

Average Consensus in Matrix-Weight-Balanced Digraphs

by

Chinnappa Yogesh B Allapanda

B.Eng., PES Institute of Technology (Bangalore, India), Autonomous under
Visvesvaraya Technological University, 2014

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Mechanical Engineering

© Chinnappa Yogesh B Allapanda, 2019
University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by
photocopying or other means, without the permission of the author.

Average Consensus in Matrix-Weight-Balanced Digraphs

by

Chinnappa Yogesh B Allapanda

B.Eng., PES Institute of Technology (Bangalore, India), Autonomous under
Visvesvaraya Technological University, 2014

Supervisory Committee

Dr. Daniela Constantinescu, Supervisor
(Department of Mechanical Engineering)

Dr. Keivan Ahmadi, Departmental Member
(Department of Mechanical Engineering)

ABSTRACT

This thesis investigates the average consensus of multi-agent systems with linear dynamics whose interconnections are modelled by balanced digraphs with matrix-weights. The thesis first introduces the notion of balanced digraphs and mirror graphs for matrix weights. Then it proves that the matrix-weight-balanced consensus controller is indeed globally asymptotically stable. The Lyapunov stability analysis exploits the properties of the mirror graph of a balanced digraph. Further, the necessary and sufficient condition for the system to achieve average consensus is shown to be positive definiteness of the matrix weights of its balanced digraph. Simulations with robots in SIMULINK verify that positive definite matrix weights on balanced graphs are indeed necessary and sufficient for average consensus. Finally formation control of a multi-robot system is shown to be an application of the matrix-weight-balanced consensus algorithm using real time simulation of *Clearpath Ridgeback* robots in Gazebo and ROS.

Contents

Supervisory Committee	ii
Abstract	iii
Table of Contents	iv
List of Figures	vi
Acknowledgements	vii
1 Introduction	1
1.1 Motivation	1
1.2 Consensus Algorithm	4
1.3 Average consensus	4
1.4 Weighted Graphs	5
1.5 Matrix-Weighted Graphs	5
1.6 Thesis Outline	6
2 Background Research	7
2.1 Scalar-weighted Consensus	8
2.1.1 Preliminaries	8
2.1.2 Scalar-weighted Consensus Literature	12
2.1.3 Time varying scalar-weights	14
2.1.4 Applications of scalar-weighted-consensus	15
2.1.5 Average Consensus Scalar-weighted Graphs	16
2.2 Matrix-weighted Consensus	18
2.2.1 Preliminaries	18
2.2.2 Matrix-Weighted Consensus Literature	19
2.3 Problem Definition and Contribution	21

3	Necessary and Sufficient Conditions	22
3.1	Relevant Definitions	22
3.2	Overview of the proof	24
3.3	Convergence Analysis	25
3.4	Necessary and Sufficient Condition For Average Consensus	28
4	Simulations	31
4.1	Consensus	32
4.1.1	Balanced Graph	32
4.1.2	Unbalanced Graph	35
4.2	Formation Control	36
5	Conclusion and Future Work	40
5.1	Future Work	41
	Bibliography	42

List of Figures

Figure 1.1 Swarm Of Ground Robots	2
Figure 1.2 Simple two node undirected graph	3
Figure 1.3 Simple two node directed graph	3
Figure 1.4 Consensus algorithm for formation control	4
Figure 2.1 Connected graph	9
Figure 2.2 Strongly Connected graph	10
Figure 2.3 Directed Spanning Tree	11
Figure 2.4 Scalar Weighted Balanced Graph	11
Figure 2.5 Scalar Weighted Unbalanced Graph	11
Figure 3.1 Balanced Matrix-weighted graph	23
Figure 3.2 Unbalanced Matrix-weighted graph	23
Figure 4.1 Communication Topology of Balanced Graph	33
Figure 4.2 Consensus of robots for Balanced Graph	33
Figure 4.3 Trajectory of states in x (Balanced Graph)	34
Figure 4.4 Trajectory of states in y (Balanced Graph)	34
Figure 4.5 Trajectory of states in z (Balanced Graph)	35
Figure 4.6 Communication Topology of Unbalanced Graph	35
Figure 4.7 Consensus of robots for Unbalanced Graph	36
Figure 4.8 Trajectory of states in x (Unbalanced Graph)	36
Figure 4.9 Trajectory of states in y (Unbalanced Graph)	37
Figure 4.10 Trajectory of states in z (Unbalanced Graph)	37
Figure 4.11 Setting Up Gazebo with SIMULINK	38
Figure 4.12 Initial Configuration of Robots	39
Figure 4.13 Final Formation	39

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Daniela Constantinescu for guiding me through the research work. Whenever I had doubts about the tractability of a problem or during the times when research seemed like it was too much to handle, she was always there for support and guidance. Always being available for questions and going beyond what was expected of an advisor has left a deep impression on me. The bi-weekly meetings and the opportunity to bounce off both tough and mediocre ideas will be something I take with me further. It is her philosophy of pushing until "things break", finding out why they do and fixing it, is something that I will forever remember and cherish. I would like to thank her for helping me grow both as a researcher and as an individual during my time here.

And I'd also like to acknowledge the help from everyone in the Assistive Robotics and Mechatronics Lab. Yuan, Lars, Phan, Xizhe and Rebecca for being there to discuss general trends in robotics and other non academic pursuits. It goes without saying that my closest friends Smarak, Robin, Nithin, Prem, Anoop, Shubha, Raghavendra and Manish all have been really helpful in offering guidance and in proofreading parts of my thesis.

I would also like to thank my parents, without whose sacrifice and commitment to a great education, I would not have been able to achieve so much.

Never let your sense of morals prevent you from doing what is right.

Isaac Asimov, Foundation

Chapter 1

Introduction

This chapter explains the motivation and benefits for roboticists to pick up problems in the growing area of multi-robot systems. Then it describes how the complex network of robots are modelled and controlled. A brief explanation of the average consensus algorithm is provided and the motivation for working on the specific case of consensus algorithms for robots whose interconnections are modelled by matrix-weighted digraphs is provided.

1.1 Motivation

When faced with tough engineering problems, science in general has always turned towards nature for solutions. Biomimetic paradigms often inspire and lead to designs that are elegant but difficult to implement.

Robotics(a term coined by Isaac Asimov in 1941) in the early 20th century focused on building robust individual anthropomorphic tools to automate mundane tasks in heavy industries. Since the beginning of the 21st century, scientists have been looking for other possible applications. Instead of a single robot designed to do a particular task, swarms of robots have been envisioned as being the key to building and automating at scales thought of as being impossible before. Quite naturally, they seem to seamlessly draw inspiration from swarms of bees, flocks of birds or schools of fish. The main idea is to find simple local rules for each robot that can in turn lead to emergent, global behaviours. Collective animal behaviours like swarming, formation control and foraging can emerge out of what can be thought of as an alchemy of these local rules. The main advantage of this distributed way of building robot controllers

is that they are robust to communication failure and failure of individual robots. The driving force behind research in this direction is that simple algorithms that work for a handful of robots will scale naturally for a million robots.

In order to control the entire group of robots as a single entity and for the purpose of analyzing and designing control laws, we need to build a good mathematical model of the network of robots. The natural question then is, what mathematical tools can we use to describe this complicated network?

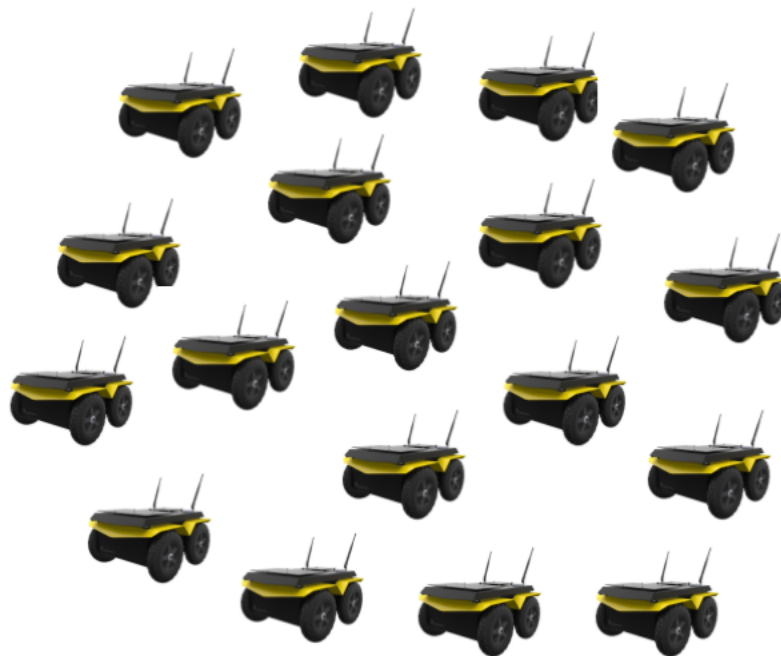


Figure 1.1: Swarm Of Ground Robots

Consider a set of robots (Figure. 1.1). The objective is to control the behaviour of the group of robots as a whole. Each individual robot evolves in time according to its own dynamics. But there is a need to model the collection of robots as a single entity. And that model must include a description of how the robots talk to and respond to each other. It turns out graph theory plays a very important role in this regard. Every robot in the network can be represented by a node and the interactions between them can be represented by edges of a graph. The positions of the nodes in

high dimensional space is used to represent the state(location) of the robots and the topology of the graph represents the communication structure.

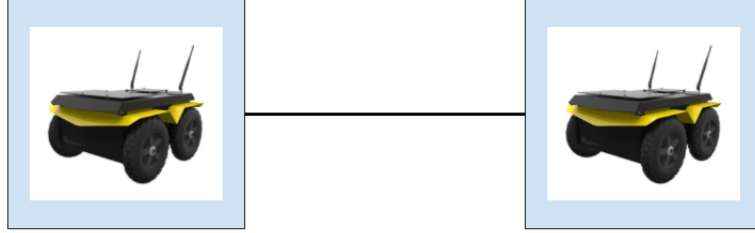


Figure 1.2: Simple two node undirected graph

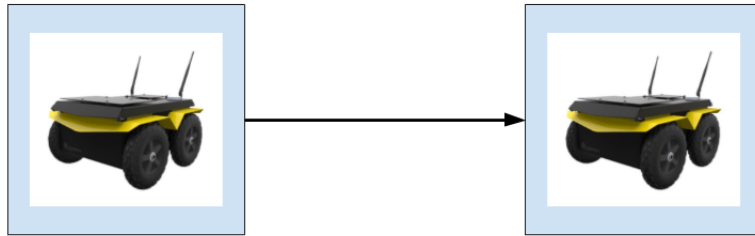


Figure 1.3: Simple two node directed graph

The fundamental building block for such a model is the elementary two node graph. An undirected edge represents the ability of two robots to exchange information with each other mutually (Figure. 1.2). Sometimes the communication between robots can be unidirectional. In this case a directed edge represents the flow of information between the robots. These blocks can then be added up to generate a model of the entire system/swarms of robots. The generated graph is called the multi-robot graph network.

The next question that needs to be answered is how accurately must the dynamics of each robot be modelled? While building a highly accurate nonlinear dynamic model is the only way to guarantee that the robots actually do what the controller commands it to, it has become really challenging to build controllers to handle the non-linearity of large networked robots. This is where hierarchical controllers become increasingly relevant. At the high level, the robots can be modeled as first order agents. So the high level controller generates a trajectory as input to every robot. And the

assumption is that each robot has its own version of a separate low-level controller, that uses a highly accurate non-linear model to guarantee trajectory tracking.

