (4.7) can robustly reach consensus with the performance γ_1^2/γ_2. □

4.5 Simulation

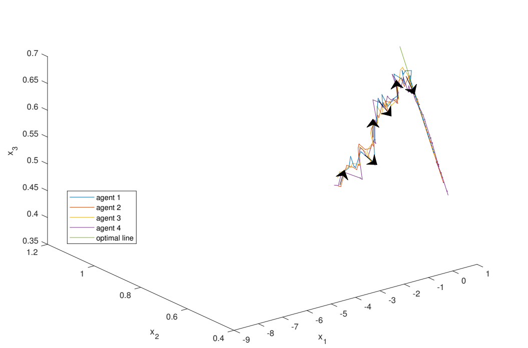
In this part, we will give a simulation example to show the effectiveness of the proposed cooperative learning consensus protocol in the nonlinear case with and without disturbances. The cost functions for the corresponding minimization problems are again $F_1(\boldsymbol{x}) = |x_1 - x_2| + |x_3 - x_1|$ and $F_2(\boldsymbol{x}) = \max\{x_1, x_2, x_3\} - \min\{x_1, x_2, x_3\}$ for $\boldsymbol{x} = [x_1, x_2, x_3]^{\mathrm{T}} \in \mathbb{R}^3$. The number of bats used in the cooperative learning consensus protocol is 4. Its graph topology is strongly connected and includes a ring path, with a Laplacian matrix L_t given by

$$L_t = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}.$$

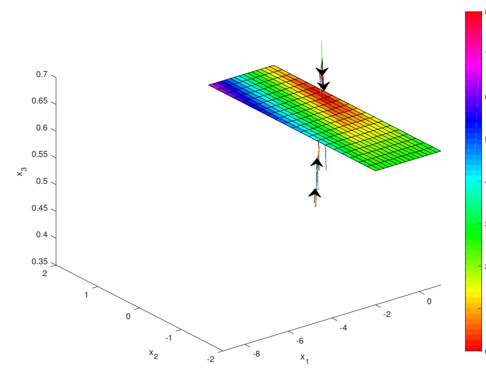
The nonlinear function $\Phi_{ij}(\cdot)$ is chosen as $\Phi_{ij}(x_j(t) - x_i(t)) = (x_j(t) - x_i(t))^3$. The frequency $f_i(t)$ is drawn randomly in $[0.2, 0.7]$ and the zooming number $\mu_i(t)$ is drawn randomly in $[0.3, 0.7]$ for every $i = 1, 2, 3, 4$. Let $\gamma = 1.7$, $\bar{q}(t) = 3 + 2 + 2 + 3 = 10$, $M(t) = \text{diag}([3\ 2\ 2\ 3]) \otimes I_3$. Also, we let $c_1 = 5.6470$, $c_2 = 0.1012$, and $c_3 = 0.4258$. By using the MATLAB LMI toolbox to solve (4.9) in Theorem 6, we can obtain the positive-definite matrix $P = 1.2975 \times I_3$, which guarantees that (4.9) in Theorem 6 holds for all $f_i(t)$ and $\mu_i(t)$. Moreover, it follows from Figures 4.1c and 4.2c that the ratio $\lambda = \max_{i,j} \|g_t^i(\boldsymbol{\theta}) - g_t^j(\boldsymbol{\theta})\| / \|\boldsymbol{\theta}_{ip} - \boldsymbol{\theta}_{jp}\|$ does not exceed 1.5, and hence, Assumption 5 holds for $\gamma = 1.7$.

Figures 4.1a and 4.1b show the convergence of the proposed cooperative learning consensus protocol without disturbances when minimizing $F_1(\boldsymbol{x}) = |x_1 - x_2| + |x_3 - x_1|$, while Figures 4.2a and 4.2b show the convergence of the proposed cooperative learning consensus protocol without disturbances when minimizing $F_2(\boldsymbol{x}) = \max\{x_1, x_2, x_3\} - \min\{x_1, x_2, x_3\}$. The simulation results demonstrate that the proposed cooperative learning consensus protocols approach the optimal value of the function.

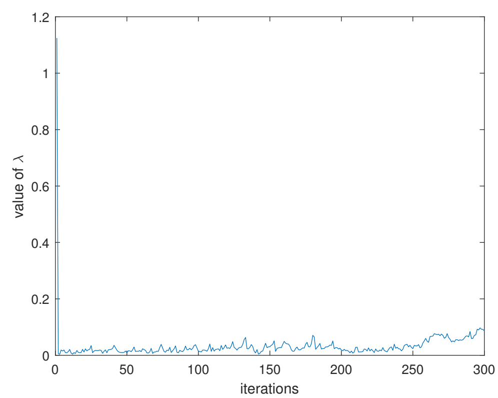
Next, we consider the proposed cooperative learning consensus protocol subject to external disturbances. We chose $\omega(t) = [\omega_1(t), \omega_2(t), \omega_3(t)]^{\mathrm{T}}$, where $\omega_i(t)$ is the random variable whose value is drawn from $[0, 1]$ at each time instant t. Similarly, the parameters $f_i(t)$ and $\mu_i(t)$ are randomly drawn from $[0.2, 0.7]$ and $[0.3, 0.7]$, respectively. Let $\gamma = 1.7$,



(a) Convergence of cooperative learning consensus.



(b) Function value map.

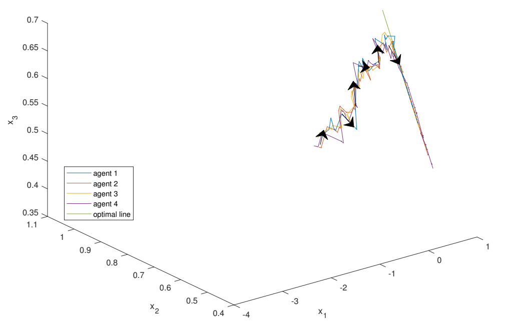


(c) Time history of $\lambda = \max_{i,j} \|g_t^i(\boldsymbol{\theta}) - g_t^j(\boldsymbol{\theta})\| / \|\boldsymbol{\theta}_{ip} - \boldsymbol{\theta}_{jp}\|$.

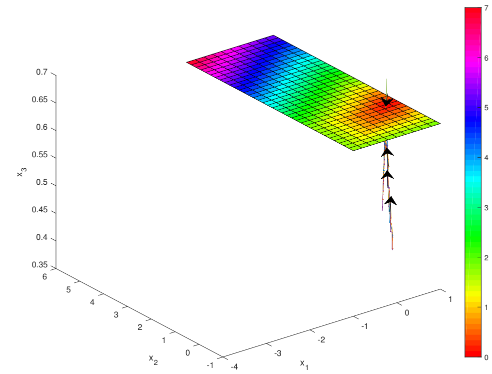
Figure 4.1: Convergence of the cooperative learning consensus protocol when minimizing $F_1(\boldsymbol{x}) = |x_1 - x_2| + |x_3 - x_1|$ without disturbances: Theorem 6.

$\gamma_1 = 2$, $\bar{q}(t) = 10$, and $M(t) = \text{diag}([3\ 2\ 2\ 3]) \otimes I_3$. Furthermore, let $c_1 = 5.0601$, $c_2 = 0.5493$, and $c_3 = 0.8493$. By solving (4.20) in Theorem 7 via the MATLAB LMI toolbox, we can obtain the positive-definite matrix $P = 1.5217 \times I_3 < (1 + c_3)I_3$, which guarantees that (4.20) in Theorem 7 holds for all $f_i(t)$ and $\mu_i(t)$. Moreover, it follows from Figures 4.3c and 4.4c that the ratio $\lambda = \max_{i,j} \|g_t^i(\boldsymbol{\theta}) - g_t^j(\boldsymbol{\theta})\| / \|\boldsymbol{\theta}_{ip} - \boldsymbol{\theta}_{jp}\|$ does not exceed 1.5, and hence, Assumption 5 holds for $\gamma = 1.7$.

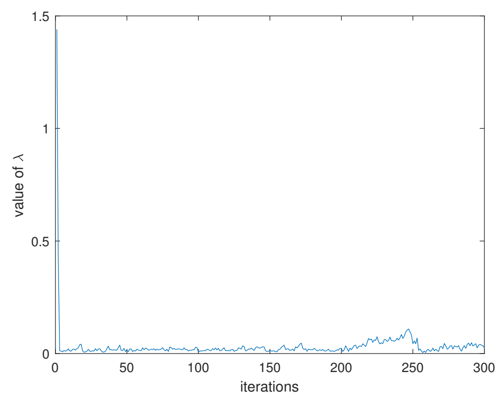
Figures 4.3a and 4.3b show the convergence of the proposed cooperative learning



(a) Convergence of cooperative learning consensus.



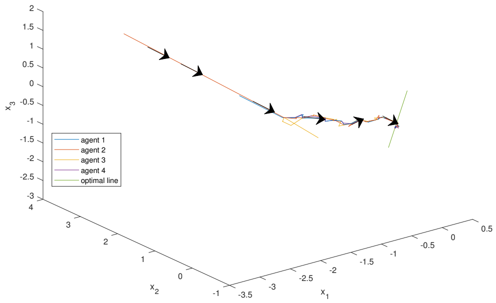
(b) Function value map.



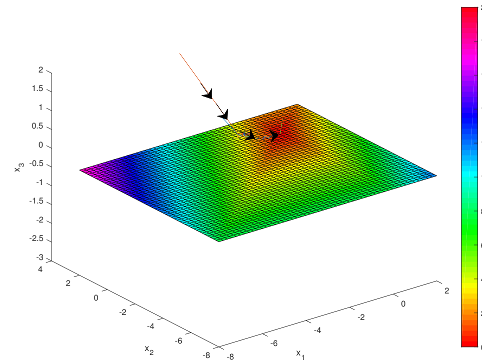
(c) Time history of $\lambda = \max_{i,j} \|g_t^i(\boldsymbol{\theta}) - g_t^j(\boldsymbol{\theta})\| / \|\boldsymbol{\theta}_{ip} - \boldsymbol{\theta}_{jp}\|$.

Figure 4.2: Convergence of the cooperative learning consensus protocol when minimizing $F_2(\boldsymbol{x}) = \max\{x_1, x_2, x_3\} - \min\{x_1, x_2, x_3\}$ without disturbances: Theorem 6.

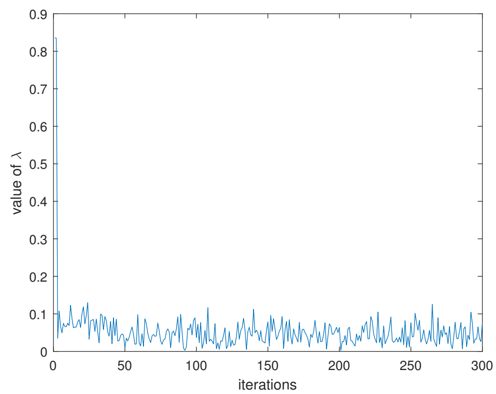
consensus protocol with disturbances when minimizing $F_1(\boldsymbol{x}) = |x_1 - x_2| + |x_3 - x_1|$, while Figures 4.4a and 4.4b show the convergence of the proposed cooperative learning consensus protocol with disturbances when minimizing $F_2(\boldsymbol{x}) = \max\{x_1, x_2, x_3\} - \min\{x_1, x_2, x_3\}$. The simulation results demonstrate that the proposed cooperative learning consensus protocols approach the optimal value of the function.



(a) Convergence of cooperative learning consensus.



(b) Function value map.

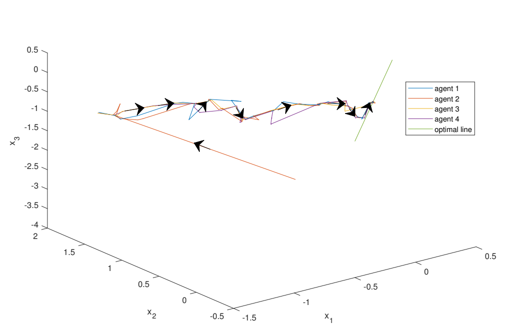


(c) Time history of $\lambda = \max_{i,j} \|g_t^i(\boldsymbol{\theta}) - g_t^j(\boldsymbol{\theta})\| / \|\boldsymbol{\theta}_{ip} - \boldsymbol{\theta}_{jp}\|$.

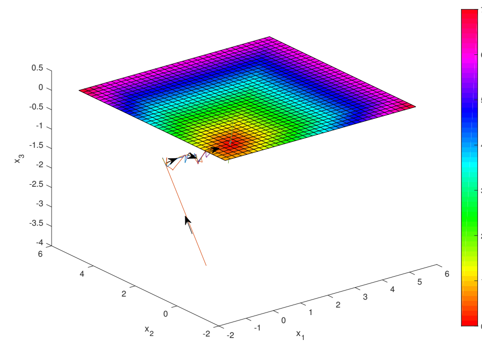
Figure 4.3: Convergence of the cooperative learning consensus protocol when minimizing $F_1(\boldsymbol{x}) = |x_1 - x_2| + |x_3 - x_1|$ with disturbances: Theorem 7.

4.6 Conclusion

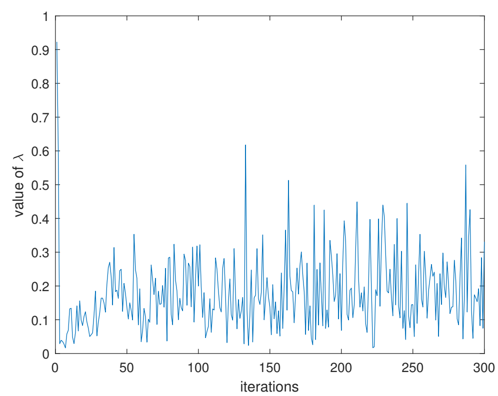
In this chapter, we proposed a class of bat-inspired consensus protocols with nonlinear dynamics based on the bat algorithm and multi-agent coordination optimization. These consensus protocols embed a suggested convergence direction that can improve the performance of their convergence restricted by certain rules via solving an additional optimization problem. A Lyapunov-based method was used to give the sufficient condition which



(a) Convergence of cooperative learning consensus.



(b) Function value map.



(c) Time history of $\lambda = \max_{i,j} \|g_t^i(\boldsymbol{\theta}) - g_t^j(\boldsymbol{\theta})\| / \|\boldsymbol{\theta}_{ip} - \boldsymbol{\theta}_{jp}\|$.

Figure 4.4: Convergence of the cooperative learning consensus protocol when minimizing $F_2(\boldsymbol{x}) = \max\{x_1, x_2, x_3\} - \min\{x_1, x_2, x_3\}$ with disturbances: Theorem 7.

can guarantee the convergence of the proposed consensus protocols. We extended the bat-inspired consensus protocols to the case where they are subjected to external disturbances and also gave the sufficient condition to guarantee their asymptotic convergence under some matrix inequality conditions.

Future work will focus on the relaxation of the semi-Lipschitz condition in Assumption 5 and further discussion on solving some nonsmooth optimization for discontinuous flux functions in the proposed consensus protocol. Also, the application of the proposed consensus protocols in multi-layer, multi-dependent cyber-physical network systems will be interested for further research.

CHAPTER 5
THE BAT-INSPIRED CONSENSUS PROTOCOLS WITH DIFFERENTIAL PRIVACY

5.1 Introduction

While the consensus problem for multi-agent systems [13] has drawn a great attention in recent years in different areas, and is studied extensively from the control-theoretic perspective, it is until recently that some analogy between this problem and swarm intelligence algorithms, such as particle swarm optimization, has been noticed by [33,56]. This similarity has inspired us to improve the performance of swarm intelligence algorithms by modifying them using some techniques from the various consensus protocols in the literature. Such a combination from a control problem and a computational intelligence algorithm offers a brand new perspective to design efficient swarm intelligence algorithms, not just from the bio-inspired direction, but also from the control-theoretic methodology, leading to a one-way exploration from control theory to swarm optimization. Now the question lies in the other direction: *Is it possible to design consensus protocols for multi-agent systems using some techniques or concepts from swarm intelligence, so that the state convergence direction of these systems can be totally guidable but not totally predictable, with agents being "smart" to their data transmission and "sensitive" to their data privacy, rather than being "dumb" and "passive" to these issues?* This is the question we will address in this chapter, and an affirmative answer will be given to this question. Hence, a two-way, positive feedback of mutual exploration and interplay is unraveled between networked control theory and swarm optimization based algorithms, based on the result in this chapter and the results in [33, 56].

We will address the above question in two aspects. First, we show that swarm intelligence can be used for designing new consensus protocols with two additional attributes of agents being "smart" to data transmission and their state convergence direction being totally guidable but not totally predictable. There are many swarm intelligence algorithms existing in the literature. Among these, the bat algorithm (BA) [35] is a recently developed algorithm to solve some unusual optimization problems that are irregular, nonconvex, nonlinear, and time-dependent. This algorithm increases the diversity of the population of candidate solutions to an optimization problem by mimicking the frequency of the bats. In this chapter, motivated by the multi-agent coordination optimization (MCO) algorithm [33, 56], a new bat-inspired consensus protocol is proposed. More specifically, by incorporating a separate, unrelated optimization problem into the protocol, our new consensus algorithm can fully guide its state convergence direction leaning toward the best solution (i.e., the optimal solution among the population of candidate solutions) to this separate, unrelated optimization problem. At the same time, although the optimal solution to this optimization problem may always exist (e.g., convex optimization), its best solution form may not be precisely calculated or numerically found. Hence, such an issue actually creates an uncertainty for exactly predicting the final state convergence direction, which turns out to be a good merit for protecting multi-agent systems from semi-honest and malicious adversaries. Moreover, the proposed consensus algorithm further takes advantage of the mechanism behind the BA algorithm to enhance agents' data transmission capability so that they become "smart" enough to not only process the neighboring and their own data, but also relay the processed data among agents in a multi-hop way.

