Connectivity and Consensus in Multi-Agent Systems with Uncertain Links

Mina Babahaji

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

November 2021

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mina Babahaji**

Entitled: **Connectivity and Consensus in Multi-Agent Systems with Uncertain Links**

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

Master of Applied Science (Electrical and Computer Engineering)

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

Signed by the Final Examining Committee:

______________________________ Chair
Dr. Krzysztof Skonieczny

______________________________ External Examiner
Dr. Walter Lucia (CIISE)

______________________________ Examiner
Dr. Krzysztof Skonieczny

______________________________ Supervisor
Dr. Amir G. Aghdam

Approved by ______________________________
Yousef Shayan, Chair
Department of Electrical and Computer Engineering

______________ 2021 ______________________________
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Connectivity and Consensus in Multi-Agent Systems with Uncertain Links

Mina Babahaji

In the analysis and design of a multi-agent system (MAS), studying the graph representing the system is essential. In particular, when the communication links in a MAS are subject to uncertainty, a random graph is used to model the system. This type of graph is represented by a probability matrix, whose elements reflect the probability of the existence of the corresponding edges in the graph. This probability matrix needs to be adequately estimated. In this thesis, two approaches are proposed to estimate the probability matrix in a random graph. This matrix is time-varying and is used to determine the network configuration at different points in time. For evaluating the probability matrix, the connectivity of the network needs to be assessed first. It is to be noted that connectivity is a requirement for the convergence of any consensus algorithm in a network. The probability matrix is used in this work to study the consensus problem in a leader-follower asymmetric MAS with uncertain communication links. We propose a novel robust control approach to obtain an approximate agreement among agents under some realistic assumptions. The uncertainty is formulated as disturbance, and a controller is developed to debilitate it. Under the proposed controller, it is guaranteed that the consensus error satisfies the global L_2-gain performance in the presence of uncertainty. The designed controller consists of two parts: one for time-varying links and one for time-invariant links. Simulations demonstrate the effectiveness of the proposed methods.

Acknowledgments

First and foremost, I want to express my thanks to Dr. Amir Aghdam for his exceptional advice in navigating challenging control theory subjects throughout my graduate studies. His persistent support and guidance were crucial in completing this thesis. My sincere thanks go to Dr. Heidar Ali Talebi, Dr. Walter Lucia, and Dr. Stéphane Blouin for their support and valuable comments through my research.

I thank the Natural Sciences and Engineering research Council of Canada, and Defence Research and Development Canada (DRDC) for providing the funding needed to complete this research.

I'd want to express my gratitude to my family and friends for their unwavering support and recommendations for the last two years. Without them, accomplishing this research would not have been possible.

List of Publications

[1] M. Babahaji, S. Blouin, W. Lucia, M. M. Asadi, H. Mahboubi and A. G. Aghdam, ”Random graphs estimation using Q-learning”, In: *2021 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)* (2021), pp. 109-114.

[2] M. Babahaji, E. Firouzmand, A. G. Aghdam, and H. A. Talebi, “Consensus control of multi-agent systems with uncertain communication links”, *submitted*.

Contribution of Authors

This thesis is presented in a manuscript-based format. Mina Babahaji wrote the papers and implemented the ideas mathematically and numerically in both papers. The first paper is co-authored by Dr. Stéphane Blouin, Dr. Walter Lucia, Dr. M. Mehdi Asadi, Dr. Hamid Mahboubi, and Dr. Amir Aghdam. They gave some comments on possible solutions to the problem and verified the results. The second paper is co-authored by Elnaz Firouzmand, whose comments helped solve problem and supervised by Dr. Amir Aghdam and Dr. Heidar A. Talebi, who verified the analytical methods and numerical simulations.

Contents

List of Figures **ix**

List of Tables **x**

1 Introduction **1**

1.1 Motivation 1

1.2 Literature Review and Preliminaries 2

1.3 Thesis Contributions 4

1.4 Thesis Layout 5

2 Random Graphs Estimation using Q-Learning **6**

2.1 Overview 6

2.2 Introduction 7

2.3 Preliminaries and Notations 8

2.4 Probability Matrix Estimation 9

2.4.1 Probability estimation via SMA 10

2.4.2 Probability estimation via EMA 12

2.4.3 Comparing algorithms 12

2.4.4 Estimation of the entire probability matrix 13

2.5 Connectivity Assessment 14

2.6 Simulation Results 15

2.7 Conclusions 19

3 Consensus Control of Multi-Agent Systems with Uncertain Communication Links **21**

3.1 Overview 21

3.2 Introduction 22

3.3 Preliminaries and Notations 23

3.4 Problem Formulation 24

3.5 Main Results 25

3.6 Simulations 30

3.7 Conclusions 34

4 Conclusions and Future Research Directions **35**

4.1 Suggestions for Future Work 35

Bibliography **37**

List of Figures

Figure 2.1 The topology of directed graph 16

Figure 2.2 Comparison of the estimation of p_{45} by different sensors using two methods: (a) ASMA, and (b) AEMA 18

Figure 2.3 Comparison of the estimation of the GAC using the ASMA and AEMA method for each sensor. The gray/light blue lines show different ASMA/AEMA estimations with different initial conditions of $\hat{P}$. The black/dark blue lines are the mean values over the number of broadcast cycles. 20

Figure 3.1 The topology of the digraph of the MAS of Example 1. Node 0, shown in gray, represents the leader; other nodes are followers. 30

Figure 3.2 The norm of the synchronization error signal when $u_0 = 0$ 32

Figure 3.3 The synchronization error signal for three states when $u_0 = 8\sin(5t)$ 33

List of Tables

Table 2.1 Comparison of the MSE of three different methods 19

Table 2.2 Comparison of the MSE of NAE with different window lengths 19

Chapter 1

Introduction

1.1 Motivation

Multi-agent systems (MASs) have attracted much attention recently in different fields due to their importance in a wide range of engineering and scientific applications such as smart grids, transportation systems, biological systems, power systems, etc. [1, 2, 3, 4, 5, 6]. One of the main objectives of this type of system is to properly coordinate individual agents to achieve a global objective collectively [7]. Some of the important problems in a MAS include formation, rendezvous, flocking, and consensus [8, 9, 10, 11, 12, 13]. In particular, in the consensus problems, it is desired that all agents converge to the same value, which could be the leader's state in a leader-follower configuration [9, 10, 14]. On the other hand, the formation problem is concerned with coordinating every agent's motion to maintain a specific relative position with respect to the other agents (see [12, 13]). Several distributed control strategies are proposed in the literature for MASs with a small or large number of agents, linear or nonlinear agent dynamics, and fixed or switching topology [15, 16, 17, 18].

Consensus protocols for MASs are distributed control rules designed to achieve a prescribed objective. For example, it is desired to perform data aggregation tasks in a sensor network. The connectivity of the graph representing the network is a crucial requirement for achieving any global objective [19]. Network connectivity is closely related to the information flow between the nodes. A graph is said to be connected if there is a directed path from any node to any other node in the network. When the information exchange between each pair of neighboring nodes in the network is bi-directional, the network is symmetric, and the graph representing it, is undirected. On the other hand, if the information flow between the nodes is not symmetric, the corresponding graph is directed and is referred to as a digraph. An underwater acoustic sensor network and traffic flow network of a city are examples of asymmetric networks [19, 20, 21, 22]. This thesis is mainly concerned with MASs with asymmetric communication channels, and hence, directed graphs.

A static graph is often used to model a deterministic MAS. However, the communication links in a real-world MAS are subject to uncertainty [23, 24, 25]. As a result, the convergence of a consensus algorithm may be negatively impacted if the uncertainty in the communication channels is not taken into consideration. Therefore, a random graph is employed to model a MAS with uncertain links [19].

1.2 Literature Review and Preliminaries

Over the past two decades, various problems in MASs have been investigated, such as consensus, rendezvous, flocking, and formation problems [9, 10, 11, 12, 13, 14]. In a graph-theoretic approach, agents in a MAS are represented by vertices and connections between them by edges [26, 27]. In these types of systems, it is desired to design a local control protocol for each node to achieve a global objective described above by minimal information exchange [28]. In particular, leader-following consensus, also known as coordinated tracking, is of interest to researchers in various fields [9, 16, 18, 29, 30]. This problem is more challenging than leaderless consensus and has more applications [23]. For example, leader-follower behavior is observed in biological systems such as the flock of birds and the school of fish [31]. Most of the existing studies in the literature consider a deterministic time-invariant topology for the leader-follower structure. Although there are some studies on time-varying leader-follower MASs, still many important challenges need to be addressed [29].

Like any real-world system, a MAS is subject to different types of uncertainty in the environment or system's dynamic [14, 32]. For example, in an uncertain environment, the communication links between agents are subject to change. Different strategies are introduced in the MAS literature to deal with an uncertain environment. In [33], the uncertainties are formulated using a neural network. The concept of a random graph model was first introduced by Erdös and Rényi [34, 35, 36, 37]. They used random variables to represent the communication links. This concept was then further developed in [38, 39]. Such models effectively describe networks with unpredictable communication links, e.g., underwater sensor networks with different sources of uncertainty like the water temperature, sound speed, and underwater currents [40]. The authors in [19, 40, 41] use a random graph to model a network with uncertain channels, where a weighted edge in the graph represents the probability of the corresponding link in the network. Consensus algorithms based on a random graph process have also attracted considerable interest. In [42], the authors investigate the consensus of linear dynamics using communication graphs, defined as a series of independently, identically distributed (iid) random graphs and prove its convergence. In the mentioned paper, the communication links between agents are undirected and not weighted, so all edges have the same probability of existence. They then generalized the results to random digraphs in [43] and to random weighted digraphs in [44], where different communication links do not necessarily have the same probability of existence. The authors in [45] introduce a necessary and sufficient condition on almost sure asymptotic consensus for iid random graphs. They show that a stochastic discrete-time linear dynamical system almost surely converges to a fixed vector under this condition. The authors in [46, 47] investigate distributed average consensus in sensor networks with quantized data and random link failures, and the authors in [28] study the distributed dynamic average consensus for asymmetric networks.

As noted before, network connectivity is an essential requirement for implementing any distributed algorithm in a MAS. Furthermore, given its importance in data propagation, a quantitative measure of connectivity would be beneficial in designing a high-performance MAS. Note that the information is propagated more efficiently in a more-connected network [48]. The in-network data propagation in random networks, where random variables represent communication links, is highly dependent on the underlying expected communication graph's connectivity level [45]. The Fiedler value, defined as the smallest nonzero eigenvalue of the Laplacian matrix of an undirected graph, is a commonly used algebraic measure of connectivity for symmetric networks. This measure is closely related to the convergence rate of any consensus algorithm applied to the network [49]. The paper [50] proposes a distributed approach for estimating and controlling the Fiedler value in ad-hoc networks with a random topology, but it is only applicable to symmetric networks. Note that random networks with bi-directional links, which are the focus of this thesis, are asymmetric. The Fiedler's notion of algebraic connectivity is extended to asymmetric networks in [51], where it is shown that many properties of Fiedler's definition apply to asymmetric networks as well. In [52], the magnitude of the smallest non-zero eigenvalue of the Laplacian matrix is proposed as algebraic connectivity for asymmetric networks. The authors in [53] generalize the algebraic connectivity measure to asymmetric networks by introducing the notion of generalized algebraic connectivity (GAC) and studying its relationship with various graph characteristics. They also show that the expected convergence speed of cooperative algorithms is highly dependent on the GAC.