1.2 Consensus Algorithm

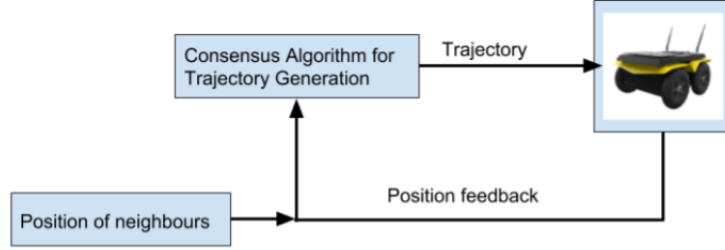


Figure 1.4: Consensus algorithm for formation control

A wide variety of multi-robot behaviours like flocking, formation control and cooperative hunting can be implemented as a specific case of the well known consensus algorithm [40]. So the consensus algorithm can be thought of as a trajectory generation algorithm or a deterministic motion planning algorithm (Figure. 1.4). The general idea behind the consensus algorithm is that robots in a network start out with each robot having a separate state. The objective then is for the robots to communicate with each other so that all of them reach an agreement/consensus on their states. While it is really difficult to explain the dynamics behind this algorithm in simple words, the easiest way to visualize the algorithm is to think of the information propagation in the network as a kind of information diffusion.

1.3 Average consensus

Instead of allowing the robots to reach an unspecified common state, sometimes it becomes important to determine the mutually agreed state beforehand. This then becomes a constrained consensus problem and is commonly referred to as the f – consensus problem [45].

The average consensus problem is defined as the f – consensus problem where it is required that all the robots reach a state which is the simple average of their initial states. This problem is derived from a similar one studied in the distributed sensor fusion community. The problem there is for a large network of sensors measuring

the same physical quantity. The solution is to describe the communication topology and information passed so that all the sensors reach consensus at point which is the average of the individual states.

1.4 Weighted Graphs

The idea of weighted graphs has been borrowed from the logistics/operations community. For a given graph, a set of numbers called weights can be associated with each edge. The traversal of the graph from one node to another leads to an accumulation of weights along the path. This model finds a lot of uses in the path planning/optimization algorithms of large logistics companies.

From a robotics perspective, the edge weights in a consensus algorithm perform the same role as gains in a controller. They can be tuned to regulate the speed of response of the individual robots.

1.5 Matrix-Weighted Graphs

Recently, the abstract idea of assigning matrix-weights for each edge has received a lot of attention. While this might seem difficult to comprehend from a robotics perspective, a good explanation for this case exists in the modelling of opinion/belief dynamics in social networks [17]. If every node represents a person and edges represent friendships in a social network graph, the set of beliefs a person holds can be the state of the node in a high dimensional space. Using matrix weights, the interdependence of beliefs among friends can be modelled. Then the conditions of consensus dictate what the social network graph must look like so that friends reach an agreement on their beliefs.

It is shown later on in this thesis, how the same consensus law can be extended to the multi-robot formation control problem.

The biggest advantage of matrix-weighted graphs from a robotics perspective is the ability to generate cluster consensus, i.e have groups of robots arrive at different consensus states. With the use of diagonal matrix-weights, selective consensus can be achieved in a subset of the state of every robot in a single unified controller. Doing the same using scalar weights would seem a little challenging.

An example for the application of cluster consensus behaviour would be the formation control of quadrotors UAV's. The unified matrix-weighted controller could allow

maintaining formation in the x and y direction(in a plane), but grant the quadrotors flexibility in choosing the attitude (z-direction). In general, this *direction of disagreement of states* can be chosen to be any arbitrary direction in space (instead of just the z-direction).

1.6 Thesis Outline

Chapter 1 A brief introduction to the topic of consensus and its relevance to the problems in robotics.

Chapter 2 A taxonomy of existing work in the area of scalar-weighted and matrix-weighted consensus algorithms.

Chapter 3 The main convergence analysis presented in the thesis. A detailed description of the proof of convergence and stability of the averaging controller.

Chapter 4 Simulations performed to show the conditions for consensus and the applications of the controller to formation control in autonomous ground robots.

Chapter 5 Conclusion and Future work.

Chapter 2

Background Research

Consensus algorithms have been well researched and its applications widely known in the robotics community for the past twenty or so years. An overwhelming majority of the work in the area considers multi-robot systems whose interactions/communications are modelled by scalar-weighted graphs. But off late, a special class of consensus algorithms have garnered interest from the multi-agent systems community. This is called the consensus algorithm for agents with the communications modelled by matrix-weighted graphs. For the remainder of the thesis, the two classes of problems are referred to as the scalar-weighted consensus (SWC) and the matrix-weighted-consensus (MWC) respectively. The main objective of this chapter is two fold. On the one hand it serves as a place to fall back on, any time the reader has doubts about the general terminology related to both the areas. On the other hand, it helps compare and contrast the main ideas between the two.

The first section discusses the research related to the scalar-weighted-consensus. A general introduction to all the terminology and the definitions that are relevant to this class of consensus algorithm is provided. It is seen that some of the definitions based on graph theory are common for both the scalar and the matrix-weighted consensus problems. Then an in depth literature review of some of the most important work is presented. The aim of this exercise is to help the reader build a map of all the existing works. This can then be used as a guide to choosing problems that can be identified and solved for the matrix-weighted-consensus case as future work.

The second section starts with defining key concepts and notations for the matrix-weighted-consensus algorithm. This brief overview contains just enough information to help understand the existing literature in the area. Any additional definitions and concepts generated as a result of research in the lab is presented in the next chapter

and they are introduced as and when the concept becomes relevant.

In the final section, the main motivation for choosing this problem and a statement of contributions is presented.

Note To avoid ambiguity in notations where the two areas share equations, the scalar-weighted representations use "s" as the subscript.

2.1 Scalar-weighted Consensus

2.1.1 Preliminaries

Let the fixed and directed scalar-weighted graph representing the multi-robot system be denoted by \mathcal{G}_s . The graph \mathcal{G}_s can also be represented as $\{(\mathcal{V}_s, \mathcal{E}_s, \mathcal{A}_s)\}$, where $\mathcal{V}_s = \{1, 2, \dots, n\}$ is the set of nodes, $\mathcal{E}_s = \{e_{ij} = (i, j) | i, j \in \mathcal{V}_s, i \neq j\}$ denotes the set of edges, and $\mathcal{A}_s = \{a_{ij} \in \mathbb{R}^d | (i, j) \in \mathcal{E}_s, a_{ij} > 0\}$ denotes the set of scalar-weights.

Let \mathbf{x}_i represent the position of robot i in cartesian space, where $\mathbf{x}_i \in \mathbb{R}^d$. The existence of a directed edge $(i, j) \in \mathcal{E}_s$ means robot i receives position information from robot j . Also robot j is called the parent node and robot i is called the child node. Hence when $a_{ij} > 0$, it implies $(i, j) \in \mathcal{E}_s$. When $(i, j) \notin \mathcal{E}_s$, $a_{ij} = 0$. There are no self loops, i.e. $a_{ii} = 0$.

The adjacency matrix of \mathcal{G}_s is $\mathbf{A} = [a_{ij}]_{i,j=1,\dots,n}$. Let $\mathcal{N}_{i_s} = \{j \in \mathcal{V}_s | e_{ij} \in \mathcal{E}_s\}$ represent the set of neighbours of i .

The out-degree of robot i , is defined as the sum of scalar-weights of the edges directed away from agent i :

$$D_{i_s}^{out} = \sum_{j=1}^n a_{ij} \quad (2.1)$$

Then in-degree of robot i , is defined as the sum of the scalar-weights of edges directed towards agent i :

$$D_{i_s}^{in} = \sum_{j=1}^n a_{ji} \quad (2.2)$$

The Degree matrix for graph \mathcal{G}_s is:

$$\Delta_s = \text{diag}(D_{i_s}^{out}) \quad (2.3)$$

The scalar-weighted Laplacian is:

$$\mathbf{L}_s = \Delta_s - \mathbf{A}_s \quad (2.4)$$

The consensus law is:

$$\dot{\mathbf{x}}_i = \sum_{v_j \in \mathcal{N}_{i_s}} a_{ij}(\mathbf{x}_j - \mathbf{x}_i) \quad \forall \quad i = 1, \dots, n \quad (2.5)$$

The overall system dynamics can be represented by

$$\dot{\mathbf{x}} = -\mathbf{L}_s \mathbf{x} \quad (2.6)$$

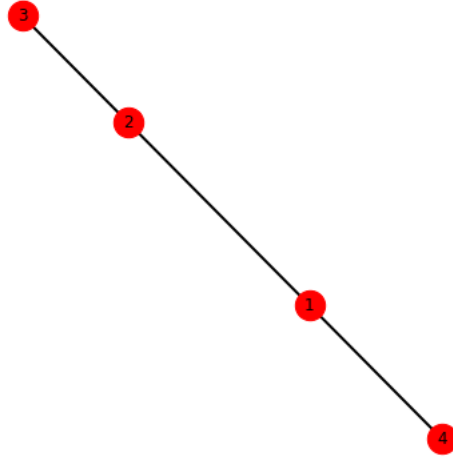


Figure 2.1: Connected graph

Definition 1. Consensus

The system of robots is said to achieve consensus if and only if $\mathbf{x}_i = \mathbf{x}_j$ for each $(i, j) \in \mathcal{V}$ [50]. The consensus space of the multi-robot system is defined as $\mathcal{R} = \text{Range}\{\mathbf{1}_n \otimes \mathbf{I}_d\}$.

Definition 2. Connected Graph

An undirected graph is said to be connected if, for every pair of vertices in \mathcal{V}_s , there is a path that has them as its end vertices [31] (Figure. 2.1).

Definition 3. Weighted Graph

If, together with the edge and vertex sets, a function $w : E \rightarrow R$ is given that associates a value to each edge, the resulting graph is a weighted graph [31].

Definition 4. Directed Graph/ Di-graph

When the edges in a graph are given directions, the resulting graph is a directed graph or di-graph [31].

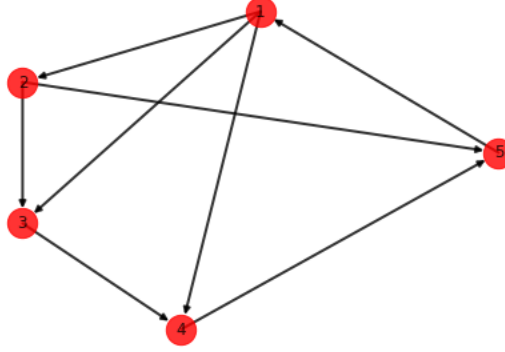


Figure 2.2: Strongly Connected graph

Definition 5. Strongly Connected Graph

A directed graph is said to be strongly connected if there exists a directed path between any two nodes in the graph [31] (Figure. 2.2).

Definition 6. Tree

A tree is a graph (containing at least one vertex) in which any two vertices are connected by exactly one path. [50]

Definition 7. Directed Spanning Tree

A directed tree is called a spanning tree if it contains all the nodes in the graph (Figure. 2.3).