Generally speaking, a consensus algorithm for a multi-agent system needs each agent to share its own state information to its neighbors in order to achieve a common state for all agents. However, in some cases, agents may not want 100% to share their information to others. This is common in the social area, where persons in rendezvous activity may not want to share their initial location information to others. Another example is when a group of individuals were asked about their opinions on a particular subject, they may not want their own opinions to be known by others but they are curious about others' opinions. Thus, it is worth to consider the privacy preserving when designing a consensus algorithm in this regard. This is the second aspect that we will address in the chapter.

The concept of differential privacy has been studied in database [46] and recently is applied in dynamical systems. In [95], the authors design a differentially private filters for dynamical systems by adding white Gaussian noise to the system. Authors of [48] proposed a differential privacy consensus algorithm, where a Laplacian noise process is considered and added. A type of convergence in the sense of mean squared operation is considered in [45], which can guarantee the privacy of the initial state and make the consensus state converge to its exact initial value. However, both of these two consensus algorithms only consider the average consensus, namely, the consensus algorithm eventually converges to the average of the initial value. This is a total predictable situation for the state convergence direction, and may not be a desired scenario in many problems. For example, the convergence problem in some stochastic optimization algorithms [17] and the temperature equipartition problem in system thermodynamics [96] will not lead to average consensus in general. The agreement algorithm in [17] shows a weighted consensus result for the state convergence direction, which is partially predictable but hard to make these weights directly controllable, and hence, is not totally guidable. In contrast, our proposed consensus protocol is fully guidable but not fully predictable, due to the unique feature that the convergence state of the multi-agent system leans toward the best solution to a separately designated, designer-controllable optimization problem that an adversary cannot predict *a* *priori*.

5.2 Bat-Inspired Consensus and Differential Privacy

The notations used in this chapter is fairly standard. Specifically, $\mathbb{R}$ denotes the set of real number. $\overline{\mathbb{R}}_+$ denotes the set of nonnegative real numbers. $\mathbb{R}^n$ denotes the set of n-dimensional real column vectors. $\mathbb{R}^{n\times m}$ denotes the set of n-by-m matrices. $(\cdot)^{\mathrm{T}}$ denotes transpose and $(\cdot)^{-1}$ denotes inverse. Let $\otimes$ denote the Kronecker product. Furthermore, we use some algebraic graph-relatd notations to describe the bat-inspired consensus protocols. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote an undirected graph with the set of vertices $\mathcal{V} = \{v_1, v_2, \ldots, v_q\}$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, where $\mathcal{E}$ denotes the set of edges. Also, we define the matrix $A \in \mathbb{R}^{q\times q}$ be the adjacency matrix, whose elements a_{ij} associated with the graph $\mathcal{G}$ are nonnegative. A finite set $\mathcal{N} = \{1, 2, \ldots, q\}$ denotes the node index of the $\mathcal{G}$. An edge of the $\mathcal{G}$ is denoted by $e_{ij} = (v_i, v_j)$. If $e_{ij} \in \mathcal{G}$, then $a_{ij} = 1$, otherwise $a_{ij} = 0$, and $a_{ii} = 0$ for all $i \in \mathcal{N}$. For each node i, we denote its neighbors as the set of $\mathcal{N}^i = \{v_j \in \mathcal{N} : (v_i, v_j) \in \mathcal{E},\ j = 1, 2, \ldots, |\mathcal{N}|, j \neq i\}$, where $|\mathcal{N}|$ denotes the cardinality of the set $\mathcal{N}$. The degree matrix of the graph $\mathcal{G}$ is defined as $D = [d_{ij}]_{i,j=1,2,\ldots,|\mathcal{N}|}$, where $d_{ij} = \begin{cases} \sum_{j=1}^{\mathcal{N}} a_{ij} & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$ The Laplacian matrix of the graph $\mathcal{G}$ can be defined by $L = D - A$. For a constant $c > 0$, *Lap*(c) denotes the Laplace distribution with probability density function $\mathbb{P}_c(x) \triangleq \frac{1}{2c}e^{-\frac{|x|}{c}}$, where $|\cdot|$ denotes a norm of a vector. The Laplace distribution has zero mean and the variance is $2c^2$. We can write this as $x \sim$ *Lap*(c). For an n-dimensional random vector, its probability density function is defined as $\mathbb{P}_c(x) = (\frac{1}{2c})^n e^{-\frac{\|x\|_1}{c}}$. We can write this as $x \sim$ *Lap*(c, n). The components of the Laplace random vector are independent and $\|\cdot\|_1$ denotes the L_1 norm.

5.2.1 Bat-Inspired Consensus

In this chapter we consider a group of q bats whose communication graph $\mathcal{G}$ is fixed. Before we present our consensus protocols, we need to make some assumptions.

Assumption 6. There is a Hamiltonian cycle in the communication graph $\mathcal{G}$.

For the Assumption 6, if there is a Hamiltonian cycle in the $\mathcal{G}$, we know that there exists a communication path in the q bats:

$$\text{Bat } 1 \leftrightarrow \text{Bat } 2 \leftrightarrow \cdots \leftrightarrow \text{Bat } q \leftrightarrow \text{Bat } 1 \tag{5.1}$$

Furthermore, we assume that for each Bat i, it is "smart" to access the state information of its neighbors, meanwhile, it can serve as routers to transfer some information to its neighbor.

Assumption 7. For a given optimization problem $\min_{x\in\mathbb{R}^n} F(x)$, the set $\mathcal{X}$ is compact and convex. Let $C_1 \triangleq \sup_{x,y\in\mathcal{X}} \|x-y\|_1$ denote the diameter of $\mathcal{X}$, where $\mathcal{X} \subseteq \mathbb{R}^n$ is the domain of optimization and $\|\cdot\|_1$ denotes the L_1 norm.

The next assumption is about a separate optimization problem embedded in the proposed bat-inspired consensus protocol.

Assumption 8. The optimization problem $\min_{x\in\mathbb{R}^n} F(x)$ has a solution, where $F : \mathbb{R}^n \to \mathbb{R}$.

This assumption is needed since if the optimization problem does not have a solution, the consensus state may not be achieved.

The original bat algorithm was inspired by the echolocation or bio-sonar characteristics of microbats [35]. The agents or the bats can update their state information by the following rules to find their prey. The update rules to solve the optimization problem $\min_{x\in\mathbb{R}^n} F(x)$

are given as follows:

$$\begin{aligned} f_i(t) &= f_{\min} + (f_{\max} - f_{\min})\beta_i(t) \\ v_i(t+1) &= v_i(t) + [x_i(t) - p(t)]f_i(t) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \tag{5.2}$$

where $t \in \overline{\mathbb{Z}}_+ = \{0, 1, 2, \ldots\}$, and $x_i(t)$ and $v_i(t)$ are the position and velocity for each Bat i at each time t, respectively. $f_i(t)$ is the frequency information of each Bat i at time t. $f_{\min}$ and $f_{\max}$ are the lower and upper bound of the frequency, respectively. $\beta_i(t) \in [0, 1]$ is a random number drawn from a uniform distribution, and $p(x)$ is the current best global solution to the optimization problem at time t.

Based on this algorithm, we can develop a bat-inspired consensus protocol, which is fully guidable but not fully predictable. Here we consider that all of the bats have the same constant speed but with different heading angles. By updating their angle information continuously, the heading angles of all bats can achieve a common heading angle asymptotically. The consensus protocol proposed for tacking heading angles of the bats is described as follows [97–99]:

$$\boldsymbol{\theta}_{1,\min}(t) = \boldsymbol{\theta}_1(t) \tag{5.3}$$

$$\begin{aligned} \boldsymbol{\theta}_{k+1,\min} &= \arg\min\{F(\boldsymbol{\theta}_{k,\min}(t)), F(\boldsymbol{\theta}_{k+1}(t))\}, \\ k &= 1, 2, \ldots, 2q-1 \end{aligned} \tag{5.4}$$

$$\boldsymbol{\theta}_{2q,\min}(t) = \boldsymbol{\theta}_{2q-1,\min} \tag{5.5}$$

$$\begin{aligned} \boldsymbol{\theta}_i(t+1) =& \boldsymbol{\theta}_i(t) + f_i(t)\Big\{ \arg \min_{\boldsymbol{\theta}(t)\in\mathbb{R}^n} \sum_{j\in\mathcal{N}^i} \\ & [\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}(t)]^{\mathrm{T}} \Phi_{ij}(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}(t)) - \boldsymbol{\theta}_i(t)\Big\} \\ & + f_i(t)\mu_i(t)[\boldsymbol{\theta}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)] \end{aligned} \tag{5.6}$$

$$f_i(t) = f_{\min} + \beta_i(t)(f_{\max} - f_{\min}) \tag{5.7}$$

where $t \in \overline{\mathbb{Z}}_+, \boldsymbol{\theta}_i = \boldsymbol{\theta}_{q+i} \in \mathbb{R}^n$ denotes the heading angle of Bat i at time t. $\beta_i(t),\ f_{\min}$, and $f_{\max}$ have the same meaning as in the (5.2). $\mu_{\min} < \mu_i(t) < \mu_{\max}$ is the zooming parameter for each Bat i. $\Phi_{ij} : \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued flux function which satisfies $\Phi_{ij}(x) = 0$ if and only if $x = 0$ and $x^T \Phi_{ij} x \geq 0$ for every $x \in \mathbb{R}^n$ and $i, j = 1, \ldots, q,\ i \neq j$. This function can be seen as an energy/mass exchange rate in compartmental modeling [90, 91] or a heat transfer rate in thermodynamics [92]. Compared with the original bat optimization algorithm, our proposed consensus algorithm has a one more term $\arg\min_{\boldsymbol{\theta}(t)\in\mathbb{R}^n} \sum_{j\in\mathcal{N}^i}[\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}(t)]^{\mathrm{T}}\Phi_{ij}(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}(t))$ in (5.6). This interconnected term comes from the speed-up and speed-down strategy and is derived from the biological swarms [100]. Also, it should be noted that the term $\boldsymbol{\theta}_{q,\min}(t)$ in (5.6) is the suggested convergence direction. This term is important for the consensus protocol to achieve convergence and it can be computed through the *multihop communication protocol* [60] of the form (5.3)-(5.5) based on the Hamiltonian path (5.1). This process includes the following two steps:

1) For Bat $k+1$, it can receive the state information $\boldsymbol{\theta}_{k,\min}(t)$ from Bat k at time t, where $k = 1, \ldots, q-1$. Meanwhile, Bat $k+1$ determines $\boldsymbol{\theta}_{k+1,\min}(t) = \arg\min\{F(\boldsymbol{\theta}_{k,\min}(t)), F(\boldsymbol{\theta}_{k+1}(t))\}$ and serves as a router to send this state information $\boldsymbol{\theta}_{k+1,\min}(t)$ to its neighbor in the communication loop path (5.1).

2) After the state information $\boldsymbol{\theta}_{q,\min}(t)$ is determined by Bat q, this state information can be passed to Bat ($q+1$ mod q), which means the state information is passed to Bat 1. The mod denotes the modulo operation. Thus, Bat (k mod q), $k = q+1, \ldots, 2q-1$, can determine $\boldsymbol{\theta}_{k,\min}(t) = \arg\min\{F(\boldsymbol{\theta}_{k-1,\min}(t)), F(\boldsymbol{\theta}_{(k \bmod q)}(t)\}$ and serve as a router to pass the state information $\boldsymbol{\theta}_{k,\min}(t)$ to Bat ($k+1$ mod q) by following the

Hamiltonian path

$$\text{Bat } q \leftrightarrow \text{Bat } (q+1 \text{ mod } q) \leftrightarrow \text{Bat } (q+2 \text{ mod } q)$$
$$\leftrightarrow \cdots \leftrightarrow \text{Bat } (2q-1 \text{ mod } q)$$

which equals

$$\text{Bat } q \leftrightarrow \text{Bat } 1 \leftrightarrow \text{Bat } 2 \leftrightarrow \cdots \leftrightarrow \text{Bat } (q-1).$$

This is the same Hamiltonian cycle compared with (5.1).

One may notice that here we used a "double-check" technique in the second step to acquire the information $\arg\min_{1\leq i\leq q} F(\boldsymbol{\theta}_i(t))$. Specifically, after Step 1, the state information $\boldsymbol{\theta}_{q,\min}(t)$ we obtained in fact is $\arg\min_{1\leq i\leq q} F(\boldsymbol{\theta}_i(t))$. Therefore, in Step 2, the state information $\boldsymbol{\theta}_{q+i,\min}(t)$ we obtained equals $\boldsymbol{\theta}_{q,\min}(t)$ for each bat $i = 1, \ldots, q-1$. Even in this situation, we still perform the comparison operation $\boldsymbol{\theta}_{q+i,\min}(t) = \arg\min\{F(\boldsymbol{\theta}_{q+i-1,\min}(t)), F(\boldsymbol{\theta}_i(t))\}$ in Step 2. This operation can ensure that there is no major error for the state information $\arg\min_{1\leq i\leq q} F(\boldsymbol{\theta}_i(t))$ we acquired.

Another important point of this consensus protocol is about the term $\Phi_{ij}(\cdot)$. We need to determine the concrete form of the function $\Phi_{ij}(\cdot)$. According to [91, 92], the most commonly used form is the linear form:

$$\Phi_{ij}(\boldsymbol{x}) = \boldsymbol{x} \tag{5.8}$$

There are also some other forms such as the signum form $\Phi_{ij}(\boldsymbol{x}) = \text{sgn}(\boldsymbol{x})$, where $\text{sgn}(\cdot)$ denotes the signum function. In this chapter, we only consider the linear form of the function

$\Phi_{ij}(\cdot)$ in (5.8) for simplicity. Thus, we have:

$$\begin{aligned}
&\arg\min_{\boldsymbol{\theta}\in\mathbb{R}^n} \sum_{j\in\mathcal{N}^i} [\boldsymbol{\theta}_j(t)-\boldsymbol{\theta}]^{\mathrm{T}}\Phi_{ij}(\boldsymbol{\theta}_j(t)-\boldsymbol{\theta}) \\
&= \arg\min \sum_{j\in\mathcal{N}^i} \|\boldsymbol{\theta}_j(t)-\boldsymbol{\theta}\|^2 \\
&= \frac{1}{\mathcal{N}^i}\sum_{j\in\mathcal{N}^i} \boldsymbol{\theta}_j(t),\ i=1,\ldots,q
\end{aligned} \tag{5.9}$$

where $\|\cdot\|$ denotes the L_2 norm. Hence, (5.6) can be written as:

$$\begin{aligned}
\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) + \frac{f_i(t)}{|\mathcal{N}^i|}\sum_{j\in\mathcal{N}^i}[\boldsymbol{\theta}_j(t)-\boldsymbol{\theta}_i(t)] \\
+ f_i(t)\mu_i(t)[\boldsymbol{\theta}_{q,\min}(t)-\boldsymbol{\theta}_i(t)]
\end{aligned} \tag{5.10}$$

for each $i=1,\ldots,q$, since $\boldsymbol{\theta}_{q+i,\min}(t)=\boldsymbol{\theta}_{q,\min}(t)=\arg\min_{1\le i\le q} F(\boldsymbol{\theta}_i(t))$.