1.3 Thesis Contributions

In this thesis, different algorithms are developed to estimate the existence probabilities of edges in a random graph which are required for assessing the GAC of the network. The proposed algorithms are based on data propagation in a strongly connected network and Q-learning update rules. These algorithms use the local knowledge of each agent to estimate the configuration of the network and then send these estimates to other agents. The main objective is to find the expected graph configuration and then evaluate the network's connectivity to assess the speed of convergence to consensus. This work generalizes the result of [19] presented for networks with time-invariant (or slowly varying) weights.

Another contribution of this work is concerned with consensus control in a MAS subject to uncertainty. First, the upper bounds on time-varying elements of the probability matrix are derived. Then, the uncertainty is formulated in the continuous-time domain as a bounded disturbance. The probability matrix is subsequently modeled as the sum of two matrices, one containing the nominal values and the other containing the time-varying elements (due to disturbances). A robust distributed control approach for a leader-follower MAS with time-varying edge probabilities is presented using the probability matrix of the expected digraph and linear matrix inequalities (LMIs). The proposed distributed control technique can be applied to any asymmetric leader-follower MAS with time-varying weighted links subject to bounded uncertainty. It is demonstrated that under this consensus algorithm, the consensus error decreases, as confirmed by simulations.

1.4 Thesis Layout

The thesis follows the following structure.

- **Chapter 1** includes the problem statement and motivation, a literature survey on consensus control, random graphs and network connectivity, and the overview of the current work.

- **Chapter 2** introduces notations used in this thesis and two adaptive Q-learning-based algorithms to estimate the expected graph of a MAS. These algorithms are used to estimate the probability matrix, which is required for assessing the GAC. The two algorithms are compared in terms of their performance in different scenarios. The simulation results confirm the effectiveness of the proposed methods compared to existing results.

- **Chapter 3** introduces a distributed algorithm for the random digraph representing a leader-follower MAS. It is assumed that the environmental uncertainty is time-varying, and hence, so is the probability of existence of edges. Each element in the probability matrix is considered as the sum of fixed (nominal) and time-varying (uncertain) terms. A novel robust control strategy is then proposed to achieve a satisfactory leader-follower agreement. The global consensus error is guaranteed to satisfy the global L_2-gain performance described in the chapter, in the presence of uncertainty. Numerical examples are provided to validate the theoretical findings in this chapter.

- **Chapter 4** presents the conclusions as well as some suggestions for future research directions.

Chapter 2

Random Graphs Estimation using Q-Learning

2.1 Overview

To correctly estimate a random graph it is important to be able to estimate, in a distributed fashion, the probability matrix (characterizing the probability of the existence of the graph's edges) and the graph connectivity. In this chapter, first, by leveraging Q-learning arguments, two different solutions to the probability estimation problem are proposed. Then, a method for the estimation of the algebraic connectivity is given. The accuracy of the proposed methods are verified by simulation for an underwater sensor network.

This chapter is based on the following publication:

M. Babahaji, S. Blouin, W. Lucia, M. M. Asadi, H. Mahboubi and A. G. Aghdam, "Random graphs estimation using Q-learning", In: *2021 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)* (2021), pp. 109-114.

2.2 Introduction

The notion of random graphs was first introduced in [34] and then developed in [38, 39]. Such graphs are used to emulate a network under uncertain conditions with unpredictable network connections e.g. underwater sensor networks with different uncertainty resources like: the water temperature, sound speed, and underwater currents [40, 41].

The connectivity of a graph is particularly important because it is closely related to the efficiency of data propagation [48] and to the solution of consensus problems. The concept of the algebraic connectivity of symmetric graphs was introduced in [54] as the second smallest eigenvalue of the Laplacian matrix. However, this definition can not be applied to random graphs, since they might be asymmetric. In [55], a bound for the algebraic connectivity of directed graphs (digraphs) is suggested. The authors in [56] investigated the connection between the Laplacian and adjacency matrices for random graphs. In [53], a more general form of the algebraic connectivity called the generalized algebraic connectivity (GAC) which can be applied to the asymmetric graphs, is introduced for random graphs. This is the method used in this research.

To compute the GAC, the adjacency matrix or the probability matrix is required. In addition, an algorithm is needed to estimate the probability matrix of the network in order to estimate the network configuration. For random graphs, computing probability matrix is needed to estimate the Laplacian matrix. The authors of [19] developed an algorithm based on Q-learning to estimate the probability matrix for random graphs using local knowledge. The algorithm is reliable for networks with slowly changing weights.

In this chapter, the estimation of the probability matrix of random graphs is performed by each sensor. The goal is to use the local knowledge of the sensors from their neighborhood to approximate the $\hat{\mathcal{G}}$. We propose adaptive algorithms to estimate the probability matrix of a time-varying random graph, which is then used to compute the GAC. This problem is particularly important when the dynamics' model of the environment is unknown and there are uncertainties in the environment like underwater conditions.

The rest of the chapter is as follows: notations and preliminaries as well as the problem formulation are introduced in Section 2.3, the adaptive algorithms for estimating the probability matrix are proposed in Section 2.4. In Section 2.5, the connectivity assessment for a digraph is explained. Simulation results are provided in Section 2.6, and the conclusion is provided in Section 2.7.

2.3 Preliminaries and Notations

In this section, first we introduce the notation, then some important graph theory concepts [19, 40, 53, 57] used in the rest the chapter are reviewed. Let $\mathbb{R}$ and $\mathbb{C}$ be the set of real and complex numbers. Furthermore, for any positive integer n, $\mathbb{N}_n$ denotes the restricted set of natural numbers $\{1, \ldots, n\}$. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a random graph, where $\mathcal{V} = \mathbb{N}_n$ and $\mathcal{E}$ are referred to as the set of nodes and edges, respectively. The probability matrix, $P = [p_{ij} | 0 \leq p_{ij} \leq 1]$, represents the existence probability of the edge $(j, i) \in \mathcal{E}$, and $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$ is the adjacency matrix, where

$$a_{ij} = \begin{cases} 1 & \text{with probability } p_{ij} \\ 0 & \text{with probability } 1 - p_{ij} \end{cases} \tag{1}$$

Definition 1 *The directed graph (digraph) $\hat{\mathcal{G}} = (\mathcal{V}, \hat{\mathcal{E}})$ is defined as the expected digraph $\mathcal{G}$, where $\mathcal{V}$ contains the same vertices as in $\mathcal{G}$, and $\hat{\mathcal{E}}$ represents the set of edges. The adjacency matrix of $\hat{\mathcal{G}}$ is equal to the probability matrix $\hat{\mathbf{A}} = [p_{ij}]$ for each i and j in $\mathcal{V}$. Therefore, $\hat{\mathcal{E}}$ can be formulated as*

$$\hat{\mathcal{E}} = \{(i, j) \in \mathcal{V} \times \mathcal{V} \mid p_{ji} \neq 0\} \tag{2}$$

If the digraph is time-varying (i.e., the elements of the probability matrix change over time), the expected digraph is denoted as $\hat{\mathcal{G}}(t)$ and its adjacency matrix as $\hat{\mathbf{A}}(t)$.

Let $L = [l_{ij}] \in \mathbb{R}^{n \times n}$ denote the Laplacian matrix of a graph. For a weighted digraph, L is a real matrix whose entries are [53]

$$l_{ij} = \begin{cases} -p_{ij}, & \text{if } (j, i) \in \mathcal{E} \\ \sum_{k \neq i} p_{ik}, & \text{if } j = i \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Furthermore, the degree matrix is defined as $\mathcal{D}$, which for an expected graph is equal to $\mathcal{D} = diag\{\sum_{j=1}^{n} p_{1j}, ..., \sum_{j=1}^{n} p_{nj}\}$. The neighbor set of node i, denoted by $\mathcal{N}_i$, is the set of all nodes from which there is an incoming edge to node i.

Definition 2 *Let* $\mathbf{A}$ *be an asymmetric matrix,* $\varphi(\mathbf{A}) = \{(\lambda_i(\mathbf{A}), \mathrm{v}_i(\mathbf{A}), \mathrm{w}_i(\mathbf{A}))| \, i \in \mathbb{N}_n\}$ *defined as a triple comprised of* $\lambda_i \in \mathbb{C}$*, which is the* i^{th} *eigenvalue of* $\mathbf{A}$*,* $\mathrm{v}_i \in \mathbb{C}^n$ *and* $\mathrm{w}_i \in \mathbb{C}^n$ *which are the* i^{th} *associated left and right eigenvectors, respectively.* $\varphi(\mathbf{A})$ *is sorted in increasing order according to the real parts of the eigenvalues. Therefore,* $Real(\lambda_1(\mathbf{A})) \leq Real(\lambda_2(\mathbf{A})) \leq ... \leq Real(\lambda_n(\mathbf{A}))$.

2.4 Probability Matrix Estimation

The authors in [19] proposed an algorithm based on Q-learning to estimate the probability matrix of an expected graph. The probability estimation algorithm presented in [19] is a non-adaptive estimation algorithm (NAE). As experimentally is shown there, such a solution is not designed to deal with unknown fast variations of $P(t)$. The main drawback is that the proposed approach is a NAE, and its learning rate is constant, while the rate of change of $P(t)$ can change over time.

Let $\hat{P}(t)$ be the estimation of $P(t)$ for the expected graph $\hat{\mathcal{G}}(t)$ at the t^{th} broadcast cycle. $\hat{P}^i(t)$ represents the estimation of $\hat{P}(t)$ by sensor i. The row j and column k of $\hat{P}^i(t)$ shows the probability of existence of $(k, j) \in \mathcal{E}$ estimated by sensor i.

After computing the probability matrix from each sensor, the sensors broadcast their expectation of $P(t)$. Therefore, sensor i can estimate the edges that are not in its vicinity. The nodes communicate with each other by broadcasting the information. To avoid data interference, we assume that each broadcast cycle comprises of n time slots, and during each time slot, only one node is allowed to transmit data. For more information about the broadcasting algorithms refer to [19].

Let τ be the most recent broadcasting cycles (also known as time window) considered to update $\hat{p}^i_{kj}$, and w be its length (i.e., $\tau = \{t_1, ..., t_w\}$). The Q-learning estimation is based on a greedy algorithm. In other words, for updating the estimation using Q-learning we need to define a learning rate. This learning rate shows which portion of learning is based on exploration or exploitation.

$0 \leq \alpha \leq 1$ is the learning rate of the Q-learning update rule in this work. $\hat{P}^i(t)$ is updated using the number of data which sensor i receives during the previous window τ. This element is computed using the moving average approach. Moving average is a measure that uses a sequence of averages of various subsets of the entire data set to analyze data points.