Definition 8. Balanced Graph Scalar-Weighted

A node in a scalar-weighted digraph is said to be balanced if its in-degree is the same as its out-degree. A graph is then said to be balanced, if all the nodes of the graph are balanced (Figure. 2.4), (Figure. 2.5) i.e,[45]

$$\sum_{j \neq i} a_{ij} = \sum_{j \neq i} a_{ji} \quad \forall \quad i \in \mathcal{V}_s \quad (2.7)$$

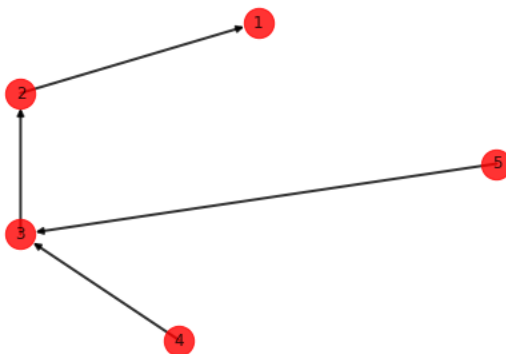


Figure 2.3: Directed Spanning Tree

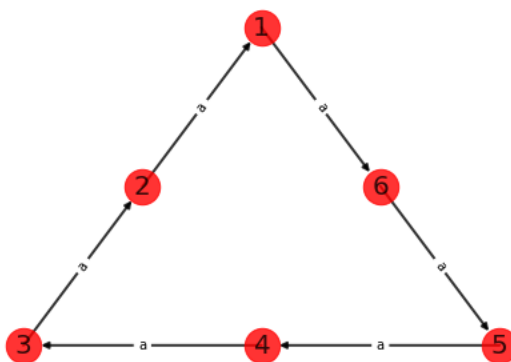


Figure 2.4: Scalar Weighted Balanced Graph

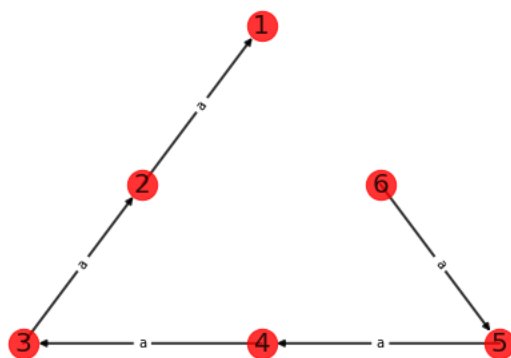


Figure 2.5: Scalar Weighted Unbalanced Graph

Laplacian

The graph Laplacian plays a fundamental role in the convergence analysis of the controller for networked robots modelled as scalar-weighted graphs. From the definition of the graph Laplacian (2.4), every row sums to zero ($\sum_j l_{ij} = 0$). Hence \mathbf{L}_s always has an eigenvalue 0 corresponding to the eigenvector $\mathbf{1}$.

The Laplacian for the directed graph is asymmetric. So proving the stability depends on estimating the eigenvalues of a square asymmetric $n \times n$ matrix. The only solution to this problem is in using the *Gershgorin's theorem* on the Laplacian.

2.1.2 Scalar-weighted Consensus Literature

The extent of work in the scalar-weighted consensus area differ from each other based on the constraints imposed on the controller, the constraints on the actuator, the dynamics of the agent considered, the weights along each edge, the nature of the communication network and the problems related to the communication infrastructure. These are the main criteria that are used to classify the large research being done. Also in this section, some of the applications of the scalar-weighted consensus algorithms are reviewed.

A framework for dealing with consensus problems in networked robotic systems or multi-robot systems was developed in [44]. This was based on an earlier work on the problem [15]. For the undirected communication topology, the condition for consensus is that the graph \mathcal{G}_s is connected [44]. The necessary condition for consensus in directed graphs was initially believed to be the existence of a strongly connected graph [46]. Later the less conservative condition of a spanning tree was proved to be the necessary condition to achieve consensus [42]. The necessary and sufficient condition for *average consensus* in scalar-weighted graphs is the existence of a strongly connected weight-balanced graph [46].

Delayed Networks

A lot of the work in consensus deals with the assumption that agents receive information from its neighbours without delay. This assumption is not true and hence has resulted in research trying to identify the conditions for which consensus can be achieved in the presence of communication delay or input delay. The main focus in this area is also to determine if consensus can be ultimately reached. The maximum

allowed time delay in the network in a continuous time controller is dependent on the out-degree of the graph [46]. In the discrete time case, the necessary and sufficient condition for multi-agent systems with directed graphs, fixed time delays and fixed graphs is the existence of spanning trees[55]. For discrete time systems with both communication and input delay, only the input delay is shown to affect ultimately achieving consensus [48].

Stochastic Networks

Instead of viewing the interaction graph as being fixed and deterministic, some research has focused on its evolution based on random distributions. This becomes a very important tool to deal with communication failures and packet dropouts. In this class of research a new problem of *almost surely* reaching consensus and the conditions for reaching them are studied [23] [63]. The necessary and sufficient conditions for almost surely reaching consensus is dependent on the spectrum of the average weight matrix [47].

Complex Networks

The dynamics of the robots also play an important role in the consensus law design. Initially most research focused on robots with general linear dynamics (first order, second order). Then the focus has also shifted to robots with non-holonomic motion constraints [12] and towards robots with Euler-Lagrange dynamics [41]. While the consensus law is based on difference of states of the neighbours, the nonlinear closed loop dynamics necessitates the existence of additional terms.

Sampled Data Consensus

Sampled data consensus considers the challenges of dealing with robots in continuous time while the communication and controller updates happen in discrete time. The necessary and sufficient conditions for both fixed and switching interaction graphs for the sampled data with and without sampled delay, for switching and fixed topologies has been well researched. [6],[7],[57],[58].

Asynchronous Updates

The consensus problem is also studied for the case when the robots do not update their states synchronously. This is the consensus problem with asynchronous state

updates. The necessary and sufficient condition for the asynchronous consensus in linear systems without time delays is the existence of a spanning tree in the union of the graphs in a given time interval. [4][14].

Finite Time Convergence

Most linear consensus algorithms guarantee exponential or asymptotic convergence. Another class of problems is finite time convergence in which the algorithm specifies the consensus time. Conditions for finite time convergence in single and double integrator systems has been presented in [10],[25],[5].

All of the sub problems listed in this section can be considered as possible future areas of research for the matrix-weighted-consensus case.

2.1.3 Time varying scalar-weights

The area of scalar-weighted-consensus that considers time varying weights is the most versatile in the field. This is because a large class of varying robot behaviour can be constructed by carefully choosing the scalar weights. Amongst all of them, the connectivity preservation problem is the most important application of time-varying scalar-weights and receives a lot of attention.

The most important topological characteristic for achieving global goals in multi-robot systems is maintaining the connectivity of the communication network. For scalar-weighted undirected graphs, connectivity of the graph can be measured algebraically by checking if the second smallest eigenvalue λ_2 of the graph laplacian (also known as the Fiedler eigenvalue), is positive. Controllers can hence be designed to maintain the connectivity of the graph. Among the vast research in the area, there are two main classes of problems,

Local Connectivity Maintenance In this class of controllers, the objective is to keep a static communication graph, i.e make sure the robots move in such a way that all the initial edges exist for all time. This is achieved by carefully designing the time-varying scalar-weights [24].

Global Connectivity Maintenance In the second class of controllers, Artificial potential fields and a centralized controller is designed to ensure that the Fiedler eigenvalue remains positive for all time [62].

Within a unified framework of connectivity maintenance several multi-robot objectives like obstacle avoidance, inter-agent collision avoidance, formation control and

maintaining sensing range can be achieved by carefully choosing time varying scalar-weights and corresponding artificial potential fields [22].

Time varying scalar-weights can be exploited to add additional constraints / robot behaviour in the system and is a very important research area in the multi-robot systems community.

2.1.4 Applications of scalar-weighted-consensus

Power Networks

Irrespective of how carefully power networks are designed, cases of drastic load changes, variations in power supply due to nature of source (time of day changes the solar energy supply and rain changes the hydro power generation rate) all lead to a difference in the demand and supply of the power network. A large number of centralized load shedding algorithms have been designed to cutoff loads with lower priority to maintain stability of the network. It has been shown that distributed consensus algorithms can be used for information passing and the resulting algorithms can be used in distributed load shedding[59].

Another important area in power networks is the economic dispatch problem. It is defined as the problem of finding a power output combination of all generators which gives the lowest operating cost while maintaining system constraints. This can be solved by a variant of the consensus algorithm that tries to drive the incremental cost to an optimal value [60].

Sensor Networks

A distributed algorithm for Kalman filtering uses a variant of the average consensus problem. The approximate Distributed Kalman Filtering problem [37], the Distributed Kalman filtering problem [36] and the linear least squares estimator problem [56] depend on using the average consensus algorithm to solve the distributed averaging problems. Sensor networks and data fusion is used in environmental health monitoring, military surveillance and most commonly in state estimation for autonomous vehicles.

Formation Control

Distributed formation control in multi-robot systems can be posed as consensus based algorithms using relative positions of neighbouring robots. This is done by carefully choosing the information states on which consensus is reached. Formation control using consensus algorithms can also be achieved on multi-agent systems modelled on directed graphs [40].

2.1.5 Average Consensus Scalar-weighted Graphs

Average consensus in interactions modelled by scalar-weighted graphs has been studied extensively and several variants of the problem have been discussed. This section aims to provide a clear taxonomy of research in this sub problem.

Static vs Dynamic Reference Average Consensus

If the reference signal for each agent is a constant static value, then the consensus the system achieves is called a static average consensus [46, 42]. The robots then converge to the average of the initial reference states of the network.

In contrast some works deal with consensus algorithms in the case that each robot has a local time varying reference. This problem is called the dynamic average consensus. The objective in this class of problems is not to converge to an exact average of initial states, but to converge to the neighbourhood of the time varying average of the state of robots in the network [38, 16, 9, 2, 8, 32].

Fixed Topology vs Switching Topology

Average consensus algorithms have been studied under the type of communication topology. Some of the works consider that the communication topology is fixed [3][18][26]. While others focus on switching topology. The conditions on the system in both the fixed and switching topology are common to a large extent. Only in the case of switching topology, there is an additional constraint on how fast the switching is allowed [34][54].

Synchronous vs Asynchronous

A lot of work on average consensus deals with agents that have continuous access to the information from their neighbours. This is when the network executions are all

synchronous [44],[42]. In [46], a continuous time algorithm is proposed for average consensus in weight-balanced digraphs. Although this is a good assumption to make, synchronous updates entails a high cost in terms of computation and communication.

Some works have addressed the problem when the information from neighbours and the communication is not continuous but rather happens only during isolated event times. The objective then is to find when the updates should happen so that the corresponding event triggered strategy guarantees average consensus. The analysis in this category of asynchronous controllers also rests on proving that an infinite number of events are not triggered in a finite time interval (called the Zeno Behaviour).

In [11] proposed a strategy for agents to update their states from triggered events while assuming the information from neighbours is continuous. In [33], a completely asynchronous event triggered controller is suggested for the undirected graph case and in [34][29], the concept is extended to weight-balanced graphs.

Continuous vs Discrete Time

Another key classification of the literature is along the lines of the controller being designed in continuous time [28],[43], [46], [45] or in discrete time [3],[18]. While the consensus law remains the same in both cases, the only difference will be in the tools used in the stability analysis.

Delayed Networks

Most average consensus algorithms considered that each agent can obtain the states of its neighbours without delay. For the average scalar-weighted-consensus case both communication and input delay have been considered in existing literature. [32],[61].

Privacy preserving problem

In most average consensus algorithms, the agents readily share their states/information with their neighbours. This is going on the assumption that all agents in the network are cooperative and are contributing towards achieving the overall task. In recent times, existence of non cooperative and adversarial agents too have been considered. This gives rise to the problem of making sure the local reference information shared between agents is not breached by adversarial agents. The problem is then called the privacy preserving problem.

This problem is solved by encoding the information to be sent to the neighbours using a pairwise unique signal. This additional signal can take the form of a random number [27], a zero mean noise [35] or using continuous-time integrable additive perturbation signals [43].

Digraph balancing problem

Since weight-balanced graphs are necessary for average consensus in scalar-weighted-consensus algorithms, some research has gone into the digraph balancing problem. In this class of problems, given a digraph with a certain topology, the conditions in which it can be balanced and algorithms to take this digraph as input and generate the balanced weights have also been studied. [19][20][21][13].