5.2.2 Differential Privacy

In this part, we introduce the notation of differential privacy. First, define $\boldsymbol{y}_i(t)\in\mathbb{R}^n$ as the observation information Bat i sent to its neighboring bats at time instant t. The value of $\boldsymbol{y}_i(t)$ can be computed as the current state information of Bat i plus a noise vector $\boldsymbol{w}_i(t)$ whose elements are drawn independently from the Laplace distribution *Lap*(M,n), where M is a parameter that we will define later. Meanwhile, we define $\boldsymbol{z}_i(t)=\frac{1}{|\mathcal{N}^i|}\sum_{j\in\mathcal{N}^i}\boldsymbol{y}_i(t)$ as the received information for each Bat i from its neighboring bats. Thus, the new bat-inspired consensus protocol is proposed as follows:

$$\boldsymbol{y}_{1,\min}(t)=\boldsymbol{y}_1(t)=\boldsymbol{\theta}_1(t)+\boldsymbol{w}_1(t) \tag{5.11}$$

$$\boldsymbol{y}_{k+1,\min} = \arg\min\{F(\boldsymbol{y}_{k,\min}(t)), F(\boldsymbol{y}_{k+1}(t))\}, \quad k = 1, 2, \ldots, q \tag{5.12}$$

$$\boldsymbol{y}_{2q,\min}(t) = \boldsymbol{y}_{2q-1,\min}(t), \tag{5.13}$$

$$\begin{aligned}\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) + \frac{f_i(t)}{|\mathcal{N}^i|}\sum_{j\in\mathcal{N}^i}[\boldsymbol{y}_j(t) - \boldsymbol{\theta}_i(t)] \\ + f_i(t)\mu_i(t)[\boldsymbol{y}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)],\end{aligned} \tag{5.14}$$

$$f_i(t) = f_{\min} + \beta_i(t)(f_{\max} - f_{\min}),\ i = 1, \ldots, q. \tag{5.15}$$

All the other parameters are the same as the ones in (5.3)-(5.7). Also, (5.14) can be rewritten as:

$$\begin{aligned}\boldsymbol{\theta}_i(t+1) = (1 - f_i(t))\boldsymbol{\theta}_i(t) + f_i(t)\boldsymbol{z}_i(t) \\ + f_i(t)\mu_i(t)[\boldsymbol{y}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)]\end{aligned} \tag{5.16}$$

The definition of differential privacy presented here is similar to the notion introduced in [101] for the streaming algorithms. Let $\boldsymbol{\Theta}(t) = [\boldsymbol{\theta}_1^{\mathrm{T}}(t), \ldots, \boldsymbol{\theta}_q^{\mathrm{T}}(t)]^{\mathrm{T}}$, $\boldsymbol{y}(t) = [\boldsymbol{y}_1^{\mathrm{T}}(t), \boldsymbol{y}_2^{\mathrm{T}}(t), \ldots, \boldsymbol{y}_q^{\mathrm{T}}(t)]^{\mathrm{T}}$, and $\boldsymbol{z}(t) = [\boldsymbol{z}_1^{\mathrm{T}}(t), \boldsymbol{z}_2^{\mathrm{T}}(t), \ldots, \boldsymbol{z}_q^{\mathrm{T}}(t)]^{\mathrm{T}}$. We define an execution of the bat inspired consensus algorithm (5.11)-(5.15) is a possibly infinite sequence of the form $\alpha = \boldsymbol{\Theta}(0), \langle \boldsymbol{y}(0), \boldsymbol{z}(0)\rangle, \boldsymbol{\Theta}(1), \langle \boldsymbol{y}(1), \boldsymbol{z}(1)\rangle, \ldots$. The observation part of this execution is the corresponding infinite sequence $\boldsymbol{y}(0), \boldsymbol{y}(1), \ldots$. Thus, we can define the observation mapping of the execution as $\mathcal{R}(\alpha) \triangleq \{\boldsymbol{y}(0), \boldsymbol{y}(1), \ldots\}$. This sequence gives us the exchanged information for the corrsponding execution α. Suppose there exists an adversary who can access to the whole communication channels, and thus, he/she can observe all of the information that these bats send to each other, namely, $\boldsymbol{y}_i(t)$ will be known to him/her for each Bat i. If given optimization problem $\min_{x\in\mathbb{R}^n} F(x)$, the observation sequence of information Y, and the initial state information $\boldsymbol{\Theta}(0)$, then $\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \boldsymbol{\Theta}(0))$ is the set of execution α that can generate the observation Y.

Definition 3. The two vectors $\boldsymbol{\Theta}(t),\ \boldsymbol{\Theta}'(t) \subseteq \mathbb{R}^{nq}$ are δ-*adjacent* if there exists one $i \in \mathcal{N}$ such that $\|\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}'_i(t)\| \leq \delta$ and for all $j \neq i$, $\boldsymbol{\theta}_j(t) = \boldsymbol{\theta}'_j(t)$, where $\delta \geq 0$.

The Definition 3 is about the adjacent, then we can define the differential privacy based on Definition 3.

Definition 4. For an $\epsilon > 0$, the bat-inspired consensus is ϵ-*differential privacy*, if for any two adjacent initial state information $\Theta(0)$ and $\Theta'(0)$, any set of observation sequences Y, and the optimization problem $\min_{x\in\mathbb{R}^n} F(x)$, the following condition is satisfied:

$$\begin{aligned} &\mathbb{P}[\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \boldsymbol{\Theta}(0))] \leq \\ &\qquad e^{\epsilon}\mathbb{P}[\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \boldsymbol{\Theta}'(0))] \end{aligned} \tag{5.17}$$

where the probability is taken over the coin-flips of the algorithm.

Generally speaking, the notation of ϵ-differential privacy ensures that the consensus algorithm for multi-agent systems keeps the privacy for themselves, which means for an adversary who can acess to all of the observation sequences would not infer the state information for each individual bat with a significant probability. Also, the smaller ϵ generally means a higher privacy level of the consensus algorithm.

Definition 5. At each time instant t, for any optimization problem $\min_{x\in\mathbb{R}^n} F(x)$, the observation sequences Y, and two adjacent initial state information $\boldsymbol{\Theta}(0)$, $\boldsymbol{\Theta}'(0)$, define $x \in \mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \boldsymbol{\Theta}(0))$ and $x' \in \mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \boldsymbol{\Theta}'(0))$, the *sensitivity* of the bat-inspired consensus can be defined as:

$$\Delta(t) \triangleq \sup\|x - x'\|_1. \tag{5.18}$$

To estimate the bound of $\Delta(t)$, we fix any observation sequence Y, the optimization problem $\min_{x\in\mathbb{R}^n} F(x)$, and the initial adjacent vector $\boldsymbol{\Theta}(0)$ and $\boldsymbol{\Theta}'(0)$, the corresponding execution can be defined as $\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n}, Y, \boldsymbol{\Theta}(0)) = \boldsymbol{\Theta}(0), \langle \boldsymbol{y}(0), \boldsymbol{z}(0)\rangle, \boldsymbol{\Theta}(1), \langle \boldsymbol{y}(1), \boldsymbol{z}(1)\rangle, \ldots$ and $\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n}, Y, \boldsymbol{\Theta}'(0)) = \boldsymbol{\Theta}'(0), \langle \boldsymbol{y}'(0), \boldsymbol{z}'(0)\rangle, \boldsymbol{\Theta}'(1), \langle \boldsymbol{y}'(1), \boldsymbol{z}'(1)\rangle$, $\ldots$, since the observation sequence Y is identical for both executions $\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n}, Y, \boldsymbol{\Theta}(0))$ and $\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n}, Y, \boldsymbol{\Theta}'(0))$, which means that $\boldsymbol{y}(t) = \boldsymbol{y}'(t)$ for all time instant t. From the definition of $\boldsymbol{z}_i(t)$, $\boldsymbol{z}_i(t) = \frac{1}{|\mathcal{N}^i|}\sum_{j\in\mathcal{N}^i} \boldsymbol{y}_i(t)$, we can know $\boldsymbol{z}(t) = \boldsymbol{z}'(t)$ for all time instant t. Hence, the sensitivity of the bat-inspired consensus can be computed as follows:

$$\begin{aligned}\Delta(t) =& \|\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \boldsymbol{\Theta}(0)) \\ & - \mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \boldsymbol{\Theta}'(0))\|_1 \\ &= \|\boldsymbol{\Theta}(t) - \boldsymbol{\Theta}'(t)\|_1 \\ &= \|\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}'_i(t)\|_1,\end{aligned} \tag{5.19}$$

From Assumption 7, we have

$$\Delta(t) \leq C_1. \tag{5.20}$$

Then, we can define the parameter M in *Lap*(M, n) as:

$$M = C_1/(aq^t) \tag{5.21}$$

where $a > 0$ is a constant and $q \in (0, 1)$.

5.3 Convergence Analysis

In this section, we want to show that the proposed bat-inspired consensus algorithm converges and to establish the bound of the accuracy of this consensus algorithm. First, we need to define the convergence and accuracy of the proposed consensus algorithm.

Definition 6. The proposed bat-inspired consensus is *convergent* if for any Bats $i, j \in \mathcal{N}$,

$$\lim_{t\to\infty} \mathbb{E}\|\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t)\| = 0, \tag{5.22}$$

where the expectation is taken over the coin-flip of the consensus algorithm.

We define the solution to the optimization problem $\min_{x\in\mathbb{R}^n} F(x)$ by $\boldsymbol{\theta}^*$, and the average of the state information of the individual bat by $\overline{\boldsymbol{\theta}}(t) \triangleq \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} \boldsymbol{\theta}_i(t)$. The accuracy of the consensus can be defined as:

Definition 7. For a constant $d \geq 0$, the proposed bat-inspired consensus is d-*accurate* if

$$\lim_{t\to\infty} \mathbb{E}\|\overline{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}^*\| \leq d, \tag{5.23}$$

where the expectation is the same as in Definition 6.

It should be noted that the smaller constant d means the consensus algorithm is more accurate. If the consensus algorithm finds the exact solution of the optimization problem $\min_{x\in\mathbb{R}^n} F(x)$, the constant d becomes 0. That is to say that the consensus has 0-accurate.

Then we can show that the proposed consensus algorithm converges, but we first need to show that the proposed consensus satisfies ϵ-differential privacy.

Lemma 10. The proposed consensus algorithm (5.11)-(5.16) is ϵ-differential privacy if $\frac{a}{1-q} \leq \epsilon$.

Proof. For an arbitrary time instant t, we fix any optimization problem $\min_{x\in\mathbb{R}^n} F(x)$, the observation sequences Y, and any adjacent initial state information $\boldsymbol{\Theta}(0)$ and $\boldsymbol{\Theta}'(0)$. Since the communication graph is fixed, the observation information $\boldsymbol{y}(t)$ at each time instant t is fixed. According to the definition of received information, $\boldsymbol{z}_i(t) = \frac{1}{|\mathcal{N}^i|}\sum_{j\in\mathcal{N}^i} \boldsymbol{y}_i(t)$ is fixed. Then, by (5.16), we know that the state information $\boldsymbol{\theta}_i(t)$ is uniquely determined. Next, we define a bijection $f : \mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \mathbf{\Theta}(0)) \to \mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \mathbf{\Theta}'(0))$. Define $\alpha \in \mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \mathbf{\Theta}(0))$ and $\alpha' \in \mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \mathbf{\Theta}'(0))$. $f(\alpha) = \alpha'$ if and only if they have the same observation sequence, which means that $\mathcal{R}(\alpha) = \mathcal{R}(\alpha')$. If we fix the observation sequence Y, then there exists an unique execution $\alpha \in \mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \mathbf{\Theta}(0))$ that can generate this observation sequence Y . Also, there exists a unique execution $\alpha' \in \mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n} F(x), Y, \mathbf{\Theta}'(0))$. Now, we relate the probability measures of the two executions α and α':

$$\frac{\mathbb{P}[\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n}, Y, \mathbf{\Theta}(0))]}{\mathbb{P}[\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n}, Y, \mathbf{\Theta}'(0))]} = \frac{\int_\alpha \mathbb{P}[\alpha]d\mu}{\int_{\alpha'} \mathbb{P}[\alpha']d\mu'}, \tag{5.24}$$

and

$$\int_{\alpha'} \mathbb{P}[\alpha']d\mu' = \int_{f(\alpha)} \mathbb{P}[f(\alpha)]d\mu = \int_{\alpha} \mathbb{P}[f(\alpha)]d\mu \tag{5.25}$$

if we change the variable in the integral.

The probability comes from the noise $\boldsymbol{w}_i(t)$ according to (6.11). We denote the kth component of $\boldsymbol{\theta}_i(t)$ as $\theta_i^k(t)$. Since $\boldsymbol{y}_i(t)$ can be obtained by adding n-dimensional random noise drawn from *Lap*(M, n). Each component of the noise is independent from others. Thus,

$$\mathbb{P}[\alpha] = \prod_{i\in\mathcal{N}, k\in n} \mathbb{P}_M(y_i^k(t) - \theta_i^k(t)). \tag{5.26}$$

Then, we relate the distance between α and $f(\alpha)$ with the sensitivity of the consensus $\Delta(t)$ defined in Definition 5. Thus, from (5.20) we can obtain

$$\|\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_i'(t)\|_1 = \Delta(t) \le C_1. \tag{5.27}$$

Since the function f is a bijection, the observations of α and $f(\alpha)$ are matched, which means that $\boldsymbol{y}(t) = \boldsymbol{y}'(t)$. According to the definiton of Laplacian distribution and (5.21), we have

$$
\begin{aligned}
&\prod_{i\in\mathcal{N},k\in n}\frac{\mathbb{P}_M(y_i^k(t)-\theta_i^k(t))}{\mathbb{P}_M(y_i'^k(t)-\theta_i'^k(t))}\\
&\leq \prod_{i\in\mathcal{N},k\in n}\exp\left(\frac{|y_i^k(t)-\theta_i^k(t)-y_i'^k(t)+\theta_i'^k(t)|}{M}\right)\\
&=\prod_{i\in\mathcal{N},k\in n}\exp\left(\frac{\theta_i^k(t)-\theta_i'^k(t)}{M}\right)\\
&=\exp\left(\sum_{i\in\mathcal{N},k\in n}\frac{|\theta_i^k(t)-\theta_i'^k(t)|}{M}\right)\\
&\leq \exp\left(\frac{\Delta(t)}{M}\right)\leq \exp\left(\frac{C_1}{M}\right)=e^{aq^t},
\end{aligned}
\tag{5.28}
$$

where $\exp$ denotes the exponential function. Then, by (5.24), (5.25), (5.26), and (5.28), we can obtain

$$
\begin{aligned}
&\frac{\mathbb{P}[\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n},Y,\mathbf{\Theta}(0))]}{\mathbb{P}[\mathcal{R}^{-1}(\min_{x\in\mathbb{R}^n},Y,\mathbf{\Theta}'(0))]}=\frac{\int_\alpha\mathbb{P}[\alpha]d\mu}{\int_\alpha\mathbb{P}[f(\alpha)]d\mu}\\
&\leq\frac{\int_\alpha e^{\sum_{t=0}^{\infty}aq^t}\mathbb{P}[f(\alpha)]d\mu}{\int_\alpha\mathbb{P}[f(\alpha)]d\mu}\\
&\leq e^{\sum_{t=0}^{\infty}aq^t}=e^{\frac{a}{1-q}}\leq e^{\epsilon}.
\end{aligned}
\tag{5.29}
$$

Thus, the consensus algorithm is ϵ-differential privacy by Definition 4. □

Next, in order to show that the proposed bat-inspired consensus algorithm is convergent, we need to define some matrices first. Let $R(t)=\text{diag}\{f_1(t),f_2(t),\dots,f_q(t)\}$, $U(t)=\text{diag}\{\mu_1(t),\mu_2(t),\dots,\mu_q(t)\}$,

$$A^{[j]}(t)=R(t)U(t)(I_q-\mathbf{1}_{q\times 1}\otimes E_{1\times q}^{[j]})+R(t)D^{-1}L, \tag{5.30}$$

and

$$B^{[j]}(t)=R(t)D^{-1}A+\mathbf{1}_{q\times 1}\otimes E_{1\times q}^{[j]} \tag{5.31}$$

where diag denotes the diagonal matrix, $\mathbf{1}_{q\times 1}$ denotes the vector of q elements and each element is one, and $E_{n\times nq}^{[j]}\in\mathbb{R}^{n\times nq}$ denotes a block-elements whose jth block-column is I_n and the rest of the block-matrices are all zero matrices, for example, $E^{[j]}_{n\times nq} = [\mathbf{0}_{n\times n},\ldots,\mathbf{0}_{n\times n}, I_q, \mathbf{0}_{n\times n},\ldots,\mathbf{0}_{n\times n}]$, $j\in\{1,2,\ldots,q\}$.

By defining these notations, the consensus protocol (5.16) can be rewritten as:

$$\mathbf{\Theta}(t+1) = (I_q - A^{[j]}(t))\otimes I_n\mathbf{\Theta}(t) + B^{[j]}(t)\otimes I_n\boldsymbol{w}(t), \tag{5.32}$$

where $\boldsymbol{w}(t) = [\boldsymbol{w}_1^{\mathrm{T}}(t), \boldsymbol{w}_2^{\mathrm{T}}(t),\ldots,\boldsymbol{w}_q^{\mathrm{T}}(t)]^{\mathrm{T}}$ is the noise vector.

We have the following result regarding the convergence of (5.32), or equivalently, (5.11)-(5.16).

Theorem 8. Consider the consensus algorithm (5.11)-(5.16), assume that for every $t\in\overline{\mathbb{Z}}_+$ and every $j\in\mathcal{N}$, there exist two constants λ_m and λ_M such that the following two conditions hold:

C1) $\lambda_m I_n < A^{[j]}(t) < \lambda_M I_n$,

C2) $2\lambda_m > \lambda_M^2$.

Then, the consensus algorithm is convergent.