In what follows, two different scenarios are discussed to motivate why α and w should be selected adaptively for an accurate estimation. Assume the number of data transferred from sensor j to i changes suddenly; the algorithm should learn fast enough to be able to estimate it correctly. In this case, if the estimation algorithm is configured to use a small window length and a large learning rate, then fast convergence to the new probability value will occur. This choice of parameters would reduce the impact of older data so that learning occurs mostly based on newer data. On the other hand, when the rate of change of $\hat{p}^i_{jk}$ is slow, there is no need to learn fast and update quickly. In this case, the learning rate could be low (near zero) and window size large to be more robust in the presence of noise. As a consequence, $\hat{p}^i_{ij}$ relies more on its previous values and it is less sensitive to noise. The above discussion implies that different values of w and α might be used in various scenarios. It is important to note that when the α is large and w is small, however the system tracks the changes faster, it is more vulnerable to noise and if the learning rate is too high and the window size too small, the estimates might follow the noise more than the actual probability.

In this chapter, we develop two algorithms based on two moving average methods to estimate $P^i(t)$. In the first one, estimation is performed using a simple moving average (SMA) over the receiving data, while in the second algorithm, an exponential moving average (EMA) is used to improve the estimation when the probability matrix changes rapidly. These two approaches are explained in the next two subsections.

2.4.1 Probability estimation via SMA

Denote by n_{ij} the number of times the node i receives uncorrupted information from node j within the time window τ, i.e.,

$$n_{ij} = \sum_{t \in \tau} n^t_{ij} \tag{4}$$

where n_{ij}^t is the number of received data points by node i from node j at time $t \in \tau$. According to the SMA scheme, all the data have the same weight (irrespective of time), and the probability of existence of (j, i) is estimated as

$$\sigma(t) = \frac{n_{ij}}{w} \tag{5}$$

where σ represents the SMA of the number of information received by node i from node j. At the beginning, the initial estimate is only based on the number of messages node i receives in the broadcast cycle from parent nodes (the nodes in the vicinity). Following a receding horizon strategy, the window is moved one step forward at the next broadcast cycle, and the operations (4)-(5) are repeated. Then the previously computed probability $p_{ij}^i(t)$ is updated according to an adaptive Q-learning paradigm as

$$\hat{p}_{ij}^i(t) = (1 - \alpha(t))\hat{p}_{ij}^i(t-1) + \alpha(t)\sigma(t) \tag{6}$$

where the first component is known as the "exploitation" term, and the second one as the "exploration" term. The rate-of-change ($r(t)$) of the probability matrix is derived after updating each estimation. This allows the learning rate and window length to be adjusted at the next iteration by $f_1(r(t))$ and $f_2(r(t))$, respectively. The functions $f_1(\cdot)$ and $f_2(\cdot)$ need to be properly defined based on the application. They should be designed such that a larger $r(t)$ results in a larger $\alpha(t)$ and a smaller $w(t)$, and vice versa. In the simulation part of Section 2.6 a possible choice for $f_1(\cdot)$ and $f_2(\cdot)$ is given for a specific graph. All the mentioned steps are collected in Algorithm 1. In the sequel, such an estimation scheme is referred to as the adaptive SMA (ASMA).

Algorithm 1: Estimation of $\hat{P}^i(i, :)$ using the ASMA

```
Initial values ← p̂^i_{i:}(0), α_0, w_0, r_0;
for each node j ∈ N_i do
    if t ≤ w_0 then
        p̂^i_{ij}(t) = n_{ij} / t;
    else
        α(t) = f_1(r(t − 1));
        w(t) = ⌈f_2(r(t − 1))⌉;
        p̂^i_{ij}(t) = (1 − α(t)) p̂^i_{ij}(t − 1) + α(t)σ(t);
    end
    r(t) = p̂^i_{ij}(t) − p̂^i_{ij}(t − 1);
end
```

2.4.2 Probability estimation via EMA

Unlike the SMA method, the EMA approach uses a weighting scheme that gives more importance to recent data. In particular, $S_{ij}(t)$ is the assigned weight to the number of data sets received by node i from node j at the time $t \in \tau$ and it is equal to

$$S_{ij}(t) = \eta n_{ij}^{t} + (1-\eta)(S_{ij}(t-1) - (1-\eta)^{(t-w)} n_{ij}^{t-w}) \tag{7}$$

where $0 \leq \eta \leq 1$ is the smoothing factor which defines how fast the weighting factor decays with time.

When $\eta = 1$, $S_{ij}(t)$ is updated only based on the new data, and the data from all the previous cycles are not considered. On the other hand, when $\eta = 0$, $S_{ij}(t)$ updates only according to the previous cycles and ignores the new data. The EMA, denoted by μ, is defined as

$$\mu(t) = \frac{S_{ij}(t)}{D(t)} \tag{8}$$

where $D(t)$ is defined as

$$D(t) = 1 + (1-\eta)(D(t-1) - (1-\eta)^{(t-w)}) \tag{9}$$

Algorithm 2 shows the resulting adaptive estimation scheme with the EMA. Such a scheme is hereafter referred to as the adaptive EMA (AEMA). Let t_1 represent the time steps that are less than t.

2.4.3 Comparing algorithms

In the NAE algorithm, the updating rule is similar to the one used in Algorithm 1, with the important difference that the values of w and α are time-invariant. Therefore, the NAE algorithm does not have enough flexibility to deal with changing environments. Algorithms 1, 2, and the NAE algorithm employ two low-pass filters. The first one is used to compute the number of transmitted data from j to i and the second one to update $\hat{p}_{ij}^{i}$. However, while Algorithm 1 and NAE use a SMA low-pass filter, Algorithm 2 uses an EMA low-pass filter.

Algorithm 2: Estimation of $\hat{P}^i(i,:)$ using the AEMA

```
Initial values ← p̂^i_{i:}(0), S_ij(0), D(0), α_0, w_0, r_0;
for each node j ∈ N_i do
    if t ≤ w_0 then
        for t_1 ≤ t do
            S_ij(t_1) = (1 − η)S_ij(t_1 − 1) + η n_ij^{t_1};
            D(t_1) = (1 − η)D(t_1 − 1) + 1;
        end
        p̂^i_ij(t) = S_ij(t) / D(t);
    else
        w(t) and α(t) are updated similarly to Algorithm 1.
        p̂^i_ij(t) = (1 − α(t)) p̂^i_ij(t − 1) + α(t)μ(t);
    end
end
```

According to (7), the following advantages and drawbacks can be underlined for the use of EMA: when the value of η is close to 1, AEMA tracks the changes faster compared to ASMA and the NAE algorithms, as they smooth out sudden changes. This property makes Algorithm 2 more sensitive to noise. However, both proposed algorithms outperform the NAE algorithm in terms of accurate tracking because they can adjust the variables (α and w) to achieve the best result. Moreover, faster tracking is expected for ASMA and AEMA by choosing α and w, respectively, close to one and zero. On the other hand, such a tuning has the drawback of making the estimation algorithms sensitive to noise. Further work is required in order to tune the parameters more effectively. Both EMA and SMA methods are elaborated in terms of noise-attenuation capabilities in [58].

2.4.4 Estimation of the entire probability matrix

As noted before, sensor i estimates the probability of existence of edge $(k, i) \in \mathcal{E}$, where k is any of the graph nodes. If node k is connected to node i, it can easily use the estimates of node i of its connected edges. However, there is no guarantee that sensor k will always receive sensor i's broadcasts [19]. As a result, sensor k will update its estimate of the existence probability of edge (j, i) in the expected graph $\hat{\mathcal{G}}^i(t)$ based on the estimated values performed by all sensors, given the connectivity of the graph. Each sensor receives the estimation of every edge's probability from different sensors at any broadcast cycle and find the most appropriate estimation.

Definition 3 *[19] Time stamp matrix $Q^k = [q^k_{ij}]$ is defined as a matrix that holds the broadcast cycle q^k_{ij} in which sensor i generated the current estimate of p^k_{ij} held by sensor k.*

The following algorithm is used to update the rows $1, \ldots, i-1, i+1, \ldots, n$ of matrix $\hat{P}^i(t)$. It is assumed that the edges with a probability smaller than a prescribed threshold value ϵ are removed.

Algorithm 3: Computing the $\hat{P}^i(: /\{i\}, :)$

for $(k, j) \in \hat{\mathcal{E}}^i(t)$ & $j \neq i$ **do**
  $s^* = \text{argmax}_{s \in \{i\} \cup \mathcal{N}_i} q^s_{jk}$;
  $q^i_{jk} = q^{s^*}_{jk}$;
  if $p^{s^*}_{jk} \geq \epsilon$ **then**
    $p^i_{jk}(t) = p^{s^*}_{jk}$;
  else
    $p^i_{jk}(t) = 0$;
    $\hat{\mathcal{E}}^i(t) = \hat{\mathcal{E}}^i(t)/\{(k, j)\}$
  end
end

2.5 Connectivity Assessment

Since the random graphs considered in this study are asymmetric, the notion of algebraic connectivity for symmetric graphs cannot be applied here. In fact, unlike symmetric networks, the eigenvalues of the Laplacian of an asymmetric network can be complex. Therefore, a more general form of connectivity is needed for directed graphs. The generalized algebraic connectivity (GAC) for weighted directed graphs is introduced in [53] to address this shortcoming. This notion reflects the convergence rate to consensus in a directed graph, similar to undirected graphs. The GAC is defined as the real part of the second smallest eigenvalue of the Laplacian matrix of a connected directed graph, as described next.

Definition 4 *[53] Let L be the Laplacian matrix of the expected digraph $\hat{\mathcal{G}}$ given by (3). The GAC is defined as*

$$\widetilde{\lambda}(L) = \min_{\lambda_i(L) \neq 0, \lambda_i(L) \in \Lambda(L)} \textit{Real}\,(\lambda_i(L))$$

where $\Lambda(L) = \{\lambda_i(L) | i \in \mathcal{V}\}$.

For an unknown graph $\mathcal{G}$, after estimating the probability matrix of $\hat{\mathcal{G}}$ via Algorithm 3, the Laplacian matrix is given by (3). It can be rewritten as

$$L^i(t) = P^i(t) - \mathcal{D}^i(t)$$

where $\mathcal{D}^i(t)$ is the estimated degree matrix by i at broadcast cycle t. Then, each node can compute the GAC using Definition 4, as an estimate of network connectivity. These steps are summarized in Algorithm 4.