2.2 Matrix-weighted Consensus

2.2.1 Preliminaries

Consider a multi-robot system with n agents. Let the fixed and directed matrix-weighted graph representing the networked system be denoted by \mathcal{G} . The graph \mathcal{G} can also be represented as $(\mathcal{V}, \mathcal{E}, \mathcal{A})$. Where $\mathcal{V} = \{1, 2, \dots, n\}$ is the set of nodes, $\mathcal{E} = \{e_{ij} = (i, j) | i, j \in \mathcal{V}, i \neq j\}$ denotes the set of directed edges. Then, $\mathcal{A} = \{\mathbf{A}_{ij} \in \mathbb{R}^{d \times d} | (i, j) \in \mathcal{E}, \mathbf{A}_{ij} > 0\}$ denote the set of matrix-weights.

Let \mathbf{x}_i be the position of robot i in cartesian space. Here $\mathbf{x}_i \in \mathbb{R}^d$. The existence of a directed edge $(i, j) \in \mathcal{E}$ means robot i receives information from robot j . Hence j is the parent node and i is the child node. If $(i, j) \in \mathcal{E}$ then $\mathbf{A}_{ij} > 0$. When $(i, j) \notin \mathcal{E}$, $\mathbf{A}_{ij} = \mathbf{0}$. There are no self loops, i.e. $\mathbf{A}_{ii} = 0$. Define $\mathcal{R} = \text{Range}\{\mathbf{1}_n \otimes \mathbf{I}_d\}$.

The adjacency matrix of \mathcal{G} is $\mathbf{A} = [\mathbf{A}_{ij}]_{i,j=1,\dots,n}$. Let $\mathcal{N}_i = \{j \in \mathcal{V} | e_{ij} \in \mathcal{E}\}$ represent the set of neighbours of i . The out-degree of agent i , is defined as the sum of weights of the edges directed away from agent i :

$$\mathbf{D}_i^{\text{out}} = \sum_{j=1}^n \mathbf{A}_{ij} \quad (2.8)$$

Then in-degree of agent i , is defined as the sum of the weights of edges directed towards agent i :

$$\mathbf{D}_i^{in} = \sum_{j=1}^n \mathbf{A}_{ji} \quad (2.9)$$

The Degree matrix for graph \mathcal{G} is:

$$\Delta = blkdiag(\mathbf{D}_i^{out}) \quad (2.10)$$

The matrix weighted Laplacian is:

$$\mathbf{L} = \Delta - \mathbf{A} \quad (2.11)$$

The elements of the laplacian are blocks of matrices, and it can be expressed as:

$$\mathbf{L} = [\mathbf{L}_{ij}]_{i,j=1,\dots,n} \quad (2.12)$$

By definition, for every matrix-weighted graph

$$\mathbf{L}_{ii} = \sum_{j=1, j \neq i} \mathbf{A}_{ij} \quad \forall i \in (1, \dots, n) \quad (2.13)$$

$$\mathbf{L}_{ij} = -\mathbf{A}_{ij} \quad (2.14)$$

The consensus law is

$$\dot{\mathbf{x}}_i = \sum_{v_j \in \mathcal{N}_i} \mathbf{A}_{ij}(\mathbf{x}_j - \mathbf{x}_i) \quad \forall i = 1, \dots, n \quad (2.15)$$

The overall system dynamics can be represented by

$$\dot{\mathbf{x}} = -\mathbf{L}\mathbf{x} \quad (2.16)$$

where

$$\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^\top \quad (2.17)$$

2.2.2 Matrix-Weighted Consensus Literature

In this section, existing works that address multi-agent systems with interactions modelled by matrix-weighted graphs are listed. This will help the reader get a good

idea of the extent of research in the area and give an understanding about the quality of the work presented in the thesis. Because of their applications in both social networks and robotic systems, multi-agent systems with interconnections modelled by matrix weights are another area of recent research interest. The matrix weights are all square and of the same dimension.

A combined strategy for global orientation estimation and formation control can be framed as a consensus problem with rotation matrix weights [30]. Necessary and sufficient conditions for distributed localizability using matrix weighted graphs have been studied [65]. The problem of formation stabilization can also be solved using consensus with rotation matrix-weights [64].

From a social network perspective, evolution of inter-dependent opinions among groups of people is an interesting problem to look at. Matrix-weighted digraphs are used to model such complex interconnections [17],[39].

Positive semi-definite matrix weights can also describe inter-agent connections in oscillator synchronization problems over undirected graphs [53],[51]. Given an oscillator system, the main objective is to find the matrix-weights between each pair of oscillators such that synchronization is guaranteed. It has been proved that matrix-weighted observability is equivalent to the connectivity of the undirected matrix-weighted graph [52].

A general formulation of the consensus problem for multi-agent systems with interconnections modelled by undirected graphs with positive semi-definite matrix weights identifies the existence of a positive spanning tree as the necessary and sufficient condition for consensus [50]. Unlike the scalar-weighted case, a connected graph doesn't necessarily lead to consensus in matrix-weighted case due to the presence of semi-definite weights. An in depth analysis of the properties of matrix-weighted Laplacians can be found in [1].

The extension of the matrix-weighted consensus to the class of multi-agent systems with directed inter-agent communications with the sufficient conditions for leader-follower consensus is presented in [49]. The class deals with leader-follower systems with a static leader, sum of the matrix weights leaving every agent being positive definite and interconnections modelled by directed acyclic graphs. Provable convergence of the followers to the static leader is obtained by splitting the graph into two sub-graphs, the first containing all follower agents and the second containing leader-followers interactions. In contrast, this thesis considers systems with interconnections modelled by a matrix-weight-balanced digraph.

2.3 Problem Definition and Contribution

The main contribution of this thesis is the convergence analysis of the consensus algorithm for a multi-robot system with interconnections modelled as matrix-weight-balanced digraphs. The tool commonly used for convergence in the scalar-weighted literature on digraphs is the Gershgorins theorem on the graph Laplacian. The presence of block diagonal elements in the matrix-weights renders it unfeasible in our case. The analysis exploits a specific property of the mirror graph of balanced digraphs and Lyapunov analysis to show asymptotic convergence. This thesis provides the necessary and sufficient condition to achieve average consensus in multi-robot systems modelled as matrix-weighted digraphs and this has not been done before. This is the first step towards generalizing the results for any given digraph. Further, an application of the above algorithm to formation control for a swarm of autonomous ground robots is shown in Gazebo and ROS.

Chapter 3

Necessary and Sufficient Conditions

This chapter lists out definitions that have been used to simplify the analysis of matrix-weighted consensus algorithms. Then the convergence analysis and proof of stability is shown. Finally, the necessary and sufficient conditions for average consensus in matrix-weight-balanced digraphs is presented.

3.1 Relevant Definitions

Definition 9. Balanced digraph for matrix-weights

A node is balanced, if the sum of weights entering the node is the same as the sum of weights leaving the node. A graph is balanced if and only if every node in the graph is balanced. And to express it algebraically,

$$\mathbf{D}_i^{out} = \mathbf{D}_i^{in} \quad \forall \quad i \in \mathcal{V} \quad (3.1)$$

Definition 9 leads to the following property for balanced digraphs with matrix weights:

$$\mathbf{L}_{ii} = - \sum_{j=1, j \neq i}^n \mathbf{L}_{ji} \quad \forall i \in \{1, \dots, n\} \quad (3.2)$$

Definition 10. Mirror Graphs

Given a matrix-weighted digraph, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$. The reverse edges ($\tilde{\mathcal{E}}$) of \mathcal{G} is obtained by reversing the order of the nodes in \mathcal{G} .

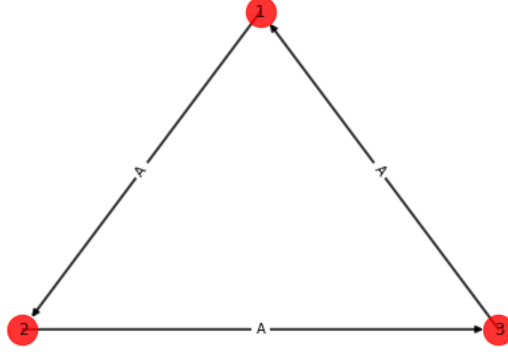


Figure 3.1: Balanced Matrix-weighted graph

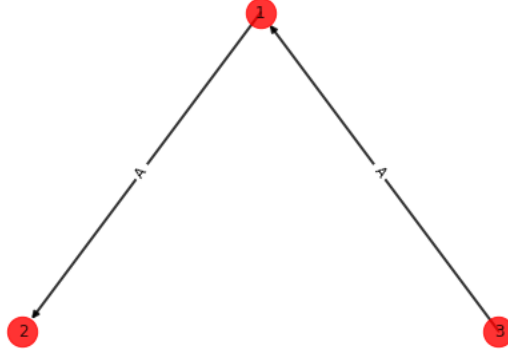


Figure 3.2: Unbalanced Matrix-weighted graph

The mirror graph of \mathcal{G} is $\hat{\mathcal{G}} = (\mathcal{V}, \hat{\mathcal{E}}, \hat{\mathcal{A}})$, containing the same nodes as in \mathcal{G} , the edges $\hat{\mathcal{E}} = \mathcal{E} \cup \tilde{\mathcal{E}}$.

The adjacency matrix of the mirror graph, $\hat{\mathbf{A}}$ is a symmetric matrix where

$$\begin{aligned} \hat{\mathbf{A}} &= [\hat{\mathbf{A}}_{ij}] \\ \hat{\mathbf{A}}_{ij} &= \hat{\mathbf{A}}_{ji} = \mathbf{A}_{ij} + \mathbf{A}_{ji} \end{aligned} \tag{3.3}$$

As long as the matrix weights \mathbf{A}_{ij} of graph \mathcal{G} are positive definite, the matrix weights $\hat{\mathbf{A}}_{ij}$ of the mirror graph $\hat{\mathcal{G}}$ are positive definite as well.

Definition 11. Block Sum

Because the weights of the edges are matrices, a concept of blocks and calculating sums of the blocks has to be defined. Instead of looking at the laplacian \mathbf{L} and the

adjacency \mathbf{A} of \mathcal{G} as having $nd \times nd$ scalar elements in \mathbb{R} , we can view them as matrices having $n \times n$ matrix elements (or blocks) in $M^{d \times d}$. This means the \mathbf{L} and \mathbf{A} both have n block rows and n block columns.