Proof. We first define a function $P(t) = \frac{1}{2}\sum_{i,j\in\mathcal{N}}[\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t)]^2$. Thus, $P(t) = \mathbf{\Theta}^{\mathrm{T}}(t)L\otimes$

$I_n\mathbf{\Theta}(t)$. By (5.32), we have:

$$\begin{aligned}
P(t+1) &= \mathbf{\Theta}(t+1)^{\mathrm{T}} L \otimes I_n \mathbf{\Theta}(t+1) \\
&= ((I_q - A^{[j]}(t)) \otimes I_n \mathbf{\Theta}(t) + B^{[j]}(t) \otimes I_n \boldsymbol{w}(t))^{\mathrm{T}} L \otimes I_n \\
&((I_q - A^{[j]}(t)) \otimes I_n \mathbf{\Theta}(t) + B^{[j]}(t) \otimes I_n \boldsymbol{w}(t)) \\
&= ((I_q - A^{[j]}(t)) \otimes I_n \mathbf{\Theta}(t))^{\mathrm{T}} L \otimes I_n \\
&((I_q - A^{[j]}(t)) \otimes I_n \mathbf{\Theta}(t)) + 2((I_q - A^{[j]}(t)) \otimes I_n \mathbf{\Theta}(t))^{\mathrm{T}} \\
&L \otimes I_n (B^{[j]}(t) \otimes I_n \boldsymbol{w}(t)) + (B^{[j]}(t) \otimes I_n \boldsymbol{w}(t))^{\mathrm{T}} \\
&L \otimes I_n (B^{[j]}(t) \otimes I_n \boldsymbol{w}(t)),
\end{aligned} \tag{5.33}$$

then we take the expectation of both sides of (5.33) with respect to the coin-flip of the consensus algorithm:

$$\begin{aligned}
\mathbb{E}\|P(t+1)\| &= \mathbb{E}\|((I_q - A^{[j]}(t)) \otimes I_n \mathbf{\Theta}(t))^{\mathrm{T}} L \otimes I_n \\
&((I_q - A^{[j]}(t)) \otimes I_n \mathbf{\Theta}(t))\| + \mathbb{E}\|2((I_q - A^{[j]}(t)) \otimes I_n \\
&\mathbf{\Theta}(t))^{\mathrm{T}} L \otimes I_n (B^{[j]}(t) \otimes I_n \boldsymbol{w}(t))\| \\
&+ \mathbb{E}\|(B^{[j]}(t) \otimes I_n \boldsymbol{w}(t))^{\mathrm{T}} L \otimes I_n (B^{[j]}(t) \otimes I_n \boldsymbol{w}(t))\|
\end{aligned} \tag{5.34}$$

Let $T(t) = (B^{[j]}(t) \otimes I_n \boldsymbol{w}(t))^{\mathrm{T}} L \otimes I_n (B^{[j]}(t) \otimes I_n \boldsymbol{w}(t))$. Since $\boldsymbol{w}(t)$ and $\Theta(t)$ are indepdent, (5.34) becomes:

$$\begin{aligned}
\mathbb{E}\|P(t+1)\| &= \mathbb{E}\|((I_q - A^{[j]}(t)) \otimes I_n \mathbf{\Theta}(t))^{\mathrm{T}} L \otimes I_n \\
&((I_q - A^{[j]}(t)) \otimes I_n \mathbf{\Theta}(t))\| + \mathbb{E}\|T(t)\| \\
&= \mathbb{E}\|\mathbf{\Theta}(t)^{\mathrm{T}} L \otimes I_n \mathbf{\Theta}(t) - 2\mathbf{\Theta}(t)^{\mathrm{T}} (A^{[j]}(t) \otimes I_n)^{\mathrm{T}} \\
&L \otimes I_n \mathbf{\Theta}(t) + \mathbf{\Theta}(t)^{\mathrm{T}} (A^{[j]}(t) \otimes I_n)^{\mathrm{T}} L \otimes I_n \\
&(A^{[j]}(t) \otimes I_n) \mathbf{\Theta}(t)\| + \mathbb{E}\|T(t)\|
\end{aligned} \tag{5.35}$$

Let $Q(t) = 2\mathbf{\Theta}(t)^{\mathrm{T}}(A^{[j]}(t) \otimes I_n)^{\mathrm{T}} L \otimes I_n \mathbf{\Theta}(t) - \mathbf{\Theta}(t)^{\mathrm{T}}(A^{[j]}(t) \otimes I_n)^{\mathrm{T}} L \otimes I_n (A^{[j]}(t) \otimes I_n)\mathbf{\Theta}(t)$, then, it follows:

$$\begin{aligned} Q(t) &\geq 2\lambda_m \mathbf{\Theta}(t)^{\mathrm{T}} L \otimes I_n \mathbf{\Theta}(t) - \lambda_M^2 \mathbf{\Theta}(t)^{\mathrm{T}} L \otimes I_n \mathbf{\Theta}(t) \\ &= (2\lambda_m - \lambda_M^2)\mathbf{\Theta}(t)^{\mathrm{T}} L \otimes I_n \mathbf{\Theta}(t) \end{aligned} \tag{5.36}$$

Thus, (5.35) can be written as:

$$\begin{aligned} &\mathbb{E}\|P(t+1)\| = \mathbb{E}\|\mathbf{\Theta}(t)^{\mathrm{T}} L \otimes I_n \mathbf{\Theta}(t)\| \\ &- \mathbb{E}\|Q(t)\| + \mathbb{E}\|T(t)\| \\ &\leq \mathbb{E}\|\mathbf{\Theta}(t)^{\mathrm{T}} L \otimes I_n \mathbf{\Theta}(t)\| \\ &- (2\lambda_m - \lambda_M^2)\mathbb{E}\|\mathbf{\Theta}(t)^{\mathrm{T}} L \otimes I_n \mathbf{\Theta}(t)\| + \mathbb{E}\|T(t)\|, \end{aligned} \tag{5.37}$$

then, for any $a \leq \min((2\lambda_m - \lambda_M^2), 1)$, we have $Q(t) \geq a\mathbf{\Theta}(t)^T L \otimes I_n \mathbf{\Theta}(t)$. Also, from Condition C2), we know that $2\lambda_m - \lambda_M^2 > 0$. Thus, for some $a \in (0, 1)$, (5.37) becomes:

$$\mathbb{E}\|P(t+1)\| \leq (1-a)\mathbb{E}\|P(t)\| + \mathbb{E}\|T(t)\|. \tag{5.38}$$

Since $\boldsymbol{w}_i(t)$ and $\boldsymbol{w}_j(t)$ are independent for $i \neq j$, and each element of $\boldsymbol{w}_i(t)$ is drawn independently from *Lap*(M, n). Then $\mathbb{E}\|\boldsymbol{w}_i(t)\boldsymbol{w}_j(t)\| = 0$ for $i \neq j$, and if $i = j$, then $\mathbb{E}\|\boldsymbol{w}_i(t)\boldsymbol{w}_j(t)\| = \mathbb{E}\|\boldsymbol{w}_i(t)^2\| = \mathrm{Var}(\boldsymbol{w}_i(t)) = 2a^2q^{2t}$, where Var denotes the finite variance. Thus $\mathbb{E}\|T(t)\|$ converges to 0 if $t \to \infty$. The first term of the right hand side of (5.38) also converges to 0, thus $\lim_{t\to\infty} \mathbb{E}\|P(t)\| = \lim_{t\to\infty} \sum_{i,j\in\mathcal{N}} \mathbb{E}\|\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t)\| = 0$. From Definition 6, the consensus algorithm is convergent. □

Theorem 8 shows that all bats will eventually converge to a common value. Next, we establish a bound of the accuracy of the proposed consensus algorithm. It should be noted that here we use weighted average state instead of average state information defined in

Definition 7, where the weighted average state information is define as $\bar{\boldsymbol{\theta}}_i(t) = \frac{\sum_{i=1}^{q} \gamma_i(t)\boldsymbol{\theta}_i(t)}{\sum_{i=1}^{q} \gamma_i(t)}$ and $\gamma_i(t) = f_i(t)/|\mathcal{N}^i|$.

Theorem 9. The proposed bat-inspired consensus algorithm can guarantee d-accuracy, i.e.,

$$\lim_{t\to\infty} \mathbb{E}\|\bar{\boldsymbol{\theta}}_i(t) - \boldsymbol{\theta}^*\| \le d,$$

where $d = C_1 e^{-f_{\min}\mu_{\min}} + \frac{f_{\max}\mu_{\max}C_1}{f_{\min}\mu_{\min}}$.

Proof. Equation (5.14) can be rewritten as:

$$\begin{aligned} \boldsymbol{\theta}_i(t-1) - \boldsymbol{\theta}^* = \boldsymbol{\theta}_i(t) - \boldsymbol{\theta}^* + \frac{f_i(t)}{|\mathcal{N}^i|} \sum_{j\in\mathcal{N}^i} [\boldsymbol{y}_j(t) - \boldsymbol{\theta}_i(t)] \\ + f_i(t)\mu_i(t)[\boldsymbol{y}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)], \end{aligned} \tag{5.39}$$

where $\boldsymbol{\theta}^*$ is the solution to the optimization problem $\min_{x\in\mathbb{R}^n} F(x)$. Let $\gamma_i(t) = f_i(t)/|\mathcal{N}^i|$. Equation (5.39) can be reformulated as:

$$\begin{aligned} \gamma_i(t)[\boldsymbol{\theta}_i(t+1) - \boldsymbol{\theta}^*] &= \gamma_i(t)[\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}^*] + \sum_{j\in\mathcal{N}^i} \boldsymbol{y}_j(t) \\ &- |\mathcal{N}^i|\boldsymbol{\theta}_i(t) + \gamma_i(t)f_i(t)\mu_i(t)[\boldsymbol{y}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)] \\ &= \gamma_i(t)[\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}^*] + \sum_{j\in\mathcal{N}^i} \boldsymbol{\theta}_j(t) - |\mathcal{N}^i|\boldsymbol{\theta}_i(t) \\ &+ \gamma_i(t)f_i(t)\mu_i(t)[\boldsymbol{\theta}_{q+i,\min}(t) - \boldsymbol{\theta}_i(t)] + \sum_{j\in\mathcal{N}^i} \boldsymbol{w}_j(t) \\ &+ \boldsymbol{w}_k(t) \end{aligned} \tag{5.40}$$

where $\boldsymbol{w}_k(t)$ denotes the noise vector corrsponding to the $\boldsymbol{y}_{q+i,\min}(t)$. Letting $\boldsymbol{\eta}_i(t) =$

$\sum_{j\in\mathcal{N}^i}\boldsymbol{w}_j(t)+\boldsymbol{w}_k(t)$ and adding up all the q equations, we can obtain:

$$\begin{aligned}\sum_{i=1}^{q}\gamma_i(t)[\boldsymbol{\theta}_i(t+1)-\boldsymbol{\theta}^*] &= \sum_{i=1}^{q}\gamma_i(t)[\boldsymbol{\theta}_i(t)-\boldsymbol{\theta}^*]\\ &+\sum_{i=1}^{q}\gamma_i(t)f_i(t)\mu_i(t)[\boldsymbol{\theta}_{q+i,\min}(t)-\boldsymbol{\theta}_i(t)]+\sum_{i=1}^{q}\boldsymbol{\eta}_i(t)\end{aligned}\tag{5.41}$$

Dividing both sides of (5.41) by $\sum_{i=1}^{q}\gamma_i(t)$, we have:

$$\begin{aligned}\frac{\sum_{i=1}^{q}\gamma_i(t)[\boldsymbol{\theta}_i(t+1)-\boldsymbol{\theta}^*]}{\sum_{i=1}^{q}\gamma_i(t)} &= \frac{\sum_{i=1}^{q}\gamma_i(t)[\boldsymbol{\theta}_i(t)-\boldsymbol{\theta}^*]}{\sum_{i=1}^{q}\gamma_i(t)}\\ &+\frac{\sum_{i=1}^{q}\gamma_i(t)f_i(t)\mu_i(t)[\boldsymbol{\theta}_{q+i,\min}(t)-\boldsymbol{\theta}_i(t)]}{\sum_{i=1}^{q}\gamma_i(t)}+\frac{\sum_{i=1}^{q}\boldsymbol{\eta}_i(t)}{\sum_{i=1}^{q}\gamma_i(t)}\\ &=\frac{\sum_{i=1}^{q}\gamma_i(t)[\boldsymbol{\theta}_i(t)-\boldsymbol{\theta}^*]}{\sum_{i=1}^{q}\gamma_i(t)}+\frac{\sum_{i=1}^{q}\boldsymbol{\eta}_i(t)}{\sum_{i=1}^{q}\gamma_i(t)}\\ &+\frac{\sum_{i=1}^{q}\gamma_i(t)f_i(t)\mu_i(t)[\boldsymbol{\theta}_{q+i,\min}(t)-\boldsymbol{\theta}^*]}{\sum_{i=1}^{q}\gamma_i(t)}\\ &-\frac{\sum_{i=1}^{q}\gamma_i(t)f_i(t)\mu_i(t)[\boldsymbol{\theta}_i(t)-\boldsymbol{\theta}^*]}{\sum_{i=1}^{q}\gamma_i(t)}\end{aligned}\tag{5.42}$$

Next, we take the L_2 norm on both sides of (5.42). By Assumption 7, it follows that:

$$\begin{aligned}\|\bar{\boldsymbol{\theta}}_i(t+1)-\boldsymbol{\theta}^*\| &\le \|\bar{\boldsymbol{\theta}}_i(t)-\boldsymbol{\theta}^*\|+f_{\max}\mu_{\max}C_1\\ &-f_{\min}\mu_{\min}\|\bar{\boldsymbol{\theta}}_i(t)-\boldsymbol{\theta}^*\|+\left\|\frac{\sum_{i=1}^{q}\boldsymbol{\eta}_i(t)}{\sum_{i=1}^{q}\gamma_i(t)}\right\|\\ &=(1-f_{\min}\mu_{\min})\|\bar{\boldsymbol{\theta}}_i(t)-\boldsymbol{\theta}^*\|\\ &+f_{\max}\mu_{\max}C_1+\left\|\frac{\sum_{i=1}^{q}\boldsymbol{\eta}_i(t)}{\sum_{i=1}^{q}\gamma_i(t)}\right\|\end{aligned}\tag{5.43}$$

where $\bar{\boldsymbol{\theta}}_i(t) = \frac{\sum_{i=1}^{q} \gamma_i(t)\boldsymbol{\theta}_i(t)}{\sum_{i=1}^{q} \gamma_i(t)}$. Recursively repeating this process, it follows that:

$$\begin{aligned} &\|\bar{\boldsymbol{\theta}}_i(t+1) - \boldsymbol{\theta}^*\| \leq (1 - f_{\min}\mu_{\min})\|\bar{\boldsymbol{\theta}}_i(0) - \boldsymbol{\theta}^*\| \\ &+ \sum_{k=0}^{t}(1 - f_{\min}\mu_{\min})^k (f_{\max}\mu_{\max}C_1) \\ &+ \sum_{k=0}^{t}(1 - f_{\min}\mu_{\min})^k \left(\left\| \frac{\sum_{i=1}^{q} \boldsymbol{\eta}_i(t)}{\sum_{i=1}^{q} \gamma_i(t)} \right\| \right). \end{aligned} \tag{5.44}$$

Next, by Assumption 7, Inequality (5.44) can be written as:

$$\begin{aligned} &\|\bar{\boldsymbol{\theta}}_i(t+1) - \boldsymbol{\theta}^*\| \leq (1 - f_{\min}\mu_{\min})C_1 \\ &+ \frac{f_{\max}\mu_{\max}C_1}{f_{\min}\mu_{\min}}(1 - (1 - f_{\min}\mu_{\min})^t) \\ &+ \sum_{k=0}^{t}(1 - f_{\min}\mu_{\min})^k \left(\left\| \frac{\sum_{i=1}^{q} \boldsymbol{\eta}_i(t)}{\sum_{i=1}^{q} \gamma_i(t)} \right\| \right), \end{aligned} \tag{5.45}$$

then, we take the expectation on both sides of (5.44), since $\boldsymbol{w}_i(t)$ and $\boldsymbol{w}_j(t)$ are indepedent and $\mathbb{E}\|\boldsymbol{w}_i(t)\| = 0$, it follows that $\mathbb{E}\|\sum_{i=1}^{q} \boldsymbol{\eta}_i(t)\| = 0$. Thus, by using the inequality $1 - a \leq e^{-a}$, we have:

$$\lim_{t\to\infty} \mathbb{E}\|\bar{\boldsymbol{\theta}}_i(t+1) - \boldsymbol{\theta}^*\| \leq C_1 e^{-f_{\min}\mu_{\min}} + \frac{f_{\max}\mu_{\max}C_1}{f_{\min}\mu_{\min}}. \tag{5.46}$$

Consequently, the conclusion of the theorem is established. □

5.4 Simulation

To show the effectiveness of the proposed bat-inspired consensus algorithm, we consider an optimization problem $\min F(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$ for $x = [x_1, x_2, x_3]^T \in \mathbb{R}^3$. The global minimum for this optimization problem is 0 at $x_i = 1$. The number of the agents is 4 and their communication graph is connected. The corresponding Laplacian matrix is shown as:

$$L = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}.$$

The parameter $f_{\min}$ and $f_{\max}$ are 0.01 and 0.07, respectively and the zooming parameters $\mu_{\min}$ and $\mu_{\max}$ are 0.3 and 0.8, respectively. For Laplacian distribution, we choose $q = 0.8$ and $a = 0.4$. Fig.5.1, Fig.5.2, and Fig.5.3 show x_1, x_2, and x_3 of all of the agents versus iteration numbers. From the simulation results, we can observe that the consensus state can be reached by this algorithm. The optimization value is 0.03278 at point $x = [1.0777, 1.1624, 1.3528]^T$ in this simulation instead of the exact optimization point, i.e., $x^* = [1, 1, 1]^T$. This can be explained from two aspects. The first aspect is that the BA optimization method does not find the true optimization point. It can only obtain the best solution among the population of candidiate solutions. Hence, the proposed bat-inspired consensus algorithm inherents this property, leading to full guidability but not full predictability. The second aspect is that we use the differentially privacy method in our consensus algorithm. Even though this can preserve the privacy, it does not guarantee its convergence to the exact optimization point. However, this can be tolorated when we concern more about the privacy, especially in the case where the accurate number d is small.

5.5 Conclusion

In this chapter, we proposed a bat-inspired consensus algorithm with privacy-awareness for multi-agent systems. By embedding a separate optimization problem, this consensus algorithm can guide the search direction that is not fully predictable. In order to prevent

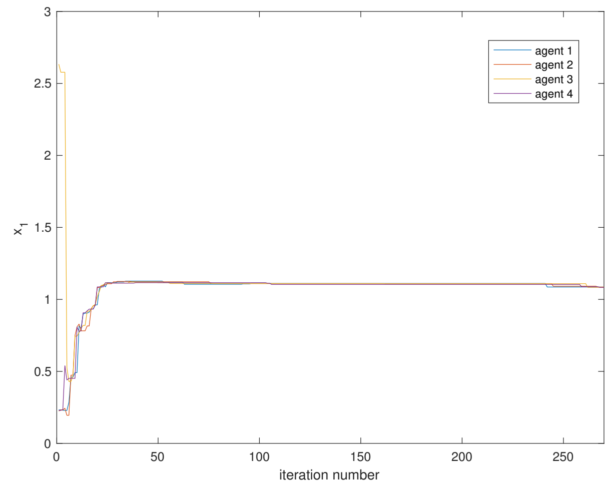


Figure 5.1: x_1 of all agents versus time for the differential privacy consensus algorithm.