Algorithm 4: Estimation of the GAC

```
for each node i do
    for node j ∈ N_i do
        Compute p^i_ij(t) using Algorithm 1 or 2;
    end
    Broadcast the expectations using Algorithm 3 to estimate the other rows of P^i(t);
    L^i(t) = P^i(t) − D^i(t);
    λ̃(L^i(t)) = min_{λ_i(L)≠0, λ_i(L)∈Λ(L)} Real(λ_i(L));
end
```

2.6 Simulation Results

To evaluate the effectiveness of the proposed algorithm, we have considered an underwater sensor network consisting of 5 sensors. The simulation results are obtained in MATLAB/Simulink environment. To this end, the probabilities of the existence of edges are assumed to evolve as follows:

$$P(t) = \begin{bmatrix} 0 & 0.9 & 0 & 0.8 & 0 \\ 0.6 & 0 & 0.6 & 0 & 0.8 \\ 0 & 0.6 & 0 & F_1(t) & 0 \\ 0.8 & F_2(t) & 0.8 & 0 & F_3(t) \\ 0 & 0.7 & 0 & 0.9 & 0 \end{bmatrix} \tag{10}$$

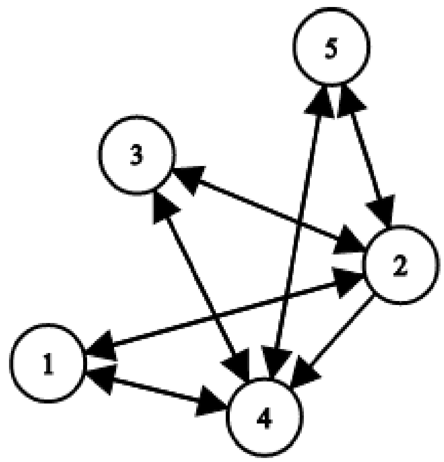


Figure 2.1: The topology of directed graph

where $F_1(t)$, $F_2(t)$, and $F_3(t)$ are the time-varying elements of the probability matrix, and are equal to

$$\begin{bmatrix} F_1(t) \\ F_2(t) \\ F_3(t) \end{bmatrix} = \begin{bmatrix} 0.4 + 0.2\sin(0.005t) \\ 0.6 + 0.1\sin(0.005t) \\ 0.5 + 0.2\sin(0.01t) \end{bmatrix} \tag{11}$$

The frequencies of the above sinusoids are chosen in accordance with real-world underwater applications. The broadcast cycle length is considered $T = 5$ seconds, and the length of each time interval is 1 second. Therefore, during each one-second interval, only one sensor can broadcast information. The update function α is considered as

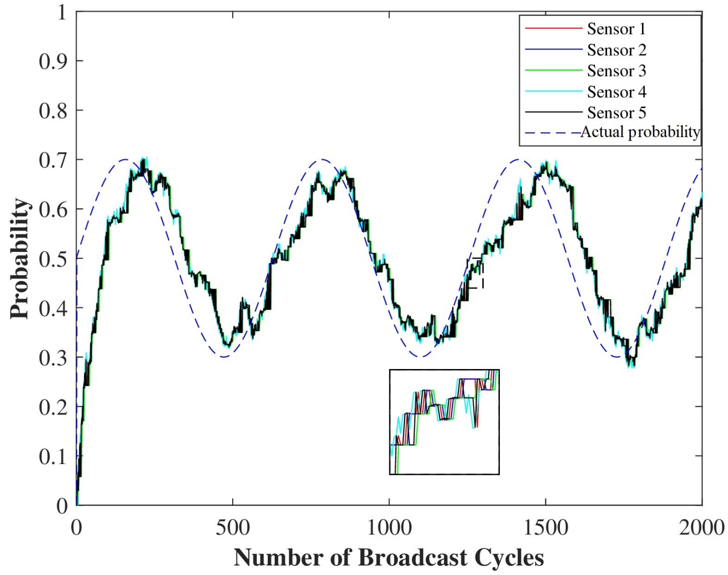
$$\alpha(t) = f_1(r(t-1)) = \begin{cases} 0.99 & \text{if } r(t-1) \geq 0.99 \\ 0.5 & \text{if } 0.8 \leq r(t-1) < 0.99 \\ 0.03 & \text{if } 0.001 < r(t-1) < 0.8 \\ 0.02 & \text{if } r(t-1) \leq 0.001 \end{cases} \tag{12}$$

and w is given by

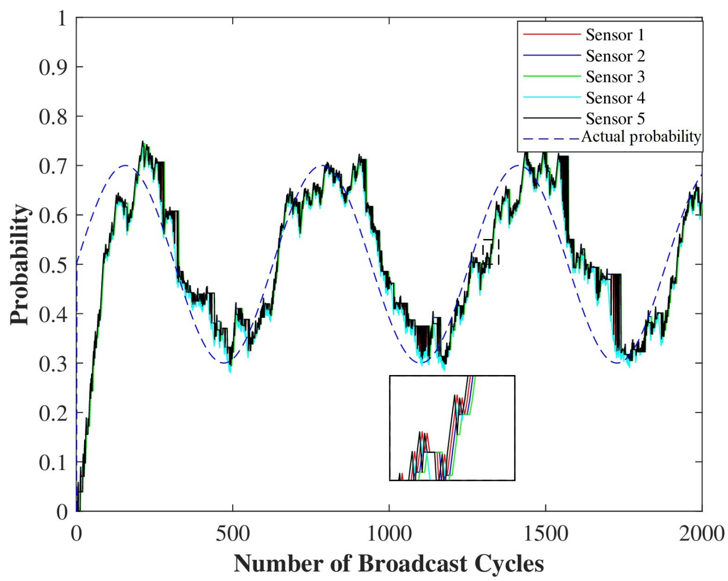
$$w(t) = f_2(r(t-1)) = \begin{cases} 1 & \text{if } r(t-1) \geq 0.9 \\ 2 & \text{if } r(t-1) < 0.9 \end{cases} \tag{13}$$

These two functions meet the required conditions. If the environment varies rapidly, a sufficiently large α (close to 1) and small w must be selected, and vice versa. The mean-square error (MSE) of the probabilities estimated by sensor 5 using Algorithms 1 and 2 for different parameters are shown in Table 2.1. First, it is important to remark that some delay in the estimation are unavoidable due to the delay to receive broadcasted information from other sensors. As an example, from (10) it can be noticed that sensor 5 directly receives information only from 2 and 4, while the information pertaining to the other sensors are delayed. In this simulation, NAE, ASMA, and AEMA are compared. It is predictable that the adaptive Algorithms 1 and 2 almost always have better result than the NAE, as this algorithm uses a fixed window and learning rate, but in the two other ones, the window size and learning rate are changing according to rate-of-change. This is shown in Table 2.1 where the NAE is used with different learning rate and in Table 2.2, where it is tested for a fixed $\alpha = 0.1$ and different window length. The obtained results show that the AEMA and ASMA methods in all cases result in a smaller MSE compared to NAE.

Fig. 2.2 illustrates how p_{45} is estimated by Algorithm 3 based on Algorithms 1 and 2. The ASMA method relies on the previous values, as much as the new ones. This attitude causes a delay in reacting to the parameter variations and makes it smoother than the AEMA method, where the estimator relies more on the latest data. This leads to a faster but more fluctuating estimator. In both cases, the learning rate and window size are updated using (12) and (13) and react faster to rapidly varying data. Fig. 2.2 demonstrates the effect of the distance and connectivity. Sensor 4, which directly computes p_{45} using Algorithms 1 and 2 provides the most accurate estimate of this probability of p_{45} to its actual value. Sensors 5 and 1, which are connected to sensor 4 with weights 0.9 and 0.8, respectively, have the second fastest reaction to the change of p_{45}. Sensor 2, which receives the data directly from sensor 1 but not from sensor 4, has some delay in updating its estimation. As for sensor 3, on the other hand, there is an edge from it to sensor 4, but its weight is relatively small, which means that this edge is not reliable, and that is why it has the most considerable delay in tracking the changes.



(a)



(b)

Figure 2.2: Comparison of the estimation of p_{45} by different sensors using two methods: (a) ASMA, and (b) AEMA

Table 2.1: Comparison of the MSE of three different methods

NAE ($\alpha = 0.01$)	NAE ($\alpha = 0.1$)	NAE ($\alpha = 0.3$)	NAE ($\alpha = 0.5$)	NAE ($\alpha = 0.7$)	ASMA	AEMA
0.096	0.081	0.123	0.2576	0.2706	0.035	0.028

Table 2.2: Comparison of the MSE of NAE with different window lengths

NAE (w=1)	NAE (w=2)	NAE (w=3)	NAE (w=4)
0.0502	0.081	0.139	0.199

Fig. 2.3 compares the GAC computed using the ASMA and AEMA methods from the perspective of each sensor with different initial choices for $\hat{P}(0)$: $0.3I_5, 0.6I_5$, and $0.8I_5$. This figure demonstrates that the EMA method converges faster to the actual GAC. The reason is that the ASMA method seeks to smooth out the changes; therefore, it takes longer to converge to the actual GAC. On the other hand, the AEMA method uses the most recent information. Note that since there are five time-steps in each broadcast cycle, at each time step, only one sensor updates its estimation using the other sensors' estimations, and then broadcasts it. For instance, sensor 5 receives all other sensors' estimations and then updates the entries of $\hat{P}^5(t)$. As a result, the estimation of sensor 5 would be the nearest to the actual value because it is the last sensor. However, both methods show relatively good tracking properties, in general.

2.7 Conclusions

In this work, we estimate the probability matrix of a random graph using two different procedures. These algorithms differ in terms of how the moving average is computed, and they have similarities in terms of adaptive learning rate and window length. As an application of estimating a random graph, the expected graph based on the estimated probability matrix is used to assess the GAC. The two proposed methods (ASMA and AEMA) are applied to a network of 5 sensors in the simulation section. Then, the impact of the connectivity between sensors on the delay in the estimation is discussed. The proposed procedures outperform the previously proposed NAE in most scenarios due to their adaptive nature. As future work, one can investigate how the proposed

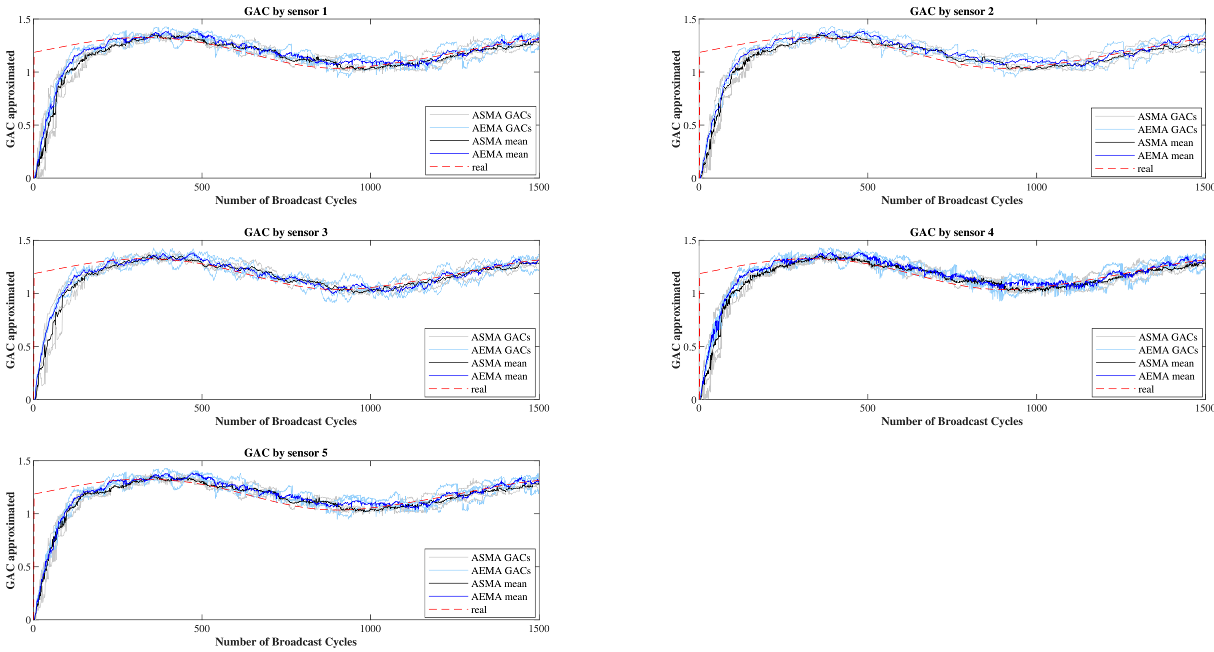


Figure 2.3: Comparison of the estimation of the GAC using the ASMA and AEMA method for each sensor. The gray/light blue lines show different ASMA/AEMA estimations with different initial conditions of $\hat{P}$. The black/dark blue lines are the mean values over the number of broadcast cycles.

algorithms perform when applied to a real underwater sensor network subject to packet loss.