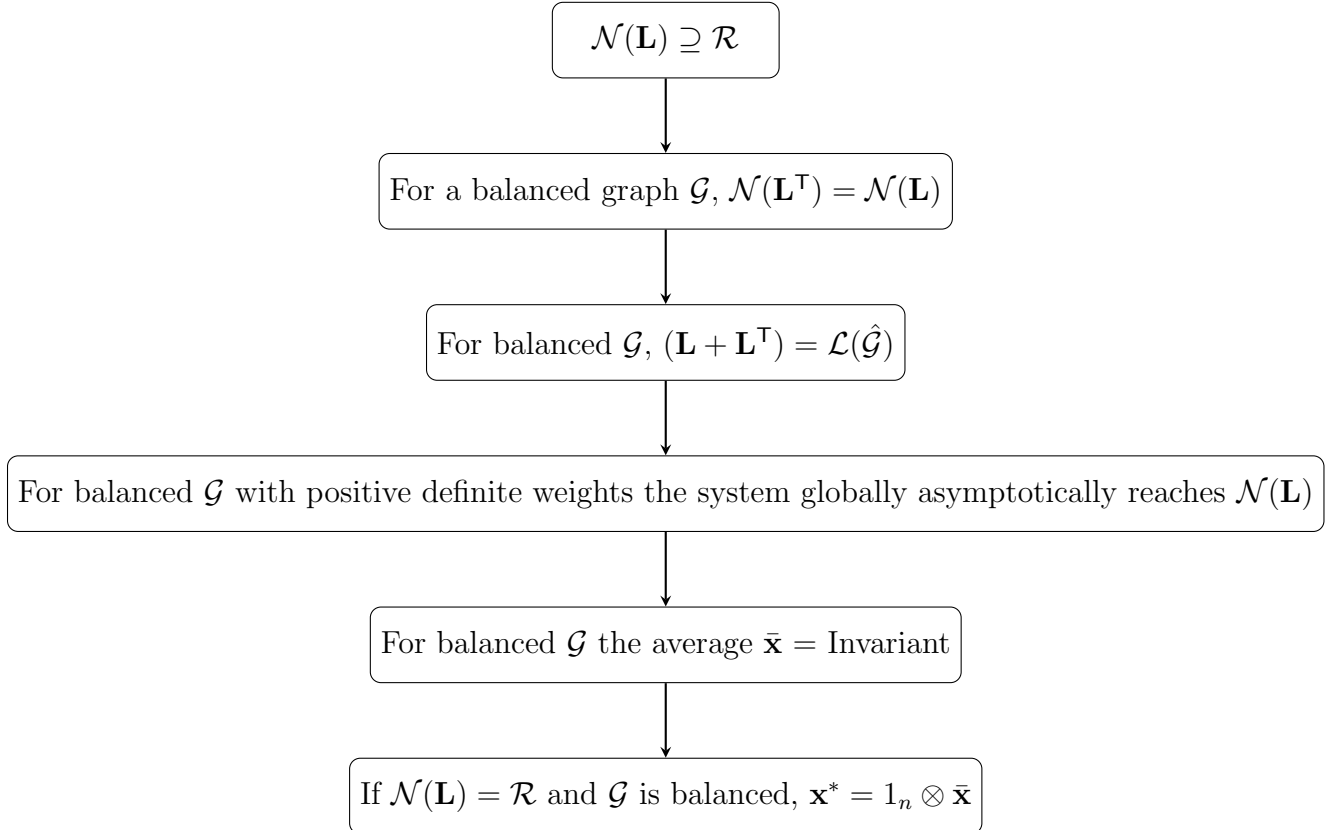
The i^{th} block row sum for the Laplacian \mathbf{L} is:

$$\sum_{j=1}^n \mathbf{L}_{ij} \quad (3.4)$$

Similarly, the i^{th} block column sum of \mathbf{L} is:

$$\sum_{j=1}^n \mathbf{L}_{ji} \quad (3.5)$$

3.2 Overview of the proof



3.3 Convergence Analysis

This section will show that the balanced single-integrator network (2.16) with positive definite matrix gains \mathbf{A}_{ij} is stable. The properties of the Laplacian of a matrix-weight-balanced digraph will help the Lyapunov analysis prove the convergence of the consensus algorithm.

Lemma 3.1. *The asymmetric matrix weighted laplacian \mathbf{L} has nullspace $\mathcal{N}(\mathbf{L}) = \text{span}\{\mathcal{R}, \{\mathbf{v} = [\mathbf{v}_1^\top, \dots, \mathbf{v}_n^\top]^\top \in \mathbb{R}^{n \times d} | (\mathbf{v}_j - \mathbf{v}_i) \in \mathcal{N}(\mathbf{A}_{ij}) \forall (i, j) \in \mathcal{E}\}, \text{ where } \mathcal{R} = \text{Range}\{\mathbf{1}_n \otimes \mathbf{I}_d\}\}.$*

Proof. Consider a vector $\mathbf{v} = [\mathbf{v}_1^\top, \dots, \mathbf{v}_n^\top]^\top \in \mathcal{N}(\mathbf{L})$ with $\mathbf{v}_i \in \mathbb{R}^d$. Then, $\mathbf{L}\mathbf{v} = \mathbf{0}$ implies that the i -th block vector of the product $\mathbf{L}\mathbf{v}$ satisfies:

$$\mathbf{L}_{ii}\mathbf{v}_i + \sum_{j=1, j \neq i}^n \mathbf{L}_{ij}\mathbf{v}_j = \mathbf{0} \quad (3.6)$$

After substitution from (2.13) and (2.14), (3.6) leads to:

$$\begin{aligned} \sum_{j=1, j \neq i}^n \mathbf{A}_{ij}\mathbf{v}_i - \sum_{j=1, j \neq i}^n \mathbf{A}_{ij}\mathbf{v}_j &= \mathbf{0} \quad \forall i \in \{1, \dots, n\} \\ \sum_{(i,j) \in \mathcal{E}} \mathbf{A}_{ij}(\mathbf{v}_i - \mathbf{v}_j) &= \mathbf{0}, \end{aligned} \quad (3.7)$$

which indicates that:

$$(\mathbf{v}_i - \mathbf{v}_j) = \mathbf{0} \quad \forall (i, j) \in \mathcal{E} \quad (3.8)$$

or

$$(\mathbf{v}_i - \mathbf{v}_j) \in \mathcal{N}(\mathbf{A}_{ij}) \quad \forall (i, j) \in \mathcal{E} \quad (3.9)$$

□

Lemma 3.2. *Given a matrix-weight-balanced graph \mathcal{G} with positive semi-definite weights \mathbf{A}_{ji} , its associated Laplacian matrix satisfies $\mathcal{N}(\mathbf{L}^\top) = \mathcal{N}(\mathbf{L})$.*

Proof. Let $\mathbf{v} = [\mathbf{v}_1^\top, \dots, \mathbf{v}_n^\top]^\top$ be a vector in the left nullspace of \mathbf{L} . Then, from $\mathbf{L}^\top \mathbf{v} = \mathbf{0}$, it follows that the i -th block vector of the product $\mathbf{L}^\top \mathbf{v}$ obeys:

$$\mathbf{L}_{ii}\mathbf{v}_i + \sum_{j=1, j \neq i}^n \mathbf{L}_{ji}\mathbf{v}_j = \mathbf{0}, \quad (3.10)$$

where the fact that the weights \mathbf{A}_{ji} are positive definite and thus symmetric and implicitly the blocks \mathbf{L}_{ij} are positive definite and thus symmetric for all $i, j \in \{1, \dots, n\}$, has been used. After substitution from (3.2), (3.10) can be rearranged into:

$$\begin{aligned}
& - \sum_{j=1, j \neq i}^n \mathbf{L}_{ji} \mathbf{v}_i + \sum_{j=1, j \neq i}^n \mathbf{L}_{ji} \mathbf{v}_j = \mathbf{0} \\
& \sum_{j=1, j \neq i}^n \mathbf{L}_{ji} (\mathbf{v}_i - \mathbf{v}_j) = \mathbf{0} \quad \forall i \in (1, \dots, n) \\
& - \sum_{j=1, j \neq i}^n \mathbf{A}_{ji} (\mathbf{v}_i - \mathbf{v}_j) = \mathbf{0} \quad \forall i \in (1, \dots, n) \\
& \sum_{j=1, j \neq i}^n \mathbf{A}_{ji} (\mathbf{v}_j - \mathbf{v}_i) = \mathbf{0} \quad \forall i \in (1, \dots, n) \\
& \sum_{(j,i) \in \mathcal{E}} \mathbf{A}_{ji} (\mathbf{v}_j - \mathbf{v}_i) = \mathbf{0} \\
& \sum_{(i,j) \in \mathcal{E}} \mathbf{A}_{ij} (\mathbf{v}_i - \mathbf{v}_j) = \mathbf{0}, \tag{3.11}
\end{aligned}$$

which implies that:

$$(\mathbf{v}_i - \mathbf{v}_j) = \mathbf{0} \quad \forall (i, j) \in \mathcal{E} \tag{3.12}$$

or:

$$(\mathbf{v}_i - \mathbf{v}_j) \in \mathcal{N}(\mathbf{A}_{ij}) \quad \forall (i, j) \in \mathcal{E} \tag{3.13}$$

and hence that $\mathcal{N}(\mathbf{L}^\top) = \mathcal{N}(\mathbf{L})$. \square

This turns out to be an important result. It shows that for a balanced digraph, the left and right nullspaces of the Laplacian are equal. This is generally a characteristic of Laplacians corresponding to undirected graphs. Which means every balanced digraph has an equivalent undirected graph. If we can find this transformation, we can exploit the results from the undirected case and modify it towards balanced digraphs. This is the idea that pushes us to look into mirror graphs.

The following lemma facilitates computations in the remainder of the paper.

Lemma 3.3. *Let the matrix-weight-balanced digraph \mathcal{G} have positive definite weights. Then, the matrix $\mathbf{L}_m = (\mathbf{L} + \mathbf{L}^\top)$ is the Laplacian associated with the mirror graph $\hat{\mathcal{G}}$.*

Proof. The blocks of the degree matrix of the mirror graph $\hat{\mathcal{G}}$ are:

$$\begin{aligned}\hat{\Delta}_{ii} &= \sum_{j=1, j \neq i}^n \hat{\mathbf{A}}_{ij} \\ &= \sum_{j=1, j \neq i}^n (\mathbf{A}_{ij} + \mathbf{A}_{ji})\end{aligned}\tag{3.14}$$

Since the digraph \mathcal{G} is balanced, its in-degree is equal to its out-degree, $\mathbf{D}_i^{out} = \mathbf{D}_i^{in}$, and it follows that:

$$\begin{aligned}\hat{\Delta}_{ii} &= 2(\mathbf{D}_i^{out}) \\ &= 2(\Delta_{ii}) \\ \hat{\Delta} &= 2(\Delta)\end{aligned}\tag{3.15}$$

and that:

$$\begin{aligned}\mathbf{L}_m &= (\mathbf{L} + \mathbf{L}^\top) \\ &= (\Delta - \mathbf{A}) + (\Delta^\top - \mathbf{A}^\top)\end{aligned}\tag{3.16}$$

The weights of \mathcal{G} being positive definite, the degree matrix of \mathcal{G} is symmetric, $\Delta = \Delta^\top$, and thus:

$$\begin{aligned}\mathbf{L}_m &= (\Delta - \mathbf{A}) + (\Delta^\top - \mathbf{A}^\top) \\ &= 2(\Delta) - (\mathbf{A} + \mathbf{A}^\top) \\ &= \hat{\Delta} - \hat{\mathbf{A}} \\ \mathbf{L}_m &= \hat{\mathbf{L}}\end{aligned}\tag{3.17}$$

□

Theorem 3.1. (*Stability*) *The first-order multi-agent system (2.16) is globally asymptotically stable if its matrix-weight-balanced digraph \mathcal{G} has positive definite weights.*

Proof. Consider the Lyapunov candidate:

$$V(\mathbf{x}) = \mathbf{x}^\top \mathbf{x},\tag{3.18}$$

whose derivative of V along the trajectories of system (2.16) is:

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}^\top \mathbf{x} + \mathbf{x}^\top \dot{\mathbf{x}} \quad (3.19)$$

$$= -\mathbf{x}^\top \mathbf{L}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{L} \mathbf{x} \quad (3.20)$$

$$= -\mathbf{x}^\top \hat{\mathbf{L}} \mathbf{x}. \quad (3.21)$$

Because the matrix-weighted Laplacian $\hat{\mathbf{L}}$ of an undirected graph $\hat{\mathcal{G}}$ is symmetric positive semidefinite [50], it follows that:

$$\dot{V}(\mathbf{x}) \leq 0,$$

and that the system (2.16) is globally stable.