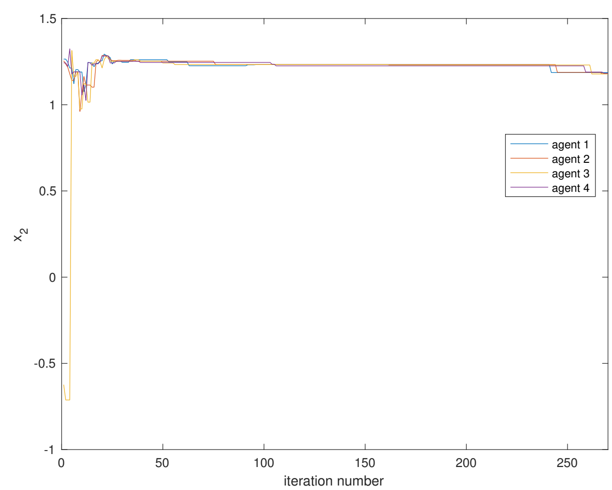


Figure 5.2: x_2 of all agents versus time for the differential privacy consensus algorithm.

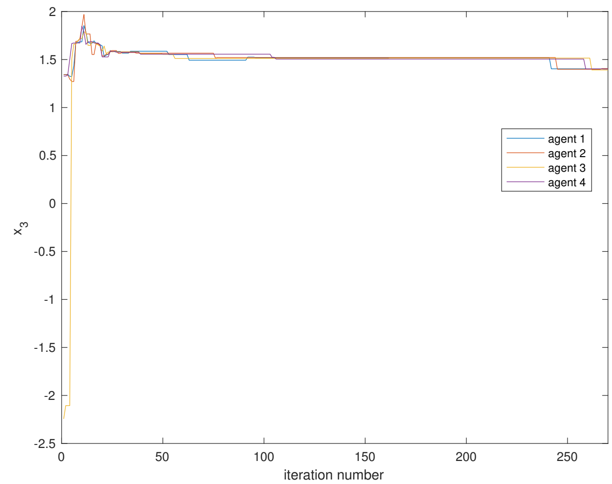


Figure 5.3: x_3 of all agents versus time for the differential privacy consensus algorithm.

semi-honest adversaries from inferring the state information of each agent, we introduced the notation of differentially privacy. We showed that the proposed bat-inspired consensus algorithm satisfies the differential privacy and the consensus algorithm is convergent. Also, we derived the upper bound of the accuracy of the proposed consensus algorithm.

CHAPTER 6
THE FORMATION CONTROL OF MULTI-AGENT WITH DEEP REINFORCEMENT LEARNING

6.1 Introduction

Cooperative control of multi-agent systems (MASs) has attracted many research interest from control and robotics communities in recent years [102]. The application of this task has a wide range in reconnaissance, surveillance, and security [103, 104]. The ability of maintain the network topology and connectivity of robots is crucial for some tasks such as target localization, oceanic search and recur, and undersea oil pipeline maintainance [105–107].

Among cooperative control of MASs, the formation control is one of the most interesting research topic since it has broad applications. Many MASs including unmanned aerial vehicles (UAVs), autonomous underwater vehicles(AUVs), and nonholonomic mobile robots are studied to address the formation control problem. These studies focus on leader-follower methods [108–110], virtual leader approaches [111, 112], and leaderless consensus method [113]. Some other results of the formation control can be found in the survey [114]. The aim of the formation control of the MASs is to design the appropriate algorithm such that it can ensure the group of the agents to achieve and maintain the desired geometric connection of their states. Formation control generally makes the autonomous agents work together to collaboratively finish the formation task. This work is generally accomplished by

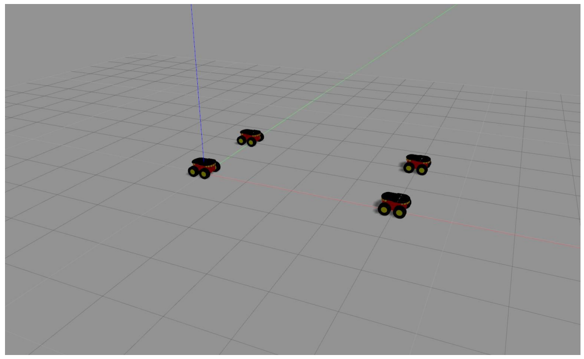


Figure 6.1: The formation control of four robots forming a rectangle.

communicating the state information of each agent with its neighbors. The leader-follower approach is one the most popular method since its simplicity. The basic idea is that the leader can be designed as the reference such that other agents can be controlled to follow the corresponding trajectory, meanwhile, the trajectory of the follower agent is designed to the desired separation and relative bearing with the leader. Another popular method is the consensus algorithm, which focuses on finding a common state for all of the agents, then driving each agent to the particular state relative to the founded common state. Based on this idea, the research on the consensus problems for multi-agent can be extended to the case of directed topology [115].

Many studies use the mobile robot systems to perform the formation control algorithm. For example, the vision-based control method is used to drive the mobile robots to form the desired formation [116]. The obstacle avoidance problems is tackled in the formation control for the mobile robots [117]. A real-time observer developed to estimate the relative state of the mobile robot to form the formation is proposed in [118].

One of the main difficult problem in the formation control is the collision avoidance in the moving of the mobile robot. The major strategies for this problem are the rule-based approaches and optimization-based approaches. For the rule-based approaches, a consensus based algorithm is proposed in [119], where the artificial potential approach is used to generate the collision avoidance strategy. By exploiting the properties of weighted graphs, the formation and the collision avoidance for the robots can be achieved in [120]. Correspondingly, one of the optimization approaches is to use the model predictive control based method [121, 122]. Another problem is that in practice, many robot system models are nonlinear and have nonholonomic constraints, which means the dynamic equation of the robots are hard to model. Some studies focus on the formation control task for nonholonomic mobile robots [123, 124]. The tracking control of the mobile robots with limited information of the desired trajectory is studied in [125].

In order to overcome these two problems, we try to find an efficient way by introducing reinforcement learning (RL) to handle them. In the control engineering domain, RL bridges the gap between the traditional control theory and the adaptive control algorithm [126]. The reinforcement learning is based on the idea of an agent can solve the different actions from learning the outcome which is optimal for some specific situation. Since RL is an end-to-end learning method, there is no need to know the model of the robot, therefore, the designer of the control algorithm can save a lot of effort since only the feedback in the form of the reward function needs to be provided. This reward function generally provides the information of the state and action about the performance of the last step the agent takes [127]. Thus, the agent can learn the appropriate policy to optimize the long-term reward by continuely interact with the environment. For the RL, an agent can evaluate the feedback of its action in each of its step, thus the whole performance of the agent can be improved for the subsequent actions [128, 129].

In this work, the deep reinforcement learning (DRL) is used to model the formation control problem. DRL has been proven to learn control policy directly from the input in [130]. Also, they present that DRL can receive the high-dimensional sensory inputs to learn the policy to excel for some challenging tasks [51]. A major progress for DRL is that it can be extends to the continuous action domain [131]. From then, using DRL to solve some classical control problems becomes easily. Some DRL algorithms in continuous action domain are proposed such as asynchronous advantage actor-critic [132] and in [133], they combine the asymmetric actor-critic with domain randomization. In this chapter, we will use an multi-agent actor-critic algorithm to train our multi robots in the simulation, and use the trained network to the real robots to demonstrate the effectiveness of our algorithm. The main advantages of our algorithm over traditional formation control are that firstly, it avoids the complex model of the nonholonomic mobile robot, hence we avoid designing the input specifically, and secondly, we combine an collision avoidance method in this formation control algorithm.

6.2 Background

In this section, we will introduce some basic knowledge and methods related to our work.

6.2.1 Formation Control

Firstly, we formulate the formation control of multi robots under the general problem setup. Consider the following N agents:

$$\begin{aligned} \dot{p}_i &= q_i \\ \dot{q}_i &= a_i, \end{aligned} \tag{6.1}$$

where p_i and q_i denote the position and velocity of each robot i, $i \in \{1 \ldots, N\}$. They are the state of the agent $s_i = [p_i \; q_i]^{\mathrm{T}}$. a_i is the control input for each robot i. It can be seen as the action of each robot i. Let $F : \mathbb{R}^{2N} \to \mathbb{R}^M$ be given. The desired formation for the agent can be specified by the constraint:

$$F(s) = F(s^*). \tag{6.2}$$

Then, the formation control problem can be defined as:

Definition 8. [114] (Formation Control Problem): The formation control problem can be defined as to design a control law such that the constraint (6.2) can be satisfied for the multi robots systems (6.1).

Based on the Definition 8, some commonly used formation control problems are shown as follows:

- Position-based problem: Each robot i can sense the absolute state of others with respect to a global coordinate system. The constraint (6.2) can be given by:

$$F(s) := s = F(s^*). \tag{6.3}$$

  Each agent can control s_i actively.

- Displacement-based problem: Each agent can sense the relative state of other agents with respect to a global coordinate system. Meanwhile, they cannot sense the absolute state of other agents with the global coordinate system. The constraint (6.2) can be given as:

$$F(s) := [\ldots, (s_j - s_i)^{\mathrm{T}}, \ldots]^{\mathrm{T}} = F(s^*), \tag{6.4}$$

  for each $i, j = 1, \ldots, N$. The constraint (6.4) is invariant to translation applied to the state s. Agent controls $[\ldots, (s_j - s_i)^{\mathrm{T}}, \ldots]^{\mathrm{T}}$ in the displacement-based problem.

- Distance-based problem: Distance-based problem requires that each agent can only sense the relative state of other agents with respect to the local coordinate system. They do not need to sense the absolute state information of other agents with respect to the global coordinate system. The constraint is given as:

$$F(s) := [\ldots, \|s_j - s_i\|, \ldots]^{\mathrm{T}} = F(s^*), \tag{6.5}$$

for each robot $i, j = 1, \ldots, N$. The constraint (6.5) is invariant to combination of translation and rotation applied to state s. Each agent actively controls $[\ldots, \|s_j - s_i\|]^{\mathrm{T}}$. The differece between distance-based and the displacement-based problem is that the distance-based problem only cares the relative distance between the two agents, however, the displacement-based problem cares the relative coordinate for each two agents.

In this work, we perform the position-based formation control, which means the multi robots system (6.1) will achieve $F(s) \rightarrow F(s^*)$. The constraint of the system will depend on the (6.3), which describes the desired position for each robot with respect to the global coordinate system, thus each robot will move to a designed state with respect to the common state. The specific form of the constraints in (6.3) will be discussed in the experiment part.

6.2.2 Markov Decision Processes

In this part, we will introduce the Markov decision processes (MDP) for multi-agent. The multi-agent MDP can be defined as a set of state $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$ which describes the possible configurations of all the agents, a set of action $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_N\}$ which describes the actions of each agent i, and a set of observations $\mathcal{O} = \{\mathcal{O}_1, \ldots, \mathcal{O}_N\}$ for each agent. In order to choose proper action for each agent, it uses a stochastic policy $\pi_{\theta_i} : \mathcal{O}_i \times \mathcal{A}_i \rightarrow [0,\ 1]$ to generate the next action it should take. By executing this action, the agent can produce the next state according to the state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_N \rightarrow \mathcal{S}$. After each agent executing its own action, the agent can acquire the rewards as a function of the state and the action $r_i : \mathcal{S} \times \mathcal{A}_i \rightarrow \mathbb{R}$, meanwhile, the agent can obtain a private observation related with the state $o_i : \mathcal{S} \rightarrow \mathcal{O}_i$. The solution of the multi-agent MDP is a control policy π_{θ_i} which can maximize the expected sum of future rewards $R_i = \sum_{t=0}^{T} \gamma^t r_i^t$ if the agent executes the policy, where γ is a discount factor and T is the time horizon.

6.2.3 Deep Q-Networks

Deep Q-Networks (DQN) is a popular method in RL and has already been proven a successful algorithm for the multi-agent scenario [134, 135]. Q-learning use the action-value function to evaluate the policy it learns. The corresponding action-value function can be described as $Q^{\pi}(s,a) = \mathbb{E}[R|s^t = s, a^t = a]$. This function can be computed recursively as $Q^{\pi}(s,a) = \mathbb{E}_{s'}[r(s,a) + \gamma \mathbb{E}_{a' \sim \pi}[Q^{\pi}(s',a')]]$. DQN can learn the optimal action-value function Q^* corresponding to the optimal policy of the agent by minimizing the function given as follows:

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s'}[(Q^*(s,a|\theta) - y)^2], \tag{6.6}$$

this function generally is referred as the loss function, where $y = r + \gamma \max_{a'} \bar{Q}^*(s',a')$. $\bar{Q}$ is the target Q function and its parameters are updated with the most recent paramenters in DQN periodically. DQN also uses experience replay buffer $\mathcal{D}$ to stabilizing the network, which is the tuples containing (s,a,r,s').

The multi-agent can perform the DQN directly by letting each agent i learn its own independently optimal function Q_i [136]. The disadvantage of this algorithm is that the environment may not maintain stationary for each agent i when each agent learns its optimal policy independently. In this case, the Markov assumption will be violated.

6.2.4 Policy Gradient Algorithms

Avoiding learning the action-value function Q, another algorithm was proposed to learn the policy directly, which is a popular choice for the DQN. The policy can be learned by adjust the parameters θ of the policy by maximizing the objective function $J(\theta) = \mathbb{E}_{s\sim p^{\pi}, a\sim\pi_{\theta}}[R]$. This can be achieved by taking the steps in the gradient direction $\nabla_{\theta} J(\theta)$. The gradient of the policy can be given as follows:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim p^\pi, a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)], \tag{6.7}$$

where p^π is the state distribution. The multi-agent scenario can use policy gradient algorithm to exhibit the high variance gradient estimates. In such case, the reward of each agent obtained depends on the actions of other agents. The agent's own reward will exhibit more variability when compared with the single agent action. Based on policy gradient algorithm, many other method are developed by learning the approximation of the true action-value function $Q^\pi(s,a)$ such as temporal-difference learning and actor-critic algorithm [137].

The policy gradient algorithm can be extended to the deterministic policies $\mu_\theta : \mathcal{S} \to \mathcal{A}$, which can be used to solve the problem where the action of each agent taken will be in the continuous domain. Thus, we can rewrite the gradient of the objective function $J(\theta) = \mathbb{E}_{s\sim p^\mu}[R(s,a)]$ as follows:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim\mathcal{D}}[\nabla_\theta \mu_\theta(a|s) \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}], \tag{6.8}$$

The term $\nabla_a Q^\mu(s,a)$ requires that the action doamin $\mathcal{A}$ should be continuous and hence the policy each agent acting also should be in the continuous domain.

The deep deterministic policy gradient (DDPG) is a variant of the deterministic policy method, where the policy μ and the critic Q^μ can be approximated by using the deep neural networks. DDPG still uses the replay buffer of experiences to sample data and trains the corresponding network. Also, it uses the target network to avoid the divergence, which is the same as in the DQN. The DDPG can learn complex policies for some tasks using low-dimensional observations. It adopts the straightforward actor-critic algorithm and makes the learning process becoming easily implemented for some difficult problem and large networks.

6.3 Algorithm

This part is to derive the algorithm that can perform the multi-agent cooperative work for the DRL. Consider now a system of N robots operating in a common environment. There is no controller can collect the rewards or make the decisions for the whole robots, which means each agent in the communication network can only decide its own action and thus receive the corresponding rewards for itself. The communication network can be denoted by $G = (N, E)$, where E represents the set of communication links, which means E is a undirected graph with the vertex set N and edge set $E \subseteq \{(i,j) : i,j \in N, i \neq j\}$. We assume that each agent can fully observe its state, i.e., $s_i = o_i$. The edge $(i,j) \in E$ denotes that the agent i and j can communicate with each other. Meanwhile, each agent can only use the local state information to learn its policies. By augmenting the policies information of other agents, we can use the extended actor-critic method to solve the formation control problem of multi robots.

For N robots, the policies of the system can be parameterized by $\theta = \{\theta_1, \ldots, \theta_N\}$. We denote $\pi = \{\pi_1, \ldots, \pi_N\}$ be the polices of all the robots. Thus, we can rewrite the gradient of the expected return for each agent i. It can be given as follows:

$$\begin{aligned} &\nabla_{\theta_i} J(\theta_i) \\ &= \mathbb{E}_{s\sim\mu, a_i\sim\pi_i}[\nabla_{\theta_i} \log \pi_i(a_i|s_i) Q_i^{\pi}(s, a_1, \ldots, a_N)], \end{aligned} \tag{6.9}$$

From the action-value function $Q_i^{\pi}(s, a_1, \ldots, a_N)$ used in (6.9), we can observe that it requires the actions of all of the robots as the input. The output is the Q value for the agent i. Each agent i learns the Q value separately, since the action-value function for each agent has its own structure, we can define different constraints (6.2) for each agent i.