Chapter 3

Consensus Control of Multi-Agent Systems with Uncertain Communication Links

3.1 Overview

This chapter is devoted to the consensus control of a leader-follower multi-agent system (MAS) modeled by a random digraph. Due to the uncertainties in the environment, the communication links between distinct agents are represented by a time-varying probability matrix. We propose a novel robust control strategy to achieve approximate agreement among agents in the presence of an asynchronous network with uncertain communication links. The uncertainties are modeled as disturbances, and the controller is designed to debilitate them. Under the proposed controller, it is guaranteed that the global consensus error satisfies the L_2-gain performance in the presence of uncertainties. The designed controller consists of two components: one for time-varying links and the other for the nominal links. The functionality of the proposed controller is proved mathematically and verified by simulations.

This chapter is based on the following publication:

M. Babahaji, E. Firouzmand, A. G. Aghdam, and H. A. Talebi, "Consensus control of multi-agent systems with uncertain communication links", *submitted*.

3.2 Introduction

MASs have attracted much attention recently in different fields due to their importance in a wide range of engineering and scientific applications such as smart grids, transportation systems, biological systems, power systems, etc. [59, 60]. Some of the important problems in a MAS include formation, rendezvous, flocking, consensus, etc. [9, 10, 11, 12, 13]. In particular, leader-following consensus, also known as coordinated tracking, is of interest to researchers in various fields e.g. biological systems, vehicle platoon, etc. [31, 61].

Like any real-world system, a MAS is subject to different types of uncertainty in the environment or system's dynamic [14, 32]. For example, in an uncertain environment, the communication links between agents are subject to change. Different strategies are introduced in the MAS literature to deal with an uncertain environment. In [33], the uncertainties are formulated using a neural network. The concept of a random graph model was first introduced in [34]. They used random variables to represent the communication links. The authors in [40] use a random graph to model a network with uncertain channels, where a weighted edge in the graph represents the probability of the corresponding link in the network. The authors in [62] introduce a controller for MASs with uncertainties and delays where the dynamic model of each agent is a single integrator. The results are extended to the MAS with double-integrator dynamics in [32], and to the MAS with dynamics having at least one zero eigenvalues [14]. In a different application, a distributed H_∞ control law is proposed in [63] to model the cyber-physical attacks as external disturbances in order to mitigate their impact using control-theoretic methods. The approach does not impose any limitations on the number of attacked agents.

This chapter considers the leader-follower consensus problem for a random graph, where the leader is the root node of the directed spanning tree. Each follower only has access to its own local information and that of its neighbors, and estimates the network configuration using [64]. The probabilities of the existence of the edges in the graph are represented by a matrix, which is considered as the weight of the graph. A robust controller is subsequently designed to achieve the desired objectives, described in Section 3.4. One of the advantages of the proposed approach is that some of the restrictions imposed on the system dynamics in the prior literature (e.g. [14, 32, 62], discussed above) are not assumed here.

The rest of the chapter is organized as follows. The notations and preliminaries are introduced in Section 3.3. The problem is then formulated in Section 3.4. The control strategy for the random graph is proposed in Section 3.5. Simulations are presented in Section 3.6 to demonstrate the effectiveness of the proposed method. Finally, the concluding remarks are given in Section 3.7.

3.3 Preliminaries and Notations

In this section, some background information about the graph theory is provided [63, 64, 65]. Let $\mathbb{R}$ and $\mathbb{C}$ be the set of real and complex numbers, $\otimes$ be the Kronecker product, and I_N be a $N \times N$ identity matrix. Let also $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a random digraph with the set of vertices and edges denoted, respectively, by $\mathcal{V} = \{1, ..., N\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$ is defined as

$$a_{ij} = \begin{cases} 1 & \text{with probability } p_{ij} \\ 0 & \text{with probability } 1 - p_{ij} \end{cases} \tag{14}$$

where $0 \leq p_{ij} \leq 1$, $i, j \in \mathcal{V}$, are the elements of the probability matrix $\mathcal{P}$, which represents the existence probability of edge $(j, i) \in \mathcal{E}$.

The expected digraph $\hat{\mathcal{G}} = \{(\hat{\mathcal{V}}, \hat{\mathcal{E}}) |\ \hat{\mathcal{V}} = \mathcal{V},\ \hat{\mathcal{E}} \subseteq \hat{\mathcal{V}} \times \hat{\mathcal{V}}\}$ is defined as an estimate of the digraph $\mathcal{G}$, where $(\hat{\mathcal{A}} = [p_{ij}])$ for each i and j in $\mathcal{V}$. Note that the expected digraph $\hat{\mathcal{G}}$ is used to represent a random digraph in the analysis and design of networked control systems in the presence of uncertainties. Let the degree matrix $D = diag(d_i) \in \mathbb{R}^{N \times N}$ be the total number of the incoming edges of node i, where $d_i = \sum_{j=1}^{N} p_{ij}$, and the Laplacian matrix L be defined as $L = D - \hat{\mathcal{A}}$. For the case of a time-varying network, the time-dependency of the probability matrix is denoted as $\mathcal{P}(t)$. The degree matrix and Laplacian are then represented by $D(t)$ and $L(t)$, respectively. Also, $\mathcal{N}_i$ is the neighbor set of node i, which is the set of all nodes from which there is an incoming edge to node i.

3.4 Problem Formulation

Consider a MAS with linear time-invariant (LTI) dynamics as follows:

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t) \tag{15}$$

where $x_i \in R^n$ is the state, $A \in R^{n\times n}$, $B \in R^{n\times m}$ are the constant matrices (which are the same for all agents), and $u_i \in R^m$ is the control input.

Assumption 1 *There are $N+1$ nodes, N of them representing the followers with dynamics represented by (15), and one representing the leader.*

Assumption 2 *The leader is represented by the root node and does not receive information from the followers.*

Assumption 3 *Assume that the network of followers is strongly connected and that the leader sends its information to at least one follower.*

Assumption 3 is essential as the information of the entire network is not available to the agents. Hence, the agents need to exchange their estimation of the whole network with other agents. Note that due to the strong connectivity of the followers, each agent's perception about network configuration converges to the exact configuration.

Let the agent dynamics be rewritten as [63]:

$$\dot{x}_i(t) = Ax_i(t) + Bu_i^a(t) + Bf_i(t, \Delta p_{ij}(t)) \tag{16}$$

where $f_i(t, \Delta p_{ij}(t))$, $i \in \mathcal{V}$, is the uncertain term of the control signal in (15), which will hereafter be denoted by $f_i(t)$, for simplicity. Write the probability matrix $\mathcal{P}$ as $\hat{\mathcal{P}} + \Delta\mathcal{P}(t)$, where $\hat{\mathcal{P}}$ is the nominal component, and $\Delta\mathcal{P}(t)$ includes the time-varying elements, which are bounded. Note that ${u_i}^a(t)$ and $f_i(t)$ are, respectively, associated with the terms in $\hat{\mathcal{P}}$ and $\Delta\mathcal{P}(t)$.

Assumption 4 *$\|\Delta p_{ij}(t)\|$ is less than or equal to ψ_{ij} for any $p_{ij} \neq 0$, and is zero otherwise, where ψ_{ij} is a prescribed positive constant which is less than or equal to the nominal value $\hat{p}_{ij}$, and $\|.\|$ denotes the 2-norm.*

Let the agents' dynamics (16) be written in the augmented form as:

$$\dot{X}(t) = (I_N \otimes A)X(t) + (I_N \otimes B)U^a(t) + (I_N \otimes B)F(t, \Delta\mathcal{P}(t)) \tag{17}$$

where $X(t) = [{x_1}^T(t), \ldots, {x_N}^T(t)]^T$, $U^a(t) = [{u_1^a}^T(t), \ldots, {u_N^a}^T(t)]^T$, and $F(t) = [f_1^T(t), \ldots, f_N^T(t)]^T$.

Definition 5 *[63] The synchronization error between the leader and each follower is defined as*

$$\delta_i(t) = x_i(t) - x_0(t), \quad i \in \mathcal{V} \tag{18}$$

Bounded L_2-Gain Synchronization Problem [66, 63]: Given the system (17) with N nodes, design a distributed control protocol $u_i(t)$, such that for any $F(t) \in \mathrm{L}_2[0, \infty)$ the bounded L_2-gain condition is satisfied as follows:

$$\int_0^\infty \delta^T Q \delta dt \leq \gamma^2 \int_0^\infty F^T W F dt \tag{19}$$

where γ is a positive real scalar representing the disturbance attenuation level. In addition, Q and W are positive definite matrices.

It is desired to develop a control strategy that satisfies (19). This problem is studied in the next section.