The derivative of the Lyapunov function V vanishes, $\dot{V}(\mathbf{x}) = 0$, if and only if $\mathbf{x} \in \mathcal{N}(\hat{\mathbf{L}})$. From [50], the null space of the Laplacian of the mirror graph is $\mathcal{N}(\hat{\mathbf{L}}) = \text{span}\{\mathcal{R}, \{\mathbf{v} = [\mathbf{v}_1^\top, \dots, \mathbf{v}_n^\top]^\top \in \mathbb{R}^{n \times d} | (\mathbf{v}_j - \mathbf{v}_i) \in \mathcal{N}(\hat{\mathbf{A}}_{ij})\}\}$. For positive definite weights \mathbf{A}_{ij} , the weights $\hat{\mathbf{A}}_{ij}$ of the mirror graph $\hat{\mathcal{G}}$ are also positive definite. Then the nullspace of the mirror graph $\mathcal{N}(\mathbf{L}) = \mathcal{R}$ and any trajectory of the multi-agent system (2.16) globally asymptotically approaches $\mathcal{N}(\mathbf{L})$.

□

3.4 Necessary and Sufficient Condition For Average Consensus

Lemma 3.4. *Let the first-order multi-agent system (2.16) have interconnections represented by a matrix-weight-balanced graph with positive semi-definite weights. Then $\bar{\mathbf{x}} = \frac{1}{n}[\mathbf{1}_n^\top \otimes \mathbf{I}_d]\mathbf{x}$ is an invariant of the system.*

Proof. After substitution from (2.16), the time derivative of $\bar{\mathbf{x}} = \frac{1}{n}[\mathbf{1}_n^\top \otimes \mathbf{I}_d]\mathbf{x}$ becomes:

$$\dot{\bar{\mathbf{x}}} = -\frac{1}{n}[\mathbf{1}_n^\top \otimes \mathbf{I}_d]\mathbf{L}\mathbf{x} \quad (3.22)$$

Since every row of $[\mathbf{1}_n^\top \otimes \mathbf{I}_d]$ is a vector $\mathbf{v} \in \mathcal{N}(\mathbf{L}^\top)$ for a balanced graph:

$$\dot{\bar{\mathbf{x}}} = \mathbf{0} \quad (3.23)$$

□

Theorem 3.2. (*Average Consensus*) For a balanced graph the system (2.16) globally asymptotically converges to the equilibrium point $\mathbf{x}^* = \mathbf{1}_n \otimes \bar{\mathbf{x}}$ if only if $\mathcal{N}(\mathbf{L}) = \mathcal{R}$.

Proof. (*Necessity*) based on Theorem 2 in [50].

For a balanced graph assume (2.16) globally asymptotically converges to the equilibrium point $\mathbf{x}^* = \mathbf{1}_n \otimes \bar{\mathbf{x}}$ but $\mathcal{N}(\mathbf{L}) \neq \mathcal{R}$. From Lemma 1, there exists an $\mathbf{x}' \in \mathbb{R}^{n \times d}$ such that $\mathbf{L}\mathbf{x}' = \mathbf{0}$ and $\mathbf{x}' \notin \mathcal{R}$. Thus $\mathbf{x} = \mathbf{x}'$ is also an equilibrium point of (2.16) and any trajectory with $\mathbf{x}(0) = \mathbf{x}'$ stays at \mathbf{x}' for all $t \geq 0$. Thus \mathbf{x}^* is not globally asymptotically stable and contradicts our initial assumption.

(*Sufficiency*) From Theorem 3.1, the system is globally asymptotically stable. Then, let its equilibrium be:

$$\begin{aligned} \mathbf{x}^* = \lim_{t \rightarrow \infty} \mathbf{x}(t) &= (\mathbf{1}_n \otimes \mathbf{q}) \\ &= (\mathbf{1}_n \otimes \mathbf{I}_d)\mathbf{q} \end{aligned} \quad (3.24)$$

where $\mathbf{q} \in \mathbb{R}^d$.

From Lemma 3.4, the average of the agent states is invariant:

$$\bar{\mathbf{x}} = \frac{1}{n}[\mathbf{1}_n^\top \otimes \mathbf{I}_d]\mathbf{x} = \frac{1}{n}[\mathbf{1}_n^\top \otimes \mathbf{I}_d]\mathbf{x}_0 \quad (3.25)$$

At time $\lim_{t \rightarrow \infty}$, the average of the agent states is:

$$\begin{aligned} \bar{\mathbf{x}} &= \frac{1}{n}[\mathbf{1}_n^\top \otimes \mathbf{I}_d]\mathbf{x}^* \\ &= \frac{1}{n}[\mathbf{1}_n^\top \otimes \mathbf{I}_d][\mathbf{1}_n \otimes \mathbf{I}_d]\mathbf{q} \\ &= \frac{1}{n}\mathbf{1}_n^\top \mathbf{1}_n \mathbf{q} \\ &= \frac{1}{n}n\mathbf{q} \\ &= \mathbf{q}, \end{aligned} \quad (3.26)$$

and, after substitution back in (3.24), it follows that:

$$\begin{aligned} \mathbf{x}^* &= (\mathbf{1}_n \otimes \mathbf{I}_d)\bar{\mathbf{x}} \\ \mathbf{x}^* &= \mathbf{1}_n \otimes \bar{\mathbf{x}} \end{aligned} \quad (3.27)$$

□

Because positive definite matrix weights are necessary and sufficient for a matrix-weight-balanced graph to have a nullspace $\mathcal{N}(\mathbf{L}) = \mathcal{R}$, it follows that they are necessary and sufficient for average consensus of a first order multi-robot system with interconnections modelled by such a graph.

Chapter 4

Simulations

In this chapter, two main applications of matrix-weighted consensus algorithms are presented to verify the claims made by the theoretical work from Chapter 3.

The first section of the chapter considers the general problem of consensus for two cases of the communication graph:

1. Balanced Graph
2. Unbalanced Graph

It is shown using simulations in MATLAB that average consensus is achieved only in the case of a balanced graph. In the unbalanced case, the robots reach consensus but the steady state value is not the average of the initial states of the network of robots.

In the second section, formation control for a swarm of autonomous ground robots is presented as an application of the matrix-weight-balanced-consensus algorithm. It is a significant contribution because this algorithm can only be used as a result of having proved its convergence in the previous chapter.

A multi-robot system of six Clearpath Ridgeback holonomic robots start out at random locations in a map and the averaging consensus algorithm drives them to a predefined formation shape. The swarm robot simulations are done in Gazebo(Real time robot simulator) using ROS(Robot operating system) as the middleware. The reason for choosing this specific set of tools is because the code generated in the simulations will work as-is on a set of real robots.

4.1 Consensus

Initial Conditions

The positions of the robots in cartesian space are given by,

$$\mathbf{x}_1 = [0, 4, 4]^\top \quad (4.1)$$

$$\mathbf{x}_2 = [-3, 2, 2]^\top \quad (4.2)$$

$$\mathbf{x}_3 = [0, 0, 0]^\top \quad (4.3)$$

$$\mathbf{x}_4 = [5, 0, 0]^\top \quad (4.4)$$

$$\mathbf{x}_5 = [8, 3, 2]^\top \quad (4.5)$$

$$\mathbf{x}_6 = [5, 5, 4]^\top \quad (4.6)$$

The average of the initial states,

$$\mathbf{x}_{Av} = \frac{(\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 + \mathbf{x}_4 + \mathbf{x}_5 + \mathbf{x}_6)}{6} = [2.5, 2.33, 2]^\top \quad (4.7)$$

4.1.1 Balanced Graph

Consider a 6 robot network modeled as single integrators. The topology of the network is shown in Figure. 4.1.

The positive definite matrices representing the edges are,

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 4 \end{bmatrix} \quad (4.8)$$

$$\mathbf{B} = \begin{bmatrix} 5 & -1 & 0 \\ -1 & 6 & -1 \\ 0 & -1 & 7 \end{bmatrix} \quad (4.9)$$

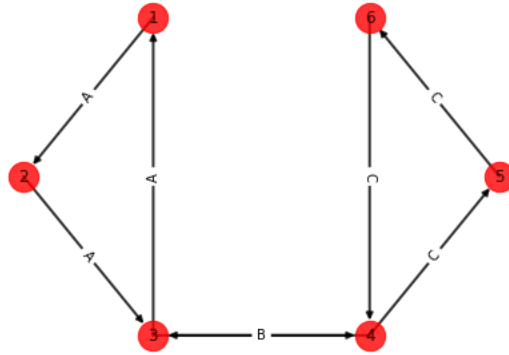


Figure 4.1: Communication Topology of Balanced Graph

$$\mathbf{C} = \begin{bmatrix} 8 & -1 & 0 \\ -1 & 9 & -1 \\ 0 & -1 & 10 \end{bmatrix} \quad (4.10)$$

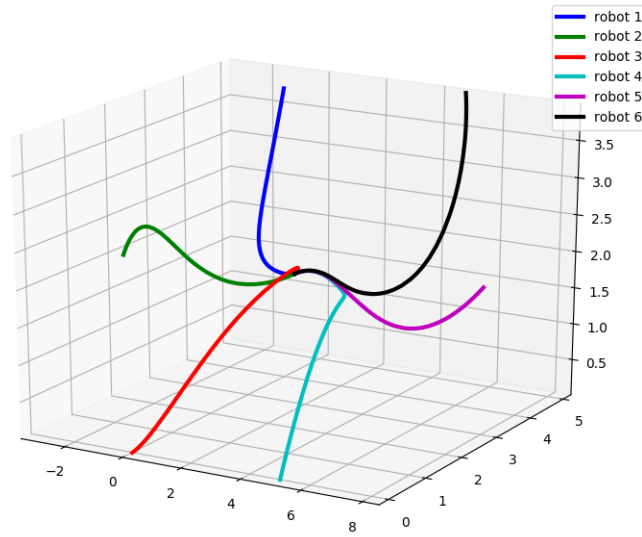


Figure 4.2: Consensus of robots for Balanced Graph

The steady state values of the consensus point in the balanced case is,

$$\bar{\mathbf{x}} = [2.5, 2.3, 2]^T \quad (4.11)$$

For a balanced graph,

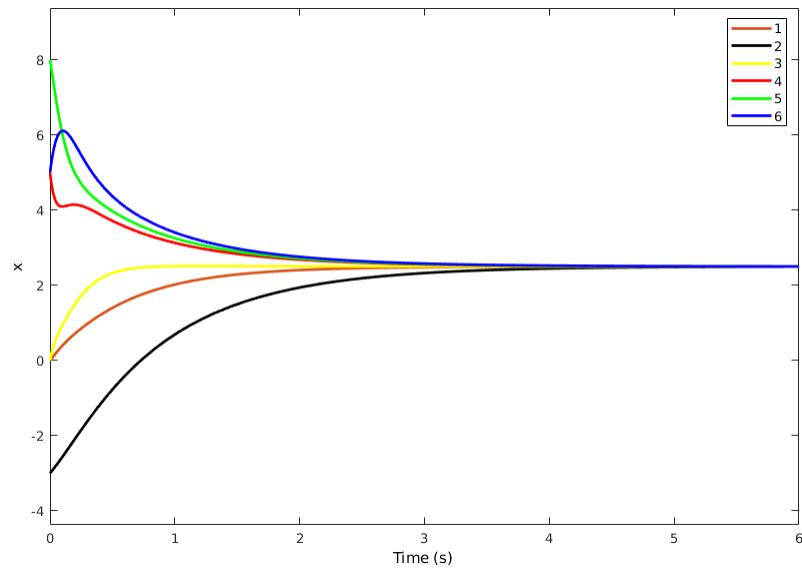


Figure 4.3: Trajectory of states in x (Balanced Graph)