Now we consider each robot will determine its own policies μ_{θ_i} in continuous domain with respect to the parameters θ_i. The gradient can be rewritten as follows:

$$\begin{aligned}&\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{x,a\sim\mathcal{D}}\\&[\nabla_{\theta_i}\mu_i(a_i|s_i)\nabla_{a_i}Q_i^{\mu}(s, a_1, \ldots, a_N)|_{a_i=\mu_i(s_i)}],\end{aligned} \tag{6.10}$$

The replay buffer $\mathcal{D}$ constitutes the tuples$(s, s', a_1, \ldots, a_N, r_1, \ldots, r_N)$. This tuple can contain the actions of all agent and hence all the rewards after they execute the action. Hence, the loss function can be given as follows:

$$\mathcal{L}(\theta_i) = \mathbb{E}_{s,a,r,s'}[(Q_i^{\mu}(s, a_1, \ldots, a_N) - y)^2], \tag{6.11}$$

where $y = r_i + \gamma Q_i^{\mu'}(s', a'_1, \ldots, a'_N)|_{a'_j=\mu'_j(s_j)}$, $\mu' = \{\mu_{\theta'_1,\ldots,\theta'_N}\}$ is the set of policies all the agent learned in target network with parameter θ'_i.

We can see that this algorithm still needs the assumption that the agent i needs to know other agent's policies. This assumption is commonly used in the Velocity Obstacle (VO) method, which is used to solve the collision avoidance problem. In order to remove this assumption, each robot i can estimate the policies $\hat{\mu}_i^j$ of other agent j takes. This estimation can be achieved by maximizing the log probability of agent j's reward and it can be given as follows:

$$\mathcal{L}(\hat{\mu}_i^j) = -\mathbb{E}_{s_j,a_j}[\log \hat{\mu}_i^j(a_j|s_j) + \lambda H(\hat{\mu}_i^j)], \tag{6.12}$$

where H is the entropy of the policy distribution. Then the corresponding estimated $\hat{y}$ can be calculated by the following:

$$\hat{y} = r_i = \gamma Q_i^{\mu'}(s', \hat{\mu}_i'^1(s_1)\ldots, \hat{\mu}_i'^N(s_N)), \tag{6.13}$$

where $\hat{\mu}_i'^j$ denotes the policies generated by the target network for the approximate policy

$\hat{\mu}_i^j$. When training the network, the action-value function Q_i^μ can be updated and the latest samples of each agent j in replay buffer can be used to perform the single gradient step to update the parameter of the critic network. Thus, we can remove the assumption that each agent needs to know other's policies. The whole algorithm is shown in Algorithm 1.

Algorithm 1 Formation control of multi robots with Deep Reinforcement Learning

```
for episode = 1 to M do
    Reveive the inital state s for each robots.
    for t = 1 to max iteration number do
        for each robot i selects action a_i = μ_θi(o_i) + N_t with respect to its policy.
        each robot executes the action a_i.
        each robot receives the reward r_i and moves to the next state s_i by the system
        dynamic (6.1).
        store (s, a, r, s') to the replay buffer D.
        for robot i = 1 to N do
            s_i ← s'_i for each robot i.
            smaple a random minibatch of S samples (s^j, a^j, r^j, s'^j) from replay buffer D.
            set ŷ_j by (6.13).
            update critic by minimizing loss function (6.12).
            update actor by sampled policy gradient by (6.10).
        end for
    end for
end for
```

6.4 Experiment

In this part, we will demonstrate our experiments in details. We first use the gazebo simulation environment to train the DRL and get the network model, then we use the trained algorithm in the real robot to show the effectiveness of the algorithm. The robot systems we used here are 4 Pioneer 3dx robot. It is a differential driven robot which has two active wheels and two velocity commands, linear and angular velocity. Each of them is with a Nvidia TX1 such that they can communicate with each other by Robot Operating System (ROS). The parameters used in the algorithm are shown is Table.6.1. After training the network, we can use the algorithm to generate the path to achieve the formation control for the robot systems.

Table 6.1: Parameters Used in the Algorithm

Parameter Name	Parameter Value
Learning rate of actor	0.0001
Learning rate of critic	0.00001
Batch size	130
γ	0.001
Max steps in one episode	60

Let $(r_{xi}, r_{yi}), \theta_i$ denote the Cartesian position and orientation of the ith robot, respectively. Let (v_i, w_i) denote the linear and angular speed of the ith robot. The kinematic equation of the ith robot can be written as:

$$\dot{r}_{xi} = v_i \cos(\theta_i),\ \dot{r}_{yi} = v_i \sin(\theta_i), \dot{\theta}_i = w_i \tag{6.14}$$

By linearize (6.14) for a fixed point off the center of the wheel axis (x_i, y_i) of the robot, where $x_i = r_{xi} + d_i \cos(\theta_i)$, $y_i = r_{yi} + d_i \sin(\theta_i)$, and $d = 0.15\text{m}$. We can write $\begin{bmatrix} v_i \\ w_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -(1/d)\sin(\theta_i) & (1/d)\cos(\theta_i) \end{bmatrix} \begin{bmatrix} a_{xi} \\ a_{yi} \end{bmatrix}$. This is a simple kinematic equation. With this equation, we can transfer the action of each agent to the velocity command used in the nonholonomic mobile robot directly.

Since our formation control is based on the consensus algorithm, we firstly is to seek the consensus state of all the robots, which is denoted by $s^r = [x_c^r, y_c^r, \theta_c^r]^{\mathrm{T}}$. The $[x_c^r, y_c^r]$ and θ_c^r denote the reference position and orientation of the formation center of the robots team, respectively. Since s^r is dynamically changing, each robot maintains a local variable $s_i^r = [x_{ci}, y_{ci}, \theta_{ci}]^T$, which is the sense value of the state s^r by each robot i. The objective of consensus is to make sure that the value of s_i^r tracks the value of s^r, $i = 1, \ldots, N$.

In order to perform the formation control, there is a constraint (6.2) needed to satisfy for each robot i, which means that each robot needs to determine its own desired position (x_i^d, y_i^d) relative to the sensed consensus state s_i^r. It can be written as follows:

$$\begin{bmatrix} x_i^d \\ y_i^d \end{bmatrix} = \begin{bmatrix} x_{ci} \\ y_{ci} \end{bmatrix} + \begin{bmatrix} \cos(\theta_{ci}) & -\sin(\theta_{ci}) \\ \sin(\theta_{ci}) & \cos(\theta_{ci}) \end{bmatrix} \begin{bmatrix} \tilde{x}_{if} \\ \tilde{y}_{if} \end{bmatrix}, \tag{6.15}$$

where $[\tilde{x}_{if}, \tilde{y}_{if}]^{\mathrm{T}}$ represents the desired deviation vector of the ith robot relative to the geometric center of the formation. Thus, this is the constraints (6.2) that each robot i should be satisfied in order to form the formation in our experiment. The real trajectory of each robot i should track $[x_i^d, y_i^d]^{\mathrm{T}}$ and thus the desired formation shape can be maintained.

Then, we can define the reward function for each robot i. The reward function is specified to give the robot award when it achieved its goal, and penalize it for getting too close with another robot to cause the collision. The reward function for each robot i can be given as:

$$r_i = \begin{cases} -(\|x_i - x_i^d\| + \|y_i - y_i^d\|) & \text{if } L > d_i \\ -10 & \text{if } L < d_i \end{cases} \tag{6.16}$$

where L is the distance between the robot i and its nearest neighbor robot, and d_i is the safe distance for each robot i to ensure that it cannot cause the collision with others. If the collision happens, the episode is end.

The initial position for the robots are $x_1 = -1.83$m, $y_1 = -1.82$m, $x_2 = 1.78$m, $y_2 = 1.80$m, $x_3 = -2.01$m, $y_3 = 2.135$m, and $x_4 = 2.135$m, $y_4 = 2.135$m. Fig.6.2 shows that the learning curve of the algorithm.

Fig.6.3 and Fig.6.4 show that the trajectory of each robot moves after 500000 training episodes. Fig.6.5 and Fig.6.6 show that the corresponding velocity in each direction for each

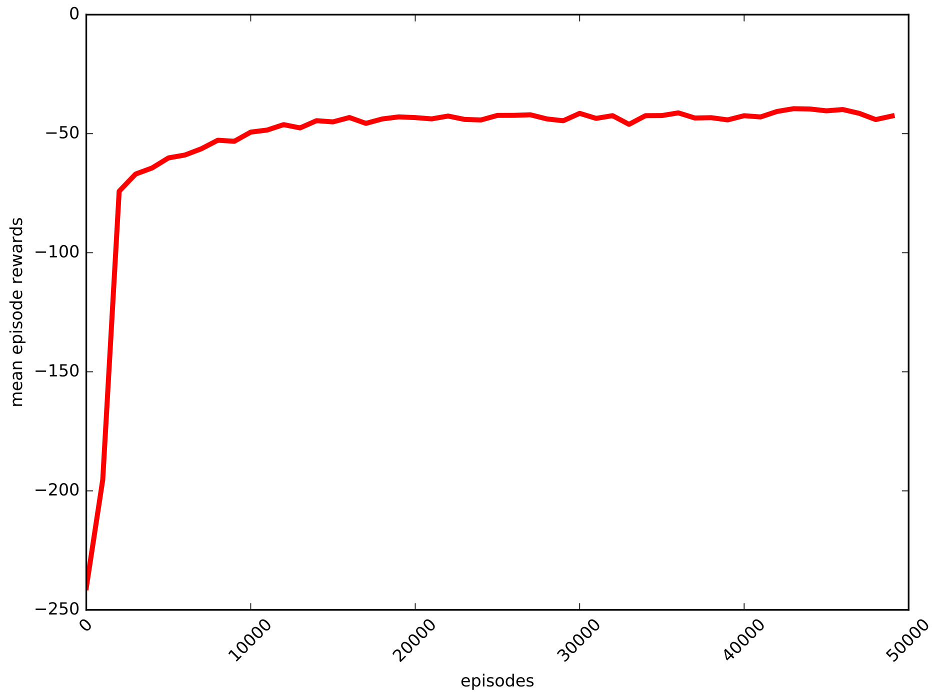


Figure 6.2: The learning curve for the formation control problem. The curve shows the mean of the average reward of 1000 episode.

robot.

Kinematics constraints of each nonholonomic mobile robot need to be considered when we run the experiments. In many works, the kinematics constraints of the mobile robot are hard to encode, thus it might result in increasing the computational complexity [138]. However, it is quiet easy to incorporate these kinematic constraints in the RL framework. Some kinematic constraints are:

$$a < v_{\max}, \tag{6.17}$$

$$|\theta_{t+1} - \theta_t| < \Delta t \cdot v_{\max}, \tag{6.18}$$

where (6.17) limits that the velocity of the robot cannot exceed the maximum value of the velocity, and (6.18) specifies a maximum turning rate that corresponds to maximum velocity. The parameters used in this work are shown in Table 6.2.

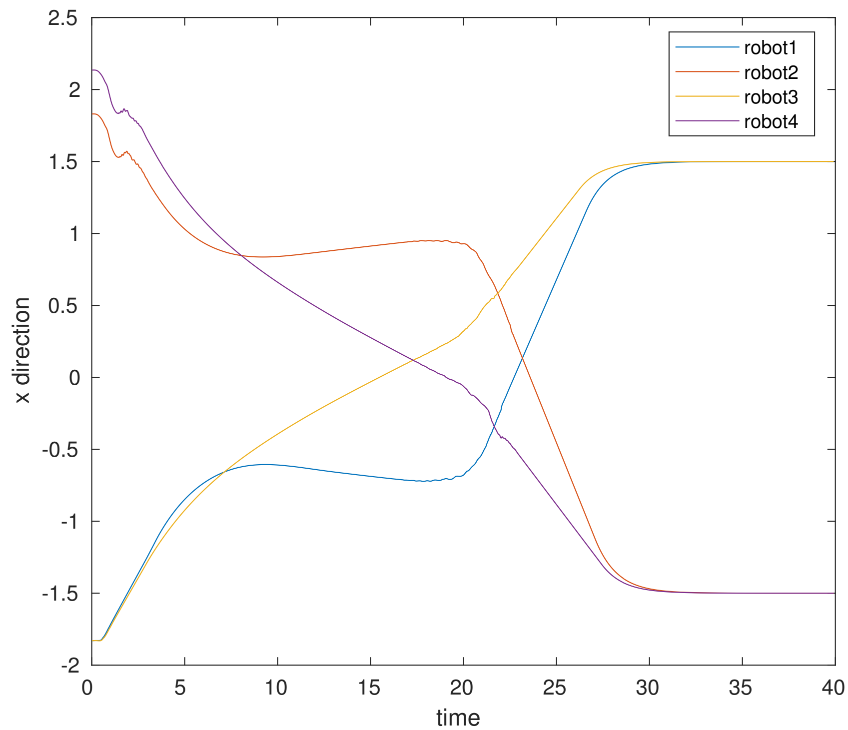


Figure 6.3: The x direction trajectory of each robot

With kinematics constraints incorporated in the experiments, the trajectory of all four robots is shown in Fig.6.7. The arrow direction denotes the direction that the robot moves toward.

Next, we will perform this trained DRL to the real robot systems. Fig.6.8-Fig.6.10 show the experiment, where Fig.6.8 shows that the robots are in the initial position, Fig. 6.9 shows that robots are running. They are trying to avoid each other in order to preventing the collision, and Fig.6.10 shows that finally, the robots are achieving the formation.

6.5 Conclusion

In this chapter, we developed a multi-agent formation control algorithm based on the application of deep reinforcement learning. In particular, this method can not only let

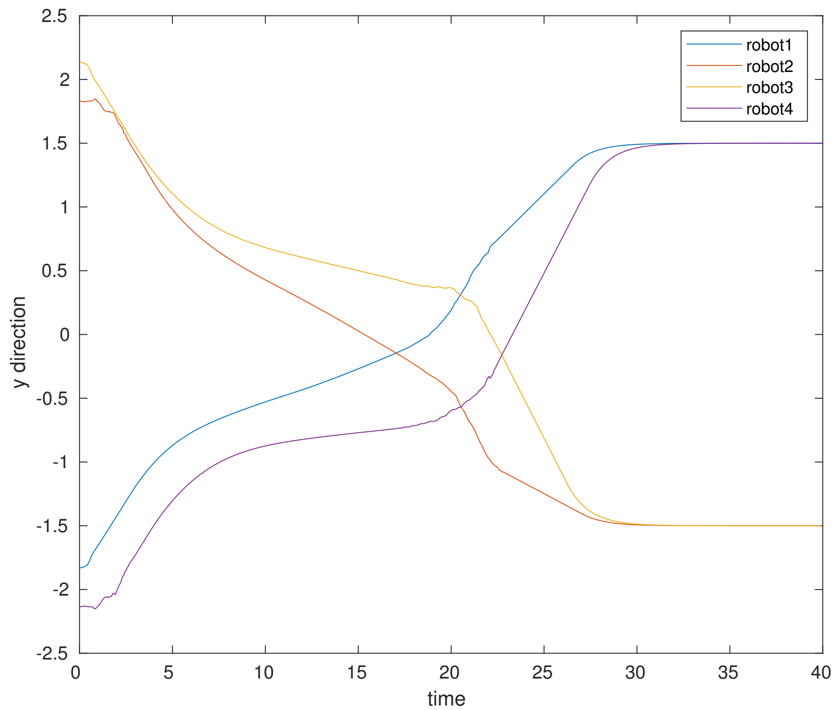


Figure 6.4: The y direction trajectory of each robot

each robot learn its own policies but also estimate the policies of other agent executing. Meanwhile, it can perform the collision avoidance while the robots are running. The simulation results show that this algorithm is effectiveness and can incorporate some kinematics constraints for each robot.