3.5 Main Results

For a MAS described by (17), the static control protocol proposed in [14] is adopted as a dynamic law in the presence of uncertainties. For the case when the agents have time-invariant dynamics like (16), one can use the following time-varying control law:

$$u_i(t) = cK(\sum_{j \in \mathcal{N}_i} p_{ij}(t)(x_i(t) - x_j(t)) + p_{i0}(t)(x_i(t) - x_0(t))), \quad i \in \mathcal{V} \tag{20}$$

where $\mathcal{N}_i$ is the neighbor set of node i, $c > 0$ is the common coupling weight among neighboring agents, $K \in R^{m \times n}$ is the feedback gain matrix, and $p_{ij}(t)$ is the (i, j) element of the probability matrix. Furthermore, $p_{i0}(t)$ is the probability of existence of an edge from the leader to follower

i. This control law aims to minimize the disagreement between the leader and followers by using a proper feedback gain matrix. The control law $U(t)$ equals:

$$U(t) = c(\mathcal{L} \otimes K)\delta(t) \tag{21}$$

where $\mathcal{L} = L + diag\{p_{i0}\}_{i\in\mathcal{V}}$, and can be decomposed into a fixed nominal term and a time-varying deviation term as $\hat{\mathcal{L}} + \Delta\mathcal{L}(t)$. In addition,

$$\delta(t) = [\delta_1^T(t), \ldots, \delta_N^T(t)]^T \tag{22}$$

According to (16), $u_i(t)$ can be expressed as $u_i^a(t) + f_i(t)$, where

$$u_i^a(t) = cK(\textstyle\sum_{j\in\mathcal{N}_i} \hat{p}_{ij}(x_i(t) - x_j(t)) + \hat{p}_{i0}(x_i(t) - x_0(t))), \quad i \in \mathcal{V} \tag{23a}$$

$$f_i(t) = cK(\textstyle\sum_{j\in\mathcal{N}_i} \Delta p_{ij}(t)(x_i(t) - x_j(t)) + \Delta p_{i0}(t)(x_i(t) - x_0(t))), \quad i \in \mathcal{V} \tag{23b}$$

Based on (23),

$$U^a(t) = c(\hat{\mathcal{L}} \otimes K)\delta(t) \tag{24a}$$

$$F(t) = c(\Delta\mathcal{L}(t) \otimes K)\delta(t) \tag{24b}$$

Using the above equations

$$\dot{\delta}(t) = (I_N \otimes A + c\hat{\mathcal{L}} \otimes BK)\delta(t) + (I_N \otimes B)F(t) \tag{25}$$

where $A_c = I_N \otimes A + c\hat{\mathcal{L}} \otimes BK$.

The following lemma is essential in the proof of the main result.

Lemma 1 *[63] Let $\hat{\mathcal{L}}$ be such that $\hat{\mathcal{L}} = V\Sigma V^{-1}$, where $\Sigma = diag\{\lambda_i\}_{i\in\mathcal{V}}$ (λ_i's are the eigenvalues of $\hat{\mathcal{L}}$), and V is a matrix whose columns are the full complement of the corresponding eigenvalues.*

Then, $Q_2 \in R^{N\times N}$ is defined as

$$Q_2 = cR_1\hat{\mathcal{L}} \tag{26}$$

where $R_1 = R_1^T = (V^{-1})^T V^{-1}$ exists such that $Q_2 = Q_2^T > 0$.

Theorem 1 *Consider a MAS whose communication graph $\mathcal{G}$ satisfies Assumptions 1-4, and let agents' dynamics be described by (16). Let also the pair (A, B) be stabilizable, and apply the control law (20) with $K = -B^T P_1$ and $c > 1/2\min(Re\{\lambda_i\})$, where λ_i's, $i \in \mathcal{V}$, are the eigenvalues of $\hat{\mathcal{L}}$, and $P_1 \in R^{n\times n}$ is the solution of the following linear matrix inequality (LMI):*

$$A^T P_1 + P_1 A + Q_1 + (1 - \frac{1}{\gamma^2})P_1 BB^T P_1 < 0 \tag{27}$$

for a given symmetric positive-definite matrix $Q_1 = Q_1^T > 0$, $Q_1 \in R^{n\times n}$, where γ is the disturbance attenuation level defined earlier. Then,

(1) if $\|\Delta p_{ij}\| = 0$, the synchronization error asymptotically converges to zero, and

(2) if $\|\Delta p_{ij}\| \neq 0$, the synchronization error satisfies the L_2-gain condition.

Proof: Consider the following non-negative quadratic function

$$V(t) = \delta^T(t)(Q_2 \otimes P_1)\delta(t) \tag{28}$$

where $\delta(t)$ is given by (22), and Q_2 is defined in (26). The derivative of the above function is

$$\dot{V}(t) = \dot{\delta}^T(t)(Q_2 \otimes P_1)\delta(t) + \delta^T(t)(Q_2 \otimes P_1)\dot{\delta}(t) \tag{29}$$

From (25):

$$\begin{aligned}\dot{V}(t) = [\delta^T(t)(I_N \otimes A^T + c\hat{\mathcal{L}}^T \otimes (BK)^T) \\ + F^T(t)(I_N \otimes B^T)](Q_2 \otimes P_1)\delta(t) + \\ \delta^T(t)(Q_2 \otimes P_1)[(I_N \otimes A + c\hat{\mathcal{L}} \otimes BK)\delta(t) + (I_N \otimes B)F(t)]\end{aligned} \tag{30}$$

It follows from (26) that

$$\begin{aligned}\dot{V}(t) = \delta^T(t)[Q_2 \otimes (A^T P_1 + P_1 A) \\ - 2Q_2 R_1^{-1} Q_2 \otimes P_1 B B^T P_1]\delta(t) \\ + F^T(t)(Q_2 \otimes B^T P_1)\delta(t) \\ + \delta^T(t)(Q_2 \otimes P_1 B)F(t)\end{aligned} \quad (31)$$

Adding and subtracting

$$\delta^T(t)\left(Q_2 \otimes \left((\frac{1}{\gamma^2} - 1)P_1 B B^T P_1\right)\right)\delta(t)$$

to (31) yields:

$$\begin{aligned}\dot{V}(t) \leq -\delta^T(t)(Q_2 \otimes Q_1)\delta(t) \\ + \delta^T(t)\left(\left(Q_2 - 2Q_2 R_1^{-1} Q_2\right) \otimes P_1 B B^T P_1\right)\delta(t) \\ - \frac{1}{\gamma^2}\delta(t)^T\left(Q_2 \otimes P_1 B B^T P_1\right)\delta(t) \\ + F^T(t)(Q_2 \otimes B^T P_1)\delta(t) + \delta^T(t)(Q_2 \otimes P_1 B)F(t)\end{aligned} \quad (32)$$

To find conditions under which the second term in the right side of the above inequality is negative, based on Lemma 1, one can write:

$$\left(Q_2 - 2Q_2 R_1^{-1} Q_2\right) = \left(V^{-1}\right)^T \left[c\Sigma - 2c^2\Sigma^2\right] V^{-1} < 0 \quad (33)$$

The above inequality holds if:

$$c\Sigma - 2c^2\Sigma^2 < 0 \quad (34)$$

Since $c > 0$ and $\Sigma > 0$, thus

$$I - 2c\Sigma < 0 \quad (35)$$

This implies that for (33) to hold, it suffices to have:

$$c > \frac{1}{2\min(\mathrm{Re}\{\lambda_i\})} \tag{36}$$

On the other hand, the third term on the right side of (32) is also negative (as it is a quadratic term with a negative sign).

$$\begin{aligned}\dot{V}(t) \leq -\delta^T(t)(Q_2 \otimes Q_1)\delta(t) \\ - \frac{1}{\gamma^2}\delta^T(t)\left(Q_2 \otimes P_1 B B^T P_1\right)\delta(t) \\ + F^T(t)(Q_2 \otimes B^T P_1)\delta(t) + \delta^T(t)(Q_2 \otimes P_1 B)F(t)\end{aligned} \tag{37}$$

And since the second term is negative, then

$$\begin{aligned}\dot{V}(t) \leq -\delta^T(t)(Q_2 \otimes Q_1)\delta(t) + F^T(t)(Q_2 \otimes B^T P_1)\delta(t) \\ + \delta^T(t)(Q_2 \otimes P_1 B)F(t)\end{aligned} \tag{38}$$

In the case of no uncertainties ($\|\Delta p_{ij}\| = 0$), $F(t)$ is zero, and hence, it results from the above inequality that $\dot{V}(t) < 0$, and $\delta(t)$ is asymptotically stable. The proof of part 1 is now complete.

For the case when $F(t) \neq 0$, consider the Hamiltonian as

$$H = \dot{V}(t) + \delta^T(t)(Q_2 \otimes Q_1)\delta(t) - \gamma^2 F^T(t)(Q_2 \otimes I)F(t) \tag{39}$$

From (39) and (32), and on noting that some of the terms on the right side of (32) are shown to be negative so far, one has:

$$\begin{aligned}H = -\frac{1}{\gamma^2}\delta^T(t)\left(Q_2 \otimes P_1 B B^T P_1\right)\delta(t) \\ + F^T(t)(Q_2 \otimes B^T P_1)\delta(t) + \delta^T(t)(Q_2 \otimes P_1 B)F(t) \\ - \gamma^2 F^T(t)(Q_2 \otimes I)F(t)\end{aligned} \tag{40}$$

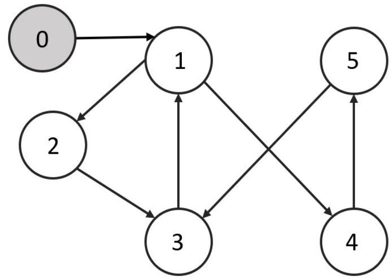


Figure 3.1: The topology of the digraph of the MAS of Example 1. Node 0, shown in gray, represents the leader; other nodes are followers.

Thus,

$$H \leq -\left\| \gamma(Q_2^{\frac{1}{2}} \otimes I_N)(z - F^T(t)) \right\|^2 \tag{41}$$

where $z = \frac{1}{\gamma^2}\delta^T(t)P_1B$. Taking the integral of (39) and using the above inequality yields:

$$V(T) - V(0) + \int_0^T (\delta^T(t)(Q_2 \otimes Q_1)\delta(t) \\ - \gamma^2 F^T(t)(Q_2 \otimes I)F(t))dt \leq 0 \tag{42}$$

Since $V(0) = 0$ and $V(T) > 0$, one has:

$$\int_0^T (\delta^T(t)(Q_2 \otimes Q_1)\delta(t) - \gamma^2 F^T(t)(Q_2 \otimes I)F(t))dt \leq 0 \tag{43}$$

Therefore, the resultant controller satisfies the L_2-gain condition. ■

3.6 Simulations

Example 1. Consider a MAS with the digraph given in Fig. 3.1, where the leader (represented by the gray node) sends its data only to agent 1. It can be observed that the leader is the root node, and Assumptions 1-3 are met. The information of the leader is only sent to node 1.

Assume that each agent has the following state-space matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \tag{44}$$

It is straightforward to verify that the above system is controllable, therefore it is stabilizable, too. Let the probability matrix (corresponding to the communication links) be given by:

$$\mathcal{P} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.8 + 0.1\sin(t/20) & 0 & 0 & 0.3 \\ 0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.7 + 0.2\sin(t/10) & 0 \end{bmatrix} \tag{45}$$

The nominal and time-varying components of the probability matrix can, respectively, be expressed by:

$$\hat{\mathcal{P}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 & 0.3 \\ 0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.7 & 0 \end{bmatrix} \tag{46}$$

and

$$\Delta\mathcal{P}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1\sin(t/20) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2\sin(t/10) & 0 \end{bmatrix}$$

Assume also that the probability of the existence of a link from the leader to agent 1 is equal to 0.5.