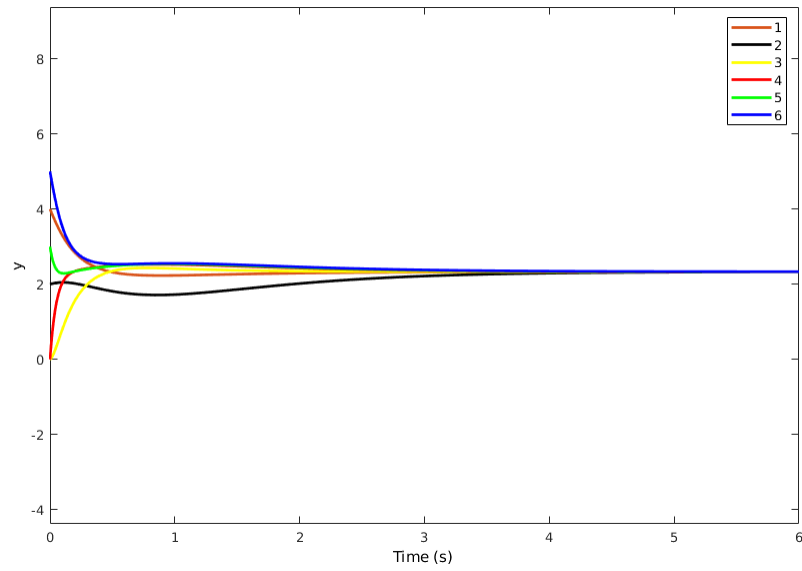


Figure 4.4: Trajectory of states in y (Balanced Graph)

$$\bar{\mathbf{x}} = \mathbf{x}_{Av} \quad (4.12)$$

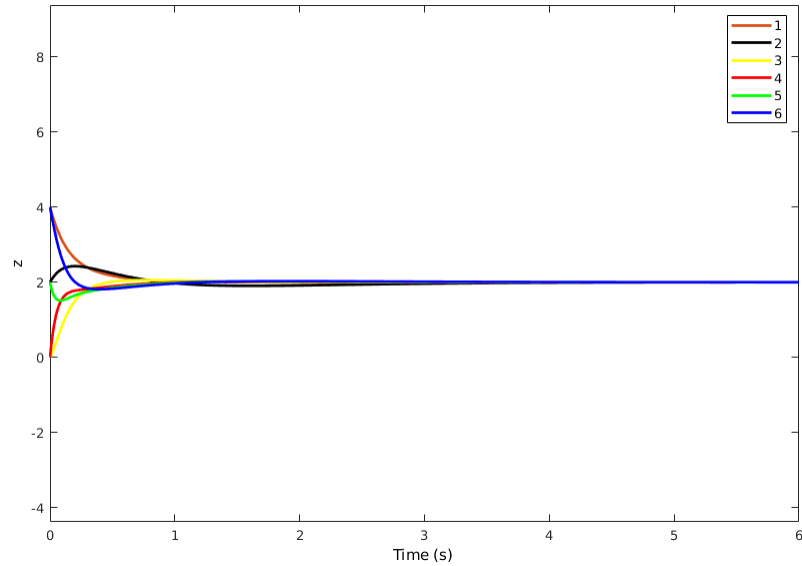


Figure 4.5: Trajectory of states in z (Balanced Graph)

4.1.2 Unbalanced Graph

For the same set of initial conditions and matrix weights as the previous case, the following are the results when the graph is not balanced.

The topology of the network is shown in (Figure. 4.6).

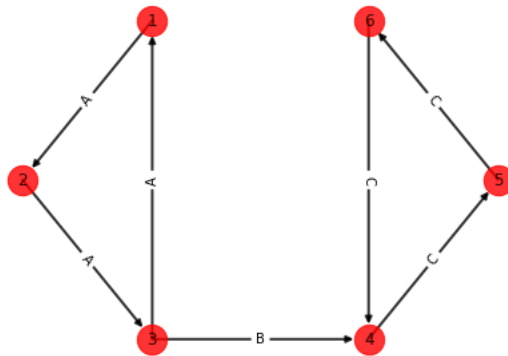


Figure 4.6: Communication Topology of Unbalanced Graph

The simulation results are presented below,

In the unbalanced case, we see that although the robots reach consensus, they do not reach average consensus.

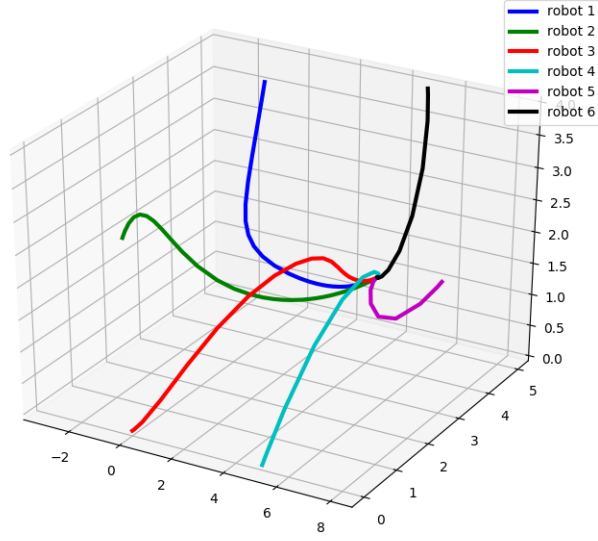
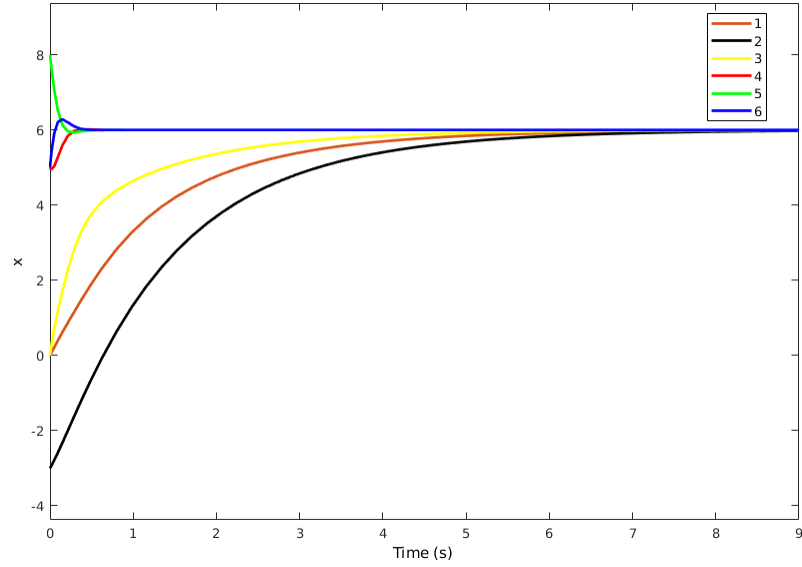
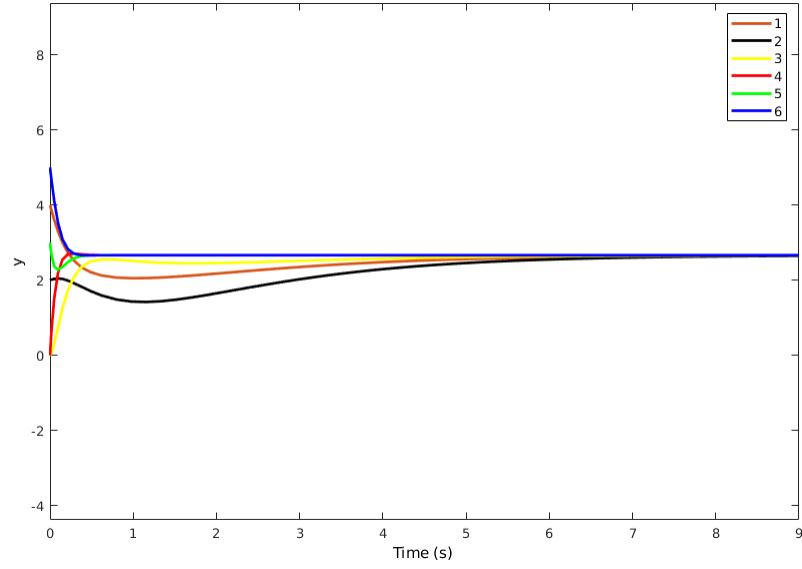
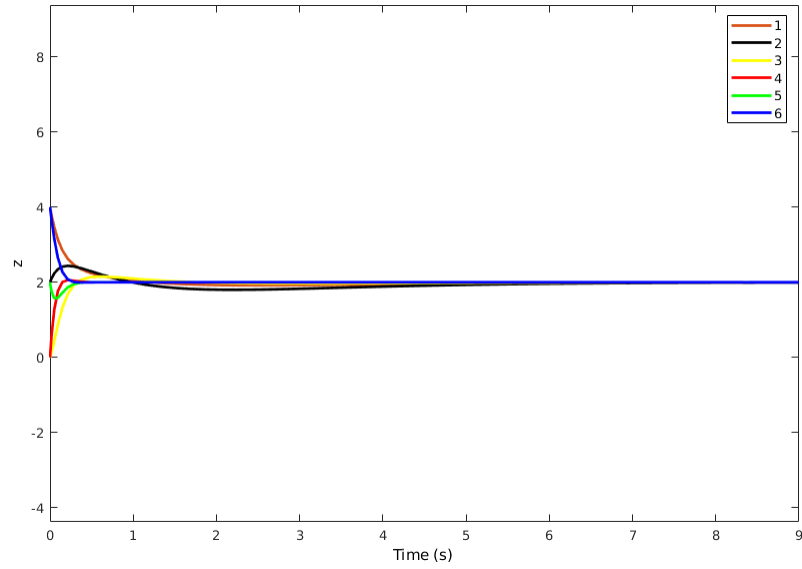


Figure 4.7: Consensus of robots for Unbalanced Graph

Figure 4.8: Trajectory of states in x (Unbalanced Graph)

4.2 Formation Control

The consensus law can be used in the formation control problem only because the asymptotic convergence for matrix-weighted balanced graphs has been proved. This represents one possible application of matrix-weight-balanced digraphs in the multi-robot systems community. To show the applicability of the theory to real robots, the

Figure 4.9: Trajectory of states in y (Unbalanced Graph)Figure 4.10: Trajectory of states in z (Unbalanced Graph)

formation control algorithm is implemented on a real world robotics simulation environment called Gazebo. The simulation uses 6 Clearpath Ridgeback robots which are holonomic robots popularly used in warehouses. The setting up of robot simulations in Gazebo, the matrix-weighted-consensus controller in SIMULINK and the bridging provided by ROS is shown in (Figure. 4.11).

Consider a variant of the consensus law (2.15),

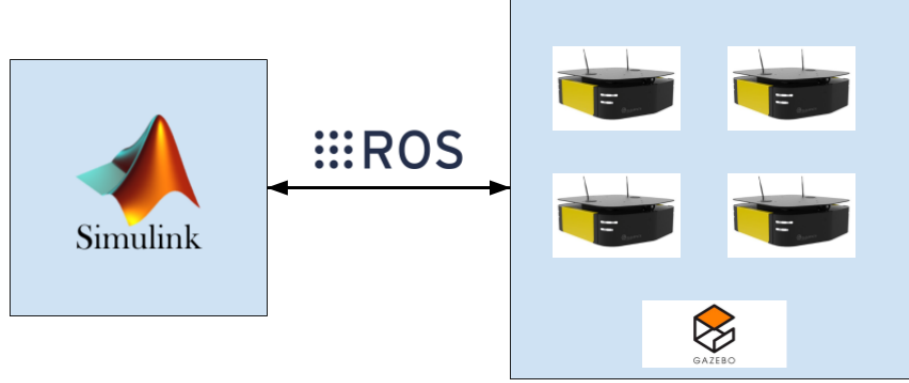


Figure 4.11: Setting Up Gazebo with SIMULINK

$$\dot{\mathbf{x}}_i = \sum_{v_j \in \mathcal{N}_i} \mathbf{A}_{ij}((\mathbf{x}_j - \delta_j) - (\mathbf{x}_i - \delta_i)) \quad (4.13)$$

Where $\delta_{ji} = (\delta_j - \delta_i) \in \mathbb{R}^d$ represents the desired relative inter agent separation between agents j and i .