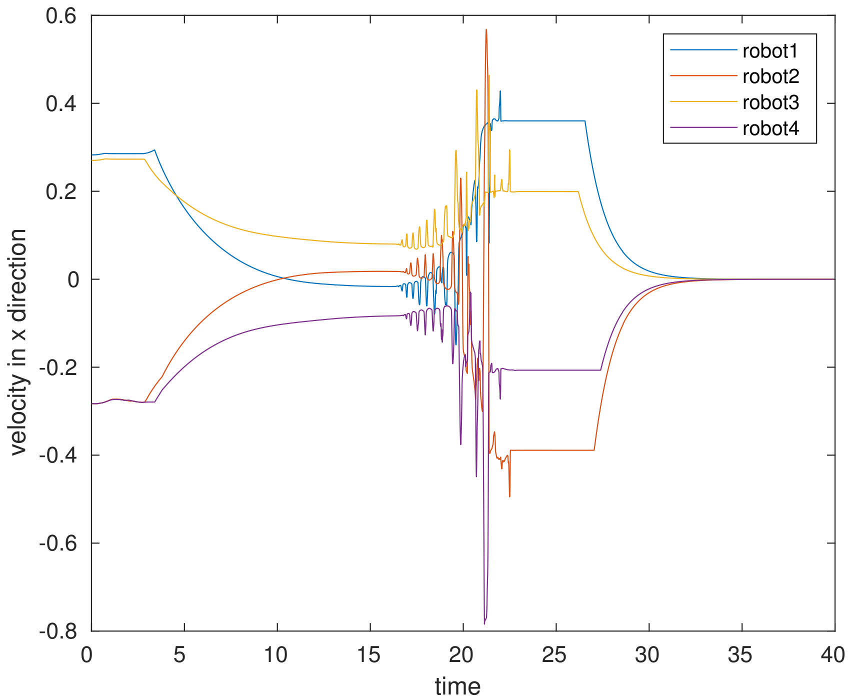


Figure 6.5: The velocity of x direction trajectory of each robot

Table 6.2: Some Parameters Used in the Experiment

Parameter Name	Parameter Value
$\tilde{x}_{1f}$	1.5m
$\tilde{y}_{1f}$	1.5m
$\tilde{x}_{2f}$	-1.5m
$\tilde{y}_{2f}$	1.5m
$\tilde{x}_{3f}$	1.5m
$\tilde{y}_{3f}$	-1.5m
$\tilde{x}_{4f}$	-1.5m
$\tilde{y}_{4f}$	-1.5m
d_i	0.6m
$v_{\max}$	1m/s

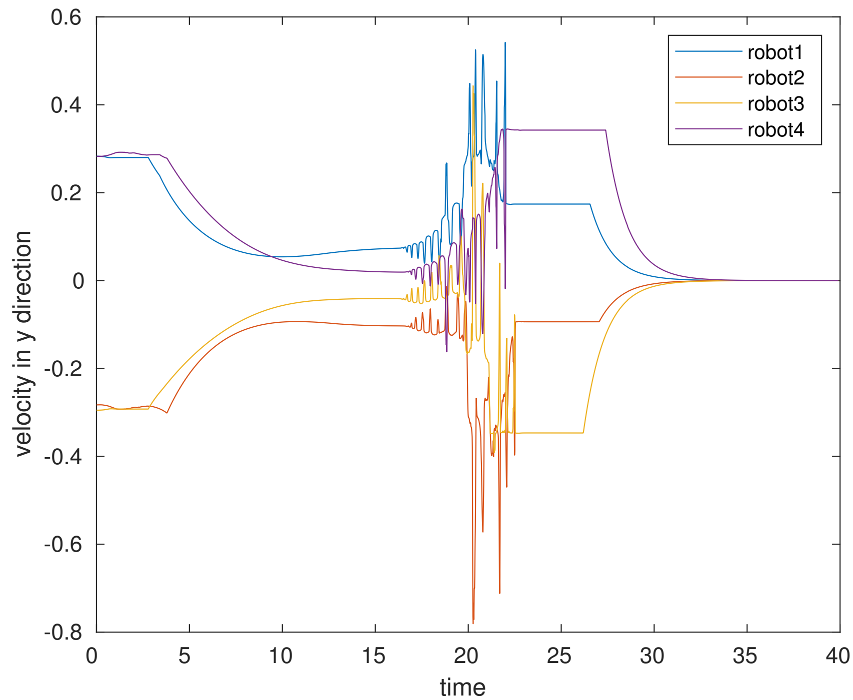


Figure 6.6: The velocity of y direction trajectory of each robot

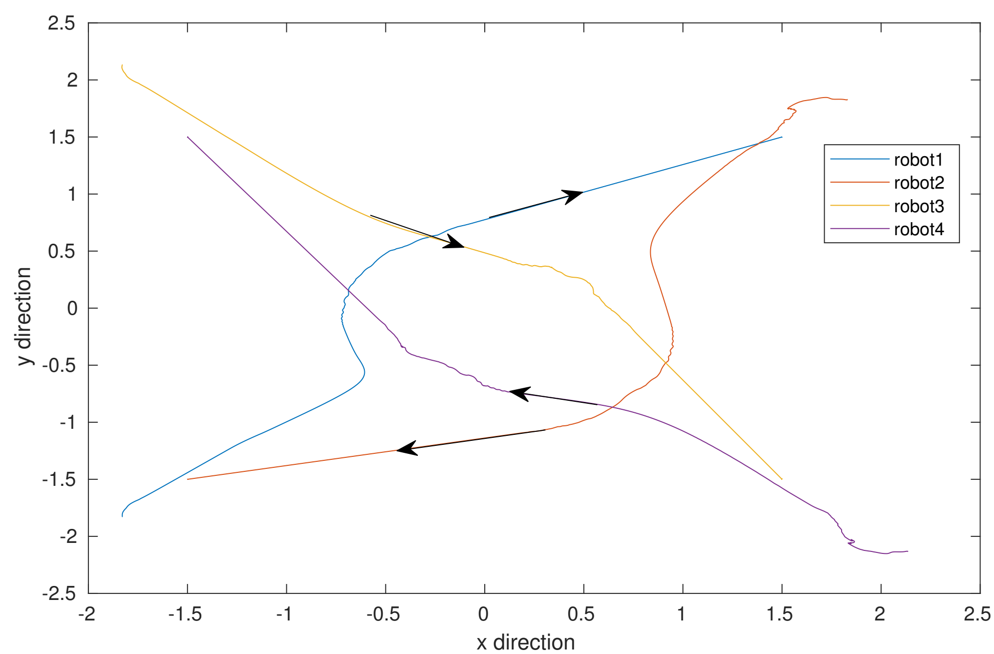


Figure 6.7: The trajectory of each robot with kinematics constraints

Figure 6.8: The initial position for all of the robots

Figure 6.9: The robots are avoiding each other to prevent the collision

Figure 6.10: The final position of all of the robots

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion of This Dissertation

The goal of this research is to develop a novel bio-inspired consensus protocol which can guide the direction of the convergence. In order to fulfill this target, the bat research algorithm is considered to be incorporated into the general consensus protocol. By introducing the flux function, the proposed consensus protocol can enhance the connection of the agents in the topological graph. Moreover, By embedding the flux function in the speed-down and speed-up term, the proposed consensus protocol can provide extra force to make this protocol converge quickly. After that, the privacy of the agents is analyzed. By borrowing the notation of the sensitivity, we can apply the differential privacy to the proposed consensus protocol. Thus the proposed consensus protocol can protect the privacy of the agents in the network. Also, we show that this new consensus protocol is convergence with the expectation taken over the coin-flip. Finally, we use the deep reinforcement learning method to achieve the goal of the consensus. The reason we use it mainly because we want to get rid of the complex mathematical model when designing a consensus protocol. We extend the deep deterministic policy gradients (DDPG) algorithm to multi-agent scenario. By carefully designing the reward function for the agents, the algorithm can reach the consensus for this multi-agent scenario.

The following conclusions can be drawn upon the fulfillment of this dissertation.

By borrowing the ability of the bat searching algorithm, the proposed consensus protocol can guide the agreement state to converge the solution of the optimization problem. In order to improve the speed of the convergence, the flux function is introduced to be incorporated to the speed-up and speed-down term. When we consider the linear case of the flux function, the proposed consensus protocol can be transformed into a matrix form. By matrix paracontracting technique and matrix rank notation, both two types of convergence, namely quadratic monotone convergence and orthant invariant convergence, can be proven and shown in simulation result.

The nonlinear case of the flux function is also considered in this dissertation. Because of the nonlinear term including the flux function, the consensus protocol cannot be transformed to the matrix form. Thus, in order to prove the convergence, the Lyapunov theory is used. We first define the error term according to the state information, then define the Lyapunov function based on this error term. Moreover, we consider the proposed consensus protocol with the external disturbances. In this situation, we prove that the protocol can robustly reach consensus with the performance γ.

Moreover, the privacy of the agents is considered in this dissertation. By defining the sensitivity of the two initial states, we can measure the two different initial states and thus we can define the notation of ϵ-*differential privacy*. This technique can prevent others to acquire the exact initial information of the agents in the network topology if certain condition is satisfied. Also, we prove that this revised consensus protocol is convergent under the differential privacy condition and determine the accuracy of the consensus state.

Finally, the deep reinforcement learning is used to reach the consensus agreement. The reason why we use it is that we want to get rid of complex mathematical model of the controller when designing the traditional consensus protocol. The deep deterministic policy gradients method is used and extended to the multi agents scenario. The reward function is designed to make the consensus problem to be fitted for the reinforcement learning algorithm.

This algorithm is performed in the mobile robots. Thus the consensus is transformed to the formation control for the multi agents and the results show that the deep reinforcement learning can reach the consensus agreement.

7.2 Contributions of This Dissertation

The contributions of the dissertation are summarized as follows:

- A review of consensus protocol and its applications were provided in the dissertation.
- The bio-inspired consensus protocol is proposed. Moreover, the matrix paracontraction is introduced and the characteristics of the linear case of the bio-inspired consensus are studied.
- By using matrix paracontraction technique, two types of convergence, namely quadratic monotone convergence and orthant invariant convergence, are proven for the linear case of the consensus protocol.
- Nonlinear case of the bio-inspired consensus is also studied. In this case, the Lyapunov theory is applied to the nonlinear case of the consensus protocol to show that it is convergent.
- We also consider the nonlinear case of the bio-inspired consensus protocol with external disturbances and prove it is convergent.
- The sensitivity of two different initial states of the consensus is defined. Then the differential privacy of the bio-inspired consensus protocol is proposed and discussed. After that, the convergence of this new type of consensus is also studied. The upper bound of the accuracy of the proposed consensus algorithm is discussed.

- The DDPG algorithm is extend to the multi agents scenario and we apply this extended version to solve the consensus problem. The extended algorithm is tested on the mobile robots platform.

7.3 Recommendation for Future Research

Some of the recommendations for future research are listed as follows:

- The mobile robots platform can be used to test the bio-inspired consensus protocol to further confirm the convergence of the bio-inspired consensus. These tests and validations are the important step to verify the effectiveness of the proposed consensus protocol.
- Some of the assumptions, such as the semi-Lipschitz condition and the convex of the optimization problem, should be relaxed for future research direction. Also, the discontinuous flux function should be considered in the further research.
- The proposed bio-inspired consensus protocol can be applied in multi-layer, multi-dependent cyber-physical network systems.
- The time consumption for the DDPG algorithm should be investigated. Moreover, the implemented formation control should also be added the ability to guide the convergence direction to the solution of the optimization problem.

BIBLIOGRAPHY

[1] A. Rogers, E. David, N. R. Jennings, and J. Schiff, "The effects of proxy bidding and minimum bid increments within ebay auctions," *ACM Transactions on the Web (TWEB)*, vol. 1, no. 2, p. 9, 2007. 1.1

[2] N. Schurr, J. Marecki, M. Tambe, P. Scerri, N. Kasinadhuni, and J. P. Lewis, "The future of disaster response: Humans working with multiagent teams using defacto." in *AAAI spring symposium: AI technologies for homeland security*, 2005, pp. 9–16. 1.1

[3] R. Sun and I. Naveh, "Simulating organizational decision-making using a cognitively realistic agent model," *Journal of Artificial Societies and Social Simulation*, vol. 7, no. 3, 2004. 1.1

[4] J. Ferber and G. Weiss, *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-Wesley Reading, 1999, vol. 1. 1.1

[5] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Comput. Graph.*, vol. 21, pp. 25–34, 1987. 2.1

[6] J. Toner and Y. Tu, "Flocks, herds, and schools: A quantitative theory of flocking," *Phys. Rev. E*, vol. 58, pp. 4828–4858, 1998. 2.1

[7] K. Warburton and J. Lazarus, "Tendency-distance models of social cohesion in animal groups," *J. Theoretical Bio.*, vol. 150, no. 4, pp. 473–488, 1991. 2.1

[8] D. Grunbaum and A. Okubo, "Modelling social animal aggregations," in *Frontiers in Mathematical Biology*, ser. Lecture Notes in Biomathematics. Springer-Verlag, 1994, vol. 100, pp. 296–325. 2.1

[9] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-deriven particles," *Phys. Rev. Lett.*, vol. 75, pp. 1226–1229, 1995. 2.1

[10] F. Cucker and S. Smale, "Emergent behavior in flocks," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 852–862, 2007. 2.1

[11] D. Ye, M. Zhang, and A. V. Vasilakos, "A survey of self-organization mechanisms in multiagent systems," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 47, no. 3, pp. 441–461, 2017. 2.1

[12] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, pp. 988–1001, 2003. 2.1, 3.1

[13] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, pp. 1520–1533, 2004. 2.1, 3.1, 5.1

[14] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Trans. Autom. Control*, vol. 50, pp. 169–182, 2005. 2.1

[15] Q. Hui and W. M. Haddad, "Distributed nonlinear control algorithms for network consensus," *Automatica*, vol. 44, pp. 2375–2381, 2008. 2.1, 3.1.1

[16] Q. Hui, W. M. Haddad, and S. P. Bhat, "Finite-time semistability and consensus for nonlinear dynamical networks," *IEEE Trans. Autom. Control*, vol. 53, pp. 1887–1900, 2008. 2.1

[17] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, pp. 803–812, 1986. 2.1, 5.1

[18] V. Borkar and P. P. Varaiya, "Asymptotic agreement in distributed estimation," *IEEE Trans. Autom. Control*, vol. 27, pp. 650–655, 1982. 2.1

[19] J. R. Marden, G. Arslan, and J. S. Shamma, "Cooperative control and potential games," *IEEE Trans. Syst. Man Cybern. Part B: Cybern.*, vol. 39, no. 6, pp. 1393–1407, 2009. 2.1

[20] Q. Hui and H. Zhang, "Optimal balanced coordinated network resource allocation using swarm optimization," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 45, no. 5, pp. 770–787, 2015. 2.1, 3.1.2

[21] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Trans. Industrial Informatics*, vol. 9, no. 1, pp. 427–438, Feb 2013. 2.1

[22] S. Knorn, Z. Chen, and R. H. Middleton, "Overview: Collective control of multiagent systems," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 4, pp. 334–347, Dec 2016. 2.1

[23] M. Nourian, P. E. Caines, and R. P. Malham'e, "A mean field game synthesis of initial mean consensus problems: A continuum approach for non-Gaussian behavior," *IEEE Trans. Automat. Control*, vol. 59, no. 2, pp. 449–455, Feb 2014. 2.1

[24] Q. Hui, W. M. Haddad, and S. P. Bhat, "Semistability, finite-time stability, differential inclusions, and discontinuous dynamical systems having a continuum of equilibria," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2465–2470, 2009. 2.1

[25] F. Xiao, L. Wang, and T. Chen, "Finite-time consensus in networks of integrator-like dynamic agents with directional link failure," *IEEE Trans. Automat. Control*, vol. 59, no. 3, pp. 756–762, March 2014. 2.1

[26] W. Chen, X. Li, W. Ren, and C. Wen, "Adaptive consensus of multi-agent systems with unknown identical control directions based on a novel Nussbaum-type function," *IEEE Trans. Automat. Control*, vol. 59, no. 7, pp. 1887–1892, July 2014. 2.1

[27] A. Pilloni, A. Pisano, Y. Orlov, and E. Usai, "Consensus-based control for a network of diffusion PDEs with boundary local interaction," *IEEE Trans. Automat. Control*, vol. 61, no. 9, pp. 2708–2713, Sept 2016. 2.1

[28] C. Altafini, "Consensus problems on networks with antagonistic interactions," *IEEE Trans. Automat. Control*, vol. 58, no. 4, pp. 935–946, April 2013. 2.1

[29] A. V. Proskurnikov, A. S. Matveev, and M. Cao, "Opinion dynamics in social networks with hostile camps: Consensus vs. polarization," *IEEE Trans. Automat. Control*, vol. 61, no. 6, pp. 1524–1536, June 2016. 2.1

[30] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006. 2.1

[31] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, Australia, 1995, pp. 1942–1948. 2.1

[32] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997. 2.1

[33] H. Zhang and Q. Hui, "Parallel multiagent coordination optimization algorithm: Implementation, evaluation, and applications," *IEEE Trans. Autom. Sci Eng.*, vol. 14, no. 2, pp. 984–995, 2017. 2.1, 3.1, 3.3.2, 5.1

[34] ——, "Binary multiagent coordination optimization with application to formation control design," in *2013 IEEE Congress on Evolutionary Computation (CEC)*, Cancun, Mexico, 2013, pp. 1968–1975. 2.1

[35] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Springer, 2010, pp. 65–74. 2.1, 3.1, 3.1.1, 4.2, 5.1, 5.2.1

[36] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM Journal on Applied Mathematics*, vol. 50, no. 6, pp. 1645–1662, 1990. 2.2

[37] S. H. Strogatz, "From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators," *Physica D: Nonlinear Phenomena*, vol. 143, no. 1-4, pp. 1–20, 2000. 2.2

[38] G. B. Ermentrout and N. Kopell, "Frequency plateaus in a chain of weakly coupled oscillators, i." *SIAM journal on Mathematical Analysis*, vol. 15, no. 2, pp. 215–237, 1984. 2.2

[39] R. Sepulchre, D. Paley, and N. Leonard, "Collective motion and oscillator synchronization," in *Cooperative control*. Springer, 2005, pp. 189–205. 2.2

[40] A. Papachristodoulou and A. Jadbabaie, “Synchronization in oscillator networks: Switching topologies and non-homogeneous delays,” in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 5692–5697. 2.2

[41] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004. 2.2

[42] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: Algorithms and theory,” CALIFORNIA INST OF TECH PASADENA CONTROL AND DYNAMICAL SYSTEMS, Tech. Rep., 2004. 2.2

[43] J. Cortés, S. Martínez, and F. Bullo, “Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions,” *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, 2006. 2.2

[44] R. Olfati-Saber, “Distributed kalman filter with embedded consensus filters,” in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 8179–8184. 2.2

[45] Y. Mo and R. M. Murray, “Privacy preserving average consensus,” *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 753–765, Feb 2017. 2.3, 5.1

[46] C. Dwork, “Differential privacy,” in *Encyclopedia of Cryptography and Security*. Springer, 2011, pp. 338–340. 2.3, 5.1

[47] J. Le Ny and G. J. Pappas, “Differentially private filtering,” *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 341–354, Feb 2014. 2.3