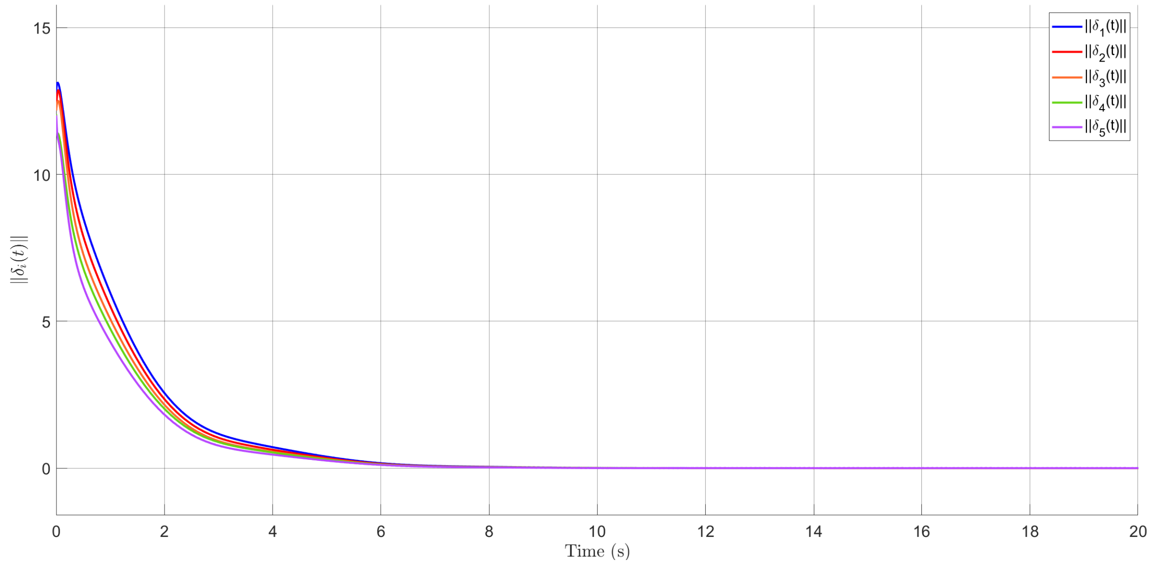


Figure 3.2: The norm of the synchronization error signal when $u_0 = 0$

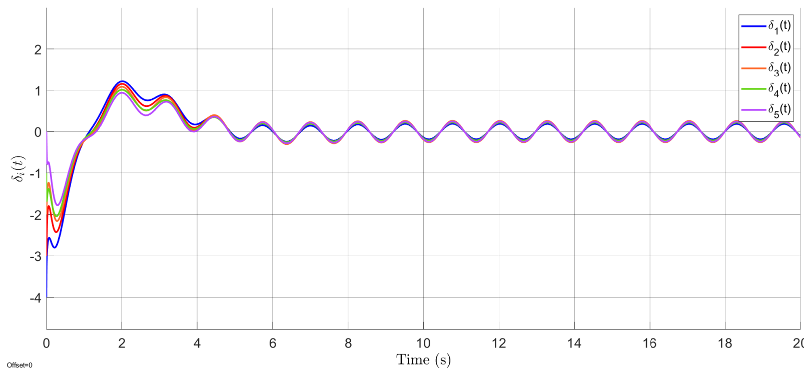
Solving the LMI equation (27) using Matlab CVX [67, 68] results in:

$$P_1 = \begin{bmatrix} 3.2425 & 1.5769 & 2.8715 \\ 1.5769 & 2.0827 & 2.1753 \\ 2.8715 & 2.1753 & 4.6914 \end{bmatrix} \tag{47}$$

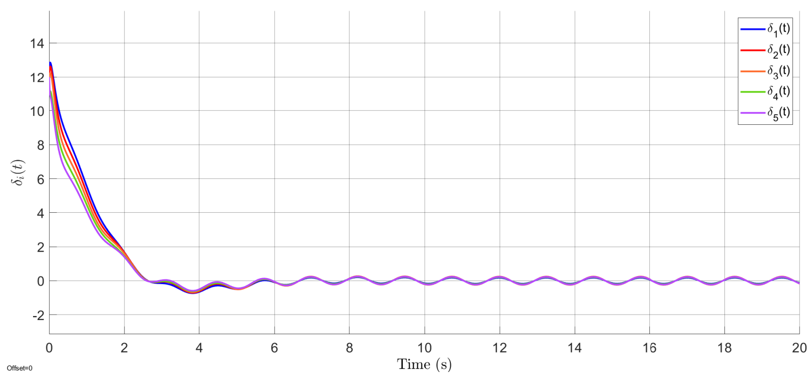
Now, with the above matrix and the parameters $\gamma = 0.5$, $Q_1 = 4I_3$ and $c = 20$, the feedback gain matrix is obtained as

$$K = \begin{bmatrix} -4.8194 & -3.6595 & -5.0468 \end{bmatrix}$$

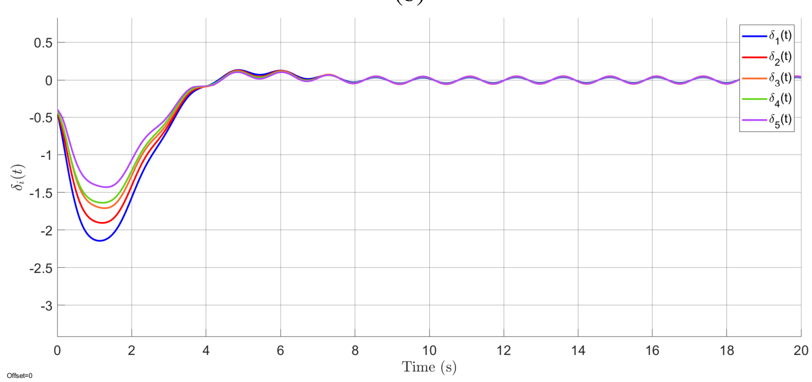
By applying the control signal (20), to this MAS, states of the followers converge to the leader's states, and the norm of the synchronization error $\|x_i - x_0\|$ converges to a small value. This is demonstrated in Fig. 3.2 for $u_0 = 0$. It can be observed from this figure that if the leader does not have any input, the norm of the error signal converges to an arbitrarily small value by increasing Q_1 and c. Fig. 3.3 demonstrates the results for the above MAS when an external input equal to $8\sin(5t)$ is applied to the leader. This figure shows that the synchronization error of each agent, in this case, reaches a small neighborhood of zero, which satisfies the L_2-gain synchronization problem.



(a)



(b)



(c)

Figure 3.3: The synchronization error signal for three states when $u_0 = 8\sin(5t)$

3.7 Conclusions

The consensus problem for an asymmetric leader-follower MAS is investigated in this chapter. The network contains unknown communication links and hence is represented by a random digraph. A robust control protocol is developed to account for network uncertainties of a prespecified limited magnitude. Simulations confirm the effectiveness of the results for various scenarios.

Chapter 4

Conclusions and Future Research Directions

In this thesis, we introduce new Q-learning-based strategies for estimating the probability of the existence of different edges in a random digraph representing a multi-agent system (MAS). These approaches use adaptive learning rates for updating the information. This information is then used to compute the digraph's generalized algebraic connectivity (GAC), which is closely related to the convergence rate of consensus algorithms performed on the network. Finally, as an experimental example, a network with five underwater sensors is considered, and the performance of the proposed estimation algorithms is evaluated. The computed probability matrix is then used to study the consensus problem in a MAS subject to uncertain communication links. A robust control law is designed to reduce the impact of uncertainty on the disagreement error among the leader and followers. Finally, simulations confirm the theoretical results.

4.1 Suggestions for Future Work

The following problems are suggested for future work:

(1) The learning rate of the update rule (α) and window size (w) can be selected by defining a cost function and solving the corresponding optimization problem.

(2) There are different data transformation methods and different environmental conditions. By implementing the probability matrix estimation in a real-world application and developing a proper algorithm for different cases, one can select the most appropriate one at any point in time.

(3) For the cases where the size of uncertainty is relatively large, one can use a multiple-model approach with a proper switching control strategy to achieve consensus. To this end, a family of models is obtained first for the network according to the uncertain parameters. A consensus control law is then designed for each network.

(4) One can minimize the disturbance attenuation level γ in the consensus problem described in Theorem 1 of Chapter 3, which makes it more realistic in practice.

(5) An adaptive control method can also be used to deal with the time-varying nature of the network. Both model-reference adaptive control and model-predictive control approaches can be used to account for the parameter variation.

(6) For a certain class of nonlinear systems, one can use the hashing-based neurofuzzy network (HBNN) introduced in [69] and design a control law for the linearized model accordingly.

Bibliography

[1] A. Ligtenberg, M. Wachowicz, A. K. Bregt, A. Beulens, and D. L. Kettenis. "A design and application of a multi-agent system for simulation of multi-actor spatial planning". In: *Journal of Environmental Management* 72.1-2 (2004), pp. 43–55.

[2] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls. "Reinforcement learning-based multi-agent system for network traffic signal control". In: *Iet Intelligent Transport Systems* 4 (2010), pp. 128–135.

[3] T. Nagata, Y. Ueda, and M. Utatani. "A multi-agent approach to smart grid operations". In: *2012 IEEE International Conference on Power System Technology (POWERCON)* (2012), pp. 1–5.

[4] A. Garrab, A. Bouallegue, and R. Bouallegue. "Multi-agent modeling of a meters network used in smart grid". In: *2014 World Congress on Computer Applications and Information Systems (WCCAIS)* (2014), pp. 1–5.

[5] E. Davis and L. Morgenstern. "A first-order theory of communication and multi-agent plans". In: *Journal of Logic and Computation* 15.5 (2005), pp. 701–749.

[6] T. Nagata and H. Sasaki. "A multi-agent approach to power system restoration". In: *IEEE Transactions on Power Systems* 17.2 (2002), pp. 457–462.

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

[8] J. Lavaei Yanesi. "High-performance robust decentralized control of interconnected systems". MA thesis. Concordia University, 2006.

[9] L. Ding, Q. L. Han, and G. Guo. “Network-based leader-following consensus for distributed multi-agent systems”. In: *Automatica* 49.7 (2013), pp. 2281–2286.

[10] Z. Li, G. Wen, Z. Duan, and W. Ren. “Designing fully distributed consensus protocols for linear multi-agent systems with directed graphs”. In: *IEEE Transactions on Automatic Control* 60.4 (2015), pp. 1152–1157.

[11] H. Zhang, F. L. Lewis, and A. Das. “Optimal design for synchronization of cooperative systems: state feedback, observer and output feedback”. In: *IEEE Transactions on Automatic Control* 56.8 (2011), pp. 1948–1952.

[12] J. Wang, J. Cao, and S. Jiang. “Fault-tolerant pattern formation by multiple robots: a learning approach”. In: *IEEE 36th Symposium on Reliable Distributed Systems (SRDS)* (2017), pp. 268–269.