[48] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*. ACM, 2012, pp. 81–90. 2.3, 5.1

[49] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135. 2.4

[50] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999. 2.4

[51] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015. 2.4, 6.1

[52] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016. 2.4

[53] M. Lauer and M. Riedmiller, "An algorithm for distributed reinforcement learning in cooperative multi-agent systems," in *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer, 2000. 2.4

[54] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *International Conference on Autonomous Agents and Multiagent Systems*. Springer, 2017, pp. 66–83. 2.4

[55] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, “Counterfactual multi-agent policy gradients,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2.4

[56] H. Zhang and Q. Hui, “Multiagent coordination optimization: A control-theoretic perspective of swarm intelligence algorithms,” in *2013 IEEE Congress on Evolutionary Computation (CEC)*, Cancun, Mexico, 2013, pp. 3339–3346. 3.1, 5.1

[57] W. M. Haddad, V. Chellaboina, and Q. Hui, *Nonnegative and Compartmental Dynamical Systems*. Princeton, NJ: Princeton Univ. Press, 2010. 3.1.1, 3.2.2

[58] J. M. Berg, D. H. S. Maithripala, Q. Hui, and W. M. Haddad, “Thermodynamics-based control of network systems,” *ASME J. Dyna. Syst. Meas. Control*, vol. 135, no. 5, 2013. 3.1.1

[59] H. Zhang, F. Zhang, and Q. Hui, “A speed-up and speed-down strategy for swarm optimization,” in *2014 ACM Genetic and Evolutionary Computation Conference (GECCO)*, Vancouver, Canada, 2014, pp. 1481–1482. 3.1.1

[60] Z. Jin and R. M. Murray, “Multi-hop relay protocols for fast consensus seeking,” in *Proc. IEEE Conf. Decision Control*, San Diego, CA, 2006. 3.1.1, 5.2.1

[61] Q. Hui, “Optimal semistable control in *ad hoc* network systems: A sequential two-stage approach,” *IEEE Trans. Autom. Control*, vol. 58, no. 3, pp. 779–784, 2013. 3.1.1

[62] ——, “Further results on paracontracting matrices and correction to ‘Optimal semistable control in *ad hoc* network systems: A sequential two-stage approach’,”

IEEE Trans. Autom. Control, vol. 60, no. 12, pp. 3305–3309, 2015. 3.1.1, 3.1.2, 3.1.2, 3.1.2, 3.1.2, 3.1.2

[63] L. Elsner, I. Koltracht, and M. Neumann, "On the convergence of asynchronous paracontractions with applications to tomographic reconstruction from incomplete data," *Linear Alg. Appl.*, vol. 130, pp. 65–82, 1990. 3.1.1, 3.1.2, 1

[64] D. P. Bertsekas and J. N. Tsitsiklis, "Comments on 'Coordination of groups of mobile autonomous agents using nearest neighbor rules'," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 968–969, 2007. 3.1.1, 3.2, 3.2.2, 3.2.2, 9

[65] S. Nelson and M. Neumann, "Generalizations of the projection method with applications to SOR theory for Hermitian positive semidefinite linear systems," *Numer. Math.*, vol. 51, no. 2, pp. 123–141, 1987. 3.1.2, 3.1.2

[66] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inform. Theory*, vol. 52, pp. 2508–2530, 2006. 3.1.2

[67] X. Wang and Z. Cheng, "The convergence of infinite product of uniformly paracontracting matrices and its application in consensus of the Vicsek model," *J. Sys. Sci. Math. Scis.*, vol. 33, no. 6, pp. 724–731, 2013. 3.1.2, 3.1.2

[68] D. S. Bernstein, *Matrix Mathematics,* 2nd ed. Princeton, NJ: Princeton Univ. Press, 2009. 3.1.2, 3.1.2, 3.1.2

[69] W. M. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton, NJ: Princeton Univ. Press, 2008. 3.1.2, 4.3, 4.4

[70] Q. Hui, "Semistability of nonlinear systems having a connected set of equilibria and time-delays," *IEEE Trans. Autom. Control*, vol. 57, no. 10, pp. 2615–2620, 2012. 3.4

[71] J. A. Fax and R. M. Murray, “Information flow and cooperative control of vehicle formations,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, Sept 2004.

[72] T. Eren, P. N. Belhumeur, and A. S. Morse, “Closing ranks in vehicle formations based on rigidity,” in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3, Dec 2002, pp. 2959–2964 vol.3. 4.1

[73] R. Vidal, O. Shakernia, and S. Sastry, “Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation,” in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA ’03.*, vol. 1, Sept 2003, pp. 584–589 vol.1. 4.1

[74] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” *ACM SIGGRAPH computer graphics*, vol. 21, no. 4, pp. 25–34, 1987. 4.1

[75] J. Toner and Y. Tu, “Flocks, herds, and schools: A quantitative theory of flocking,” *Physical review E*, vol. 58, no. 4, p. 4828, 1998. 4.1

[76] J. Cortés and F. Bullo, “Coordination and geometric optimization via distributed dynamical systems,” *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2005. 4.1

[77] 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, Sept 2004. 4.1

[78] J. Cortés, “Distributed algorithms for reaching consensus on general functions,” *Automatica*, vol. 44, no. 3, pp. 726–737, 2008. 4.1

[79] P. Lin, Y. Jia, and L. Li, "Distributed robust H_∞ consensus control in directed networks of agents with time-delay," *Systems & Control Letters*, vol. 57, no. 8, pp. 643–653, 2008. 4.1

[80] R. Carli, F. Bullo, and S. Zampieri, "Quantized average consensus via dynamic coding/decoding schemes," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 2, pp. 156–175, 2010. 4.1

[81] 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, Feb 2011. 4.1

[82] Y. Hong, G. Chen, and L. Bushnell, "Distributed observers design for leader-following control of multi-agent networks," *Automatica*, vol. 44, no. 3, pp. 846–850, 2008. 4.1

[83] M. Areak, "Passivity as a design tool for group coordination," in *2006 American Control Conference*, June 2006, pp. 6 pp.–. 4.1

[84] K. Warburton and J. Lazarus, "Tendency-distance models of social cohesion in animal groups," *Journal of Theoretical Biology*, vol. 150, no. 4, pp. 473–488, 1991. 4.1

[85] D. Grünbaum and A. Okubo, "Modelling social animal aggregations," in *Frontiers in mathematical biology*. Springer, 1994, pp. 296–325. 4.1

[86] 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, June 2003. 4.1

[87] 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, May 2005. 4.1

[88] Q. Hui, W. M. Haddad, and S. P. Bhat, "Finite-time semistability and consensus for nonlinear dynamical networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1887–1900, Sept 2008. 4.1

[89] H. Zhang and Q. Hui, "Multiagent coordination optimization: A control-theoretic perspective of swarm intelligence algorithms," in *2013 IEEE Congress on Evolutionary Computation*, June 2013, pp. 3339–3346. 4.1

[90] Q. Hui and W. M. Haddad, "Distributed nonlinear control algorithms for network consensus," *Automatica*, vol. 44, no. 9, pp. 2375–2381, 2008. 4.2, 5.2.1

[91] W. M. Haddad, V. Chellaboina, and Q. Hui, *Nonnegative and compartmental dynamical systems*. Princeton University Press, 2010. 4.2, 5.2.1

[92] J. M. Berg, D. H. S. Maithripala, Q. Hui, and W. M. Haddad, "Thermodynamics-based control of network systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 135, no. 5, p. 051003, 2013. 4.2, 5.2.1

[93] H. Zhang, F. Zhang, and Q. Hui, "A speed-up and speed-down strategy for swarm optimization," in *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO Comp '14. New York, NY, USA: ACM, 2014, pp. 1481–1482. [Online]. Available: http://doi.acm.org/10.1145/2598394.2602285 4.2

[94] A. Zemouche and M. Boutayeb, "A new observer design method for a class of Lipschitz nonlinear discrete-time systems with time-delay: Extension to H_∞ performance analysis," in *46th IEEE Conf. Decision Control*, 2007, pp. 414–419. 4.4

[95] J. Le Ny and G. J. Pappas, "Differentially private filtering," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 341–354, 2014. 5.1

[96] W. M. Haddad, V. Chellaboina, and S. G. Nersesov, *Thermodynamics: A Dynamical Systems Approach*. Princeton, NJ: Princeton Univ. Press, 2005. 5.1

[97] H. Zhang, Q. Hui, and Q. Liu, "Bio-inspired consensus under suggested convergence direction," in *2017 American Control Conference (ACC)*, Seattle, WA, May 2017, pp. 4262–4267. 5.2.1

[98] Q. Liu and Q. Hui, "The convergence analysis of bat-inspired consensus protocols with nonlinear dynamics," in *13th IEEE International Conference on Automation Science and Engineering*, Xi'an, China, Aug 2017. 5.2.1

[99] Q. Hui, W. Qiao, and C. Peng, "Neuromorphic-computing-based feedback control: A cognitive supervisory control framework," in *IEEE Conf. Decision Control*, Las Vegas, NV, 2016. 5.2.1

[100] H. Zhang, F. Zhang, and Q. Hui, "A speed-up and speed-down strategy for swarm optimization," in *Proceedings of the Companion Publication of the 2014 ACM Conference on Genetic and Evolutionary Computation*. ACM, 2014, pp. 1481–1482. 5.2.1

[101] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proceedings of the 42nd ACM Symposium on Theory of Computing*. ACM, 2010, pp. 715–724. 5.2.2

[102] X. Ge, Q.-L. Han, D. Ding, X.-M. Zhang, and B. Ning, "A survey on recent advances in distributed sampled-data cooperative control of multi-agent systems," *Neurocomputing*, vol. 275, pp. 1684–1701, 2018. 6.1

[103] M. Fields, E. Haas, S. Hill, C. Stachowiak, and L. Barnes, "Effective robot team control methodologies for battlefield applications," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 5862–5867. 6.1

[104] J. Ding, J. H. Gillula, H. Huang, M. P. Vitus, W. Zhang, and C. J. Tomlin, "Hybrid systems in robotics," *IEEE Robotics & Automation Magazine*, vol. 18, no. 3, pp. 33–43, 2011. 6.1

[105] H. Rezaee and F. Abdollahi, "Pursuit formation of double-integrator dynamics using consensus control approach," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 7, pp. 4249–4256, 2015. 6.1

[106] Q. Lu, Q.-L. Han, X. Xie, and S. Liu, "A finite-time motion control strategy for odor source localization," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5419–5430, 2014. 6.1

[107] H. Li, P. Xie, and W. Yan, "Receding horizon formation tracking control of constrained underactuated autonomous underwater vehicles," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5004–5013, 2017. 6.1

[108] P. Millán, L. Orihuela, I. Jurado, and F. R. Rubio, "Formation control of autonomous underwater vehicles subject to communication delays," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 770–777, 2014. 6.1

[109] M. Ou, H. Du, and S. Li, "Finite-time formation control of multiple nonholonomic mobile robots," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 1, pp. 140–165, 2014. 6.1

[110] T. Liu and Z.-P. Jiang, "Distributed formation control of nonholonomic mobile robots without global position measurements," *Automatica*, vol. 49, no. 2, pp. 592–600, 2013. 6.1

[111] C. Ma and J. Zhang, "On formability of linear continuous-time multi-agent systems," *Journal of Systems Science and Complexity*, vol. 25, no. 1, pp. 13–29, 2012. 6.1

[112] X. Dong, B. Yu, Z. Shi, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles: Theories and applications," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 340–348, 2015. 6.1

[113] Z. Lin, L. Wang, Z. Han, and M. Fu, "Distributed formation control of multi-agent systems using complex laplacian," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1765–1777, 2014. 6.1

[114] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015. 6.1, 8

[115] 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. 6.1

[116] M. Aranda, G. Lpez-Nicols, C. Sags, and Y. Mezouar, "Formation control of mobile robots using multiple aerial cameras," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 1064–1071, Aug 2015. 6.1

[117] S.-C. LIU, D.-L. TAN, and G.-J. LIU, "Formation control of mobile robots with active obstacle avoidance," *Acta Automatica Sinica*, vol. 33, no. 5, pp. 529 – 535, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1874102907600238 6.1

[118] H. Wang, D. Guo, X. Liang, W. Chen, G. Hu, and K. K. Leang, "Adaptive vision-based leaderfollower formation control of mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 2893–2902, April 2017. 6.1

[119] Y. Kuriki and T. Namerikawa, "Consensus-based cooperative formation control with collision avoidance for a multi-uav system," in *2014 American Control Conference*, June 2014, pp. 2077–2082. 6.1

[120] R. Falconi, L. Sabattini, C. Secchi, C. Fantuzzi, and C. Melchiorri, "Edge-weighted consensus-based formation control strategy with collision avoidance," *Robotica*, vol. 33, no. 2, p. 332347, 2015. 6.1

[121] A. Richards and J. How, "Decentralized model predictive control of cooperating uavs," in *43rd IEEE Conference on Decision and Control*, vol. 4. Citeseer, 2004, pp. 4286–4291. 6.1

[122] A. T. Hafez, A. J. Marasco, S. N. Givigi, A. Beaulieu, and C. A. Rabbath, "Encirclement of multiple targets using model predictive control," in *2013 American Control Conference*, June 2013, pp. 3147–3152. 6.1

[123] R. Vidal, O. Shakernia, and S. Sastry, "Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 584–589. 6.1

[124] W. Dong, "Distributed observer-based cooperative control of multiple nonholonomic mobile agents," *International Journal of Systems Science*, vol. 43, no. 5, pp. 797–808, 2012. 6.1

[125] ——, “Tracking control of multiple-wheeled mobile robots with limited information of a desired trajectory,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 262–268, Feb 2012. 6.1

[126] N. Hovakimyan and C. Cao, *L1 adaptive control theory: guaranteed robustness with fast adaptation*. SIAM-Society for Industrial and Applied Mathematics, 2010, vol. 21. 6.1

[127] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013. 6.1

[128] B.-Q. Huang, G.-Y. Cao, and M. Guo, “Reinforcement learning neural network to the problem of autonomous mobile robot obstacle avoidance,” in *2005 International Conference on Machine Learning and Cybernetics*, vol. 1, Aug 2005, pp. 85–89. 6.1

[129] C. Li, J. Zhang, and Y. Li, “Application of artificial neural network based on q-learning for mobile robot path planning,” in *2006 IEEE International Conference on Information Acquisition*, Aug 2006, pp. 978–982. 6.1

[130] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013. 6.1

[131] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2015. 6.1

[132] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016, pp. 1928–1937. 6.1

[133] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 23–30. 6.1

[134] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 2137–2145. 6.2.3

[135] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, 2017, pp. 6379–6390. 6.2.3

[136] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the tenth international conference on machine learning*, 1993, pp. 330–337. 6.2.3

[137] R. S. Sutton, A. G. Barto, *et al.*, *Reinforcement learning: An introduction*. MIT press, 1998. 6.2.4

[138] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 1928–1935. 6.4

APPENDIX A

LIST OF PUBLICATIONS

[1] **Qishuai Liu**, Emmanuel Moulay, Patrick Coirault, and Qing Hui,“Deep Learning Based Formation Control for the Multi-Agent Coordination,” *16th IEEE International Conference on Networking, Sensing and Control*, Banff, Canada, May 2019.

[2] Fayrouz Isfoula, Emmanuel Bernuau, Emmanuel Moulay, Patrick Coirault, **Qishuai Liu**, and Qing Hui, “Practical Consensus Tracking of Multi-Agent Systems with Linear Controllers,” *2019 European Control Conference*, Napoli, Italy, June 2019.

[3] **Qishuai Liu** and Qing Hui, “B-Splines-Based Fuzzy C-Means to Maximizing Overlap Areas for Interconnected Power Systems,” *14th IEEE Conference on Industrial Electronics and Applications*, Xian, China, June 2019.

[4] **Qishuai Liu** and Qing Hui, “The Formation Control of Mobile Autonomous Multi-Agent Systems Using Deep Reinforcement Learning,” *13th Annual IEEE International Systems Conference*, Orlando, FL, April 2019.

[5] Jie Cheng, **Qishuai Liu**, Qing Hui, and Fred Choobineh, “The Joint Optimization of Critical Interdependent Infrastructure of an Electricity-Water-Gas System,” *16th Annual Conference on Systems Engineering Research, Charlottesville*, VA, May 2018.

[6] **Qishuai Liu** and Qing Hui, “The Bat-Inspired Consensus Protocols with Differential Privacy,” *14th IEEE International Conference on Control and Automation*, Anchorage,

AK, June 2018.

[7] **Qishuai Liu** and Qing Hui, “Hybrid Stability Analysis via Extended Small-Gain Theorem for Networked Cyber-Physical Systems with Transmission Delay,” *13th IEEE Conference on Automation Science and Engineering*, Xian, China, August 2017.

[8] **Qishuai Liu** and Qing Hui, “The Convergence Analysis of Bat-Inspired Consensus Protocols with Nonlinear Dynamics,” *13th IEEE Conference on Automation Science and Engineering*, Xian, China, August 2017.

[9] Haopeng Zhang, Qing Hui, and **Qishuai Liu**, “Bio-Inspired Consensus under Suggested Convergence Direction,” *2017 American Control Conference*, Seattle, WA, May 2017.

[10] **Qishuai Liu** and Qing Hui, “A Hybrid ACO Algorithm Based on Bayesian Factorizations and Reinforcement Learning for Continuous Optimization,” *2016 IEEE Congress on Evolutionary Computation, Vancouver*, Canada, July 2016.