[13] C. Vickery and S. A. Salehi. “A mean shift-based pattern formation algorithm for robot swarms”. In: *7th International Conference on Automation, Robotics and Applications (ICARA)* (2021), pp. 16–20.

[14] D. H. Nguyen. “Distributed consensus design for a class of uncertain linear multiagent systems under unbalanced randomly switching directed topologies”. In: *Mathematical Problems in Engineering* (2018).

[15] R. Olfati-Saber and R. M. Murray. “Consensus problems in networks of agents with switching topology and time-delays”. In: *IEEE Transactions on Automatic Control* 49.9 (2004), pp. 1520–1533.

[16] W. Ni and D. Cheng. “Leader-following consensus of multi-agent systems under fixed and switching topologies”. In: *Systems & Control Letters* 59.3 (2010), pp. 209–217.

[17] B. Hou, F. Sun, H. Li, and G. Liu. “Consensus of second-order multi-agent systems with time-varying delays and antagonistic interactions”. In: *Tsinghua Science and Technology* 20.2 (2015), pp. 205–211.

[18] Q. Wei, X. Wang, X. Zhong, and N. Wu. “Consensus control of leader-following multi-agent systems in directed topology with heterogeneous disturbances”. In: *IEEE/CAA Journal of Automatica Sinica* 8.2 (2021), pp. 423–431.

[19] M. M. Asadi, A. Ajorlou, A. G. Aghdam, and S. Blouin. “Global network connectivity assessment via local data exchange for underwater acoustic sensor networks”. In: *Proceedings of the 2013 Research in Adaptive and Convergent Systems* (2013), pp. 277–282.

[20] M. Stojanovic. “On the relationship between capacity and distance in an underwater acoustic communication channel”. In: *Proceedings of the 1st ACM International Workshop on Underwater Networks* 11.7 (2006), pp. 41–47.

[21] M. Garavello and B. Piccoli. *Traffic flow on networks (applied Mathematics)*. MO, USA: Springfield, 2006.

[22] M. M. Asadi, M. Khosravi, A. G. Aghdam, and S. Blouin. “Expected convergence rate to consensus in asymmetric networks: analysis and distributed estimation”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50.3 (2020), pp. 972–987.

[23] C. Wang, X. Wang, and H. Ji. “A continuous leader-following consensus control strategy for a class of uncertain multi-agent systems”. In: *IEEE/CAA Journal of Automatica Sinica* 1.2 (2014), pp. 187–192.

[24] P. Xiao, Z. Zhang, and T. Ma. “Impulsive consensus of uncertain multi-agent systems with control gain error”. In: *2017 Chinese Automation Congress (CAC)* (2017), pp. 7757–7780.

[25] Y. Khazaeni and C. G. Cassandras. “Event-driven cooperative receding horizon control for multi-agent systems in uncertain environments”. In: *IEEE Transactions on Control of Network Systems* 5.1 (2018), pp. 409–422.

[26] M. Mesbahi and M. Egerstedt. “Graph theoretic methods in multiagent networks”. In: *Princeton Series in Applied Mathematics* (2010).

[27] D. Buzorgnia. “Fast convergence in consensus control of leader-follower multi-agent systems”. MA thesis. Concordia University, 2019.

[28] M. M. Asadi, S. Blouin, and A. G. Aghdam. “Distributed dynamic average consensus in asymmetric networks”. In: (2018), pp. 864–869.

[29] J. Jiang and Y. Jiang. “Leader-following consensus of linear time-varying multi-agent systems under fixed and switching topologies”. In: *Automatica* 113 (2020), p. 108804.

[30] S. Jafari. “Structural controllability of multi-agent systems subject to partial failure”. MA thesis. Concordia University, 2010.

[31] D. Hummel. “Formation flight as an energy saving-mechanism”. In: *Israel Journal of Zoology* 41.3 (1995), pp. 261–278.

[32] P. Lin and Y. Jia. “Robust H_∞ consensus analysis of a class of second-order multi-agent systems with uncertainty”. In: *IET control theory & applications* 4.3 (2010), pp. 487–498.

[33] T. Xiong, Zh. Pu, J. Yi, and Z. Sui. “Adaptive neural network time-varying formation tracking control for multi-agent systems via minimal learning parameter approach”. In: *2019 International Joint Conference on Neural Networks (IJCNN)* (2019), pp. 1–8.

[34] P. Erdös and A. Rényi. “On random graphs I”. In: *Publicationes Mathematicae Debrecen* 6 (1959), pp. 290–297.

[35] P. Erdös and A. Rényi. “On the evolution of random graphs”. In: *Magyar Tud. Akad. Mat. Kutató Int. Közl.* 5 (1960), pp. 17–61.

[36] P. Erdös and A. Rényi. “On the evolution of random graphs”. In: *Bull. Inst. Internat. Statist.* 38 (1961), pp. 343–347.

[37] P. Erdös and A. Rényi. “On the strength of connectedness of a random graph”. In: *Acta Math. Acad. Sci. Hungar.* 12 (1961), pp. 261–267.

[38] B. Bollobás. *Random graphs*. New York, NY: Springer New York, 1998, pp. 215–252.

[39] B. Bollobás. *Random graphs*. 2nd ed. Cambridge University Press, 2001.

[40] M. M. Asadi, H. Mahboubi, J. Habibi, A. G. Aghdam, and S. Blouin. “Connectivity assessment of random directed graphs with application to underwater sensor networks”. In: *IEEE Transactions on Control Systems Technology* 25.4 (2017).

[41] S. Blouin. “Intermission-based adaptive structure estimation of wireless underwater networks”. In: *10th IEEE International Conference on Networking, Sensing and Control (ICNSC)* (2013), pp. 146–151.

[42] Y. Hatano and M. Mesbahi. “Agreement over random networks”. In: *IEEE Transactions on Automatic Control* 50.11 (2005), pp. 1867–1872.

[43] C. W. Wu. “Synchronization and convergence of linear dynamics in random directed networks”. In: *IEEE Transactions on Automatic Control* 51.7 (2006), pp. 1207–1210.

[44] M. Porfiri and D. J. Stilwell. “Consensus seeking over random weighted directed graphs”. In: *IEEE Transactions on Automatic Control* 52.9 (2007), pp. 1767–1773.

[45] A. Tahbaz-Salehi and A. Jadbabaie. “A necessary and sufficient condition for consensus over random networks”. In: *IEEE Transactions on Automatic Control* 53.3 (2008), pp. 791–795.

[46] S. Kar and J. M. F. Moura. “Distributed consensus algorithms in sensor networks: quantized data and random link failures”. In: *IEEE Transactions on Signal Processing* 58.3 (2010), pp. 1383–1400.

[47] S. Patterson, B. Bamieh, and A. El Abbadi. “Convergence rates of distributed average consensus with stochastic link failures”. In: *IEEE Transactions on Automatic Control* 55.4 (2010), pp. 880–892.

[48] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed. “Bio-inspired decentralized radio access based on swarming mechanisms over adaptive networks”. In: *IEEE Transactions on Signal Processing* 61.12 (2013), pp. 3183–3197.

[49] M. Fiedler. “Algebraic connectivity of graphs”. In: *Czechoslovak Mathematical Journal* 23.2 (1973), pp. 298–305.

[50] P. Di Lorenzo and S. Barbarossa. “Distributed estimation and control of algebraic connectivity over random graphs”. In: *IEEE Transactions on Signal Processing* 62 (2013).

[51] C. W. Wu. “Algebraic connectivity of directed graphs”. In: *Linear and Multilinear Algebra* 53.3 (2005), pp. 203–223.

[52] C. Li and Z. Qu. “Distributed estimation of algebraic connectivity of directed networks”. In: *Systems & Control Letters* 62 (2013), pp. 517–524.

[53] M. M. Asadi, M. Khosravi, A. G. Aghdam, and S. Blouin. “Generalized algebraic connectivity for asymmetric networks”. In: *2016 American Control Conference (ACC)* (2016), pp. 5531–5536.

[54] F. Miroslav. “Algebraic connectivity of graphs”. In: *Czechoslovak Mathematical Journal* (1973), pp. 298–305.

[55] C. Li, Z. Qu, D. Qi, and F. Wang. “Distributed finite-time estimation of the bounds on algebraic connectivity for directed graphs”. In: *Automatica* 107 (2019), pp. 289–295.

[56] X. Ding and T. Jiang. “Spectral distributions of adjacency and laplacian matrices of random graphs”. In: *The Annals of Applied Probability* 20.6 (2010), pp. 2086–2117.

[57] H. Mahboubi, M. M. Asadi, A. G. Aghdam, and S. Blouin. “A computationally efficient connectivity measure for random graphs”. In: *2015 IEEE Global Communications Conference (GLOBECOM)* (2015), pp. 1–6.

[58] S. Blouin. “Exponential filters for passive underwater acoustic detections - A global processing gain perspective”. In: *accepted for publication in IEEE International Conference in Sensor Signal Processing for Defence (SSPD)* (2021).

[59] P. Stone and M. Veloso. “Multiagent systems: a survey from a machine learning perspective”. In: *Autonomous Robots* 8 (2000), pp. 345–383.

[60] A. Ligtenberg, M. Wachowicz, A. K. Bregt, A. Beulens, and D. L. Kettenis. “A design and application of a multi-agent system for simulation of multi-actor spatial planning”. In: *Journal of Environmental Management* 72 (2004), pp. 43–55.

[61] S. Sheikholeslam and C. A. Desoer. “Longitudinal control of a platoon of vehicles”. In: *1990 American Control Conference* (1990), pp. 291–296.

[62] P. Lin, Y. Jia, and L. Li. “Distributed robust H_∞ consensus control in directed networks of agents with time-delay”. In: *Systems & Control Letters* 57.8 (2008), pp. 643–653.

[63] H. Modares, B. Kiumarsi, F. L. Lewis, F. Ferrese, and A. Davoudi. "Resilient and robust synchronization of multi-agent systems under attacks on sensors and actuators". In: *IEEE Transactions on Cybernetics* 50.3 (2020), pp. 1240–1250.

[64] M. Babahaji, S. Blouin, W. Lucia, M. M. Asadi, H. Mahboubi, and A. G. Aghdam. "Random graphs estimation using Q-learning". In: *2021 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)* (2021).

[65] C. Godsil and G. Royle. *Algebraic graph theory*. Springer-Verlag New York, Inc., 2001.

[66] T. Başar and P. Bernhard. H_∞ *optimal control and related minimax design problems*. Boston, MA, USA: Birkhäuser, 1995.

[67] M. Grant and S. Boyd. *CVX: Matlab software for disciplined convex programming, version 2.1*. http://cvxr.com/cvx. Mar. 2014.

[68] M. Grant and S. Boyd. "Graph implementations for non-smooth convex programs". In: *Recent Advances in Learning and Control*. Ed. by V. Blondel, S. Boyd, and H. Kimura. Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, 2008, pp. 95–110.

[69] O. Ajayi, M. Babahaji, and A. Aghdam. "A neurofuzzy function approximator approach to nonlinear systems state-estimation". In: *2020 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)* (2020), pp. 118–123.