If the conditions for consensus in matrix-weighted balanced digraphs is satisfied then $(\mathbf{x}_j - \delta_j) \rightarrow (\mathbf{x}_i - \delta_i)$ and hence, $(\mathbf{x}_j - \mathbf{x}_i) \rightarrow (\delta_{ji})$. By specifying $\delta_{ji}(t)$, any formation shape can be produced [40].

Let six holonomic robots be positioned randomly in 2d space as shown in (Figure. 4.12).

The initial positions of the robots are $\mathbf{x}_1 = [6, 5]^\top$, $\mathbf{x}_2 = [-4, 6]^\top$, $\mathbf{x}_3 = [-5, -5]^\top$, $\mathbf{x}_4 = [3, 5]^\top$, $\mathbf{x}_5 = [14, -4]^\top$, $\mathbf{x}_6 = [10, 7]^\top$.

The final desired position of each robot must be specified in the local coordinate frame (local origin attached to one of the robots) as $\delta_1 = [5, 8.6]^\top$, $\delta_2 = [2.5, 4.3]^\top$, $\delta_3 = [0, 0]^\top$, $\delta_4 = [5; 0]^\top$, $\delta_5 = [10; 0]^\top$, $\delta_6 = [7.5; 4.3]^\top$.

The communication topology is shown by (Figure. 4.13).

The matrix weights are all equal, $\mathbf{A} = \begin{bmatrix} 2 & -1 \\ -1 & 3 \end{bmatrix}$. Then using the consensus law (4.13), the robots converge to the desired formation shape shown in (Figure. 4.13)

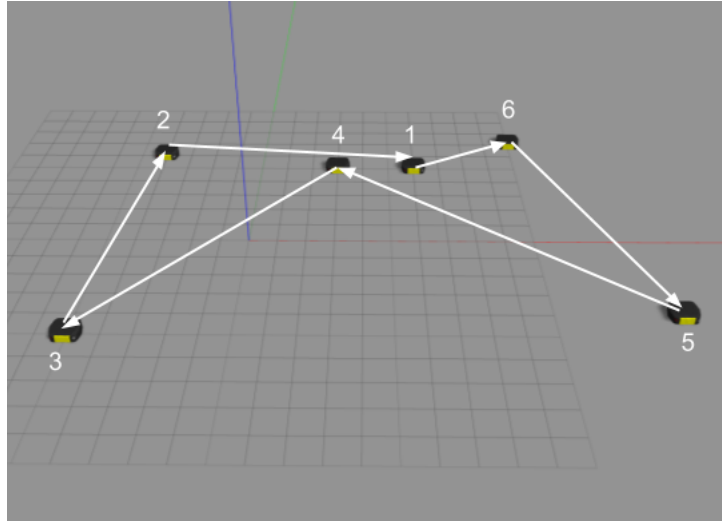


Figure 4.12: Initial Configuration of Robots

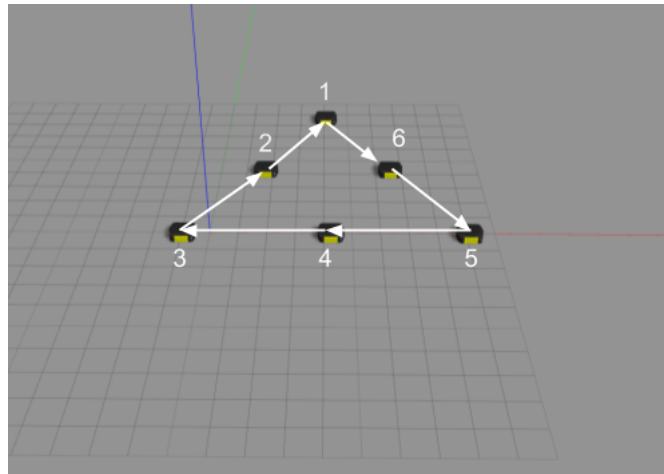


Figure 4.13: Final Formation

Chapter 5

Conclusion and Future Work

Given a multi-robot system, graph theory plays a very important role in the modelling of dynamics and the interaction of the robots. A unified approach to dealing with a collection of a large number of robots is possible by combining the areas of control theory and spectral graph theory.

For a long time, scalar-weighted consensus algorithms have been studied and a lot of work has been presented on the same. Since 2018 however, matrix-weighted consensus, i.e consensus algorithms where the graph edges are weighted by matrices instead of scalar numbers has received much attention. This has been mostly from the multi-agent and the social network community.

While the concept of matrix weights seems abstract from a robotics/control theory perspective, the area looks promising and offers a rich and a much more simpler class of solutions to some of the common problems in the multi-robot systems community. And this simplicity of the controller drives research in this new and exciting area. Matrix-weights can be viewed as abstract forms of modelling a tensor interaction between robots. As a result this is an interesting research direction to pursue.

While the use of matrix-weighted-consensus algorithms for the general case of undirected graphs has been done before, their use in directed graphs is still an open problem. This thesis then is a first step towards exploring the new area.

The main challenge in implementing controllers based on matrix-weighted consensus on digraphs lies in proving that the algorithm converges and hence the controller is stable. The tools used for analysis in the matrix-weighted-consensus case for undirected graphs cannot be used in our case. This necessitates the need to identify a bridge to tie up work in the two areas. As was the case in the scalar-weighted-consensus, balanced graphs play that fundamentally important role.

The Laplacian of matrix-weighted *balanced* digraphs is shown to have the same characteristic as the Laplacian of a matrix-weighted undirected graph. This property leads to the idea that every balanced matrix-weighted digraph has a corresponding matrix-weighted undirected graph. The transformation between the two holds the key to solving a small yet significant area of research under matrix-weighted digraphs. This transformation was shown to be the mirror graph or the mirroring operation. By exploiting the properties of a mirror graph, several result from the matrix-weighted undirected case was extended to the directed case. Lyapunov analysis provided the stability for matrix-weighted-consensus algorithms on balanced graphs.

As a result of the controller being stable, a novel formation control algorithm based on matrix-weighted-consensus is proposed. The usefulness of the algorithm in the formation control of a swarm of autonomous ground robots(Clearpath Ridgeback Warehouse Robots) is presented.

5.1 Future Work

While this thesis is the first work to consider the consensus problem in matrix-weight-balanced digraphs, the area of general digraphs is still an open problem. Possible future research includes finding the conditions for consensus in any given matrix-weighted digraph.

Since time varying scalar-weights are versatile and provide a way to implement a large number of multi-robot system behaviour and embed additional constraints, consensus in multi-robot systems with time-varying matrix-weights can also be considered as a promising direction for future research.

Apart from these, real world constraints in the form of bounded actuation, controller constraints, sensor constraints and non-linear dynamics of the multi-robot systems are all areas that need further research.

Matrix-weighted-consensus algorithms have a wide range of applications. Time varying matrix-weights can be used with potential field based controllers to unify robot behaviours like formation control, obstacle avoidance, inter robot collision avoidance and in cooperative hunting. This unified deterministic motion planning algorithm is the most useful application of the matrix-weighted-consensus algorithm. A lot of robotics research has lately focused on integrating perception, planning and control into a consolidated framework and matrix-weighted-consensus algorithm might just hold the key to success in the area.

Bibliography

- [1] Hyo-Sung Ahn, Quoc Van Tran, Minh Hoang Trinh, Kevin L. Moore, Mengbin Ye, and Ji Liu. Cooperative opinion dynamics on multiple interdependent topics: Modeling and analysis. *arXiv e-prints*, page arXiv:1807.04406, July 2018.
- [2] H. Bai, R. A. Freeman, and K. M. Lynch. Robust dynamic average consensus of time-varying inputs. In *49th IEEE Conference on Decision and Control (CDC)*, pages 3104–3109, Dec 2010.
- [3] Kai Cai and Hideaki Ishii. Average consensus on general strongly connected digraphs. *Automatica*, 48(11):2750 – 2761, 2012.
- [4] M. Cao, A. S. Morse, and B. D. O. Anderson. Agreeing asynchronously. *IEEE Transactions on Automatic Control*, 53(8):1826–1838, Sep. 2008.
- [5] Y. Cao and W. Ren. Finite-time consensus for second-order multi-agent networks with inherent nonlinear dynamics under an undirected fixed graph. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 3452–3457, Dec 2011.
- [6] Yongcan Cao and Wei Ren. Multi-vehicle coordination for double-integrator dynamics under fixed undirected/directed interaction in a sampled-data setting. *International Journal of Robust and Nonlinear Control*, 20(9):987–1000, 2010.
- [7] Yongcan Cao and Wei Ren. Sampled-data discrete-time coordination algorithms for double-integrator dynamics under dynamic directed interaction. *International Journal of Control*, 83(3):506–515, 2010.
- [8] F. Chen, Y. Cao, and W. Ren. Distributed average tracking of multiple time-varying reference signals with bounded derivatives. *IEEE Transactions on Automatic Control*, 57(12):3169–3174, Dec 2012.

- [9] F. Chen, W. Ren, W. Lan, and G. Chen. Distributed average tracking for reference signals with bounded accelerations. *IEEE Transactions on Automatic Control*, 60(3):863–869, March 2015.
- [10] Jorge Cortes. Discontinuous Dynamical Systems: A tutorial on solutions, nonsmooth analysis, and stability. *arXiv e-prints*, page arXiv:0901.3583, Jan 2009.
- [11] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson. Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*, 57(5):1291–1297, May 2012.
- [12] D. V. Dimarogonas and K. J. Kyriakopoulos. On the rendezvous problem for multiple nonholonomic agents. *IEEE Transactions on Automatic Control*, 52(5):916–922, May 2007.
- [13] Yuan Fan, Runzhe Han, and Jianbin Qiu. Graph-balancing algorithms for average consensus over directed networks. *International Journal of Systems Science*, 47(1):135–148, 2016.
- [14] L. Fang and P. J. Antsaklis. Asynchronous consensus protocols using nonlinear paracontractions theory. *IEEE Transactions on Automatic Control*, 53(10):2351–2355, Nov 2008.
- [15] Joseph Alexander Fax. *Optimal and cooperative control of vehicle formations*. PhD thesis, California Institute of Technology, 2002.
- [16] R. A. Freeman, P. Yang, and K. M. Lynch. Stability and convergence properties of dynamic average consensus estimators. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 338–343, Dec 2006.
- [17] Noah E. Friedkin, Anton V. Proskurnikov, Roberto Tempo, and Sergey E. Parsegov. Network science on belief system dynamics under logic constraints. *Science*, 354(6310):321–326, 2016.
- [18] Huan Gao and Yongqiang Wang. Dynamics Based Privacy Protection for Average Consensus on Directed Graphs. *arXiv e-prints*, page arXiv:1812.02255, December 2018.

- [19] B. Gharesifard and J. Corts. Distributed strategies for making a digraph weight-balanced. In *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 771–777, Sep. 2009.
- [20] B. Gharesifard and J. Corts. When does a digraph admit a doubly stochastic adjacency matrix? In *Proceedings of the 2010 American Control Conference*, pages 2440–2445, June 2010.
- [21] Bahman Gharesifard and Jorge Corts. Distributed strategies for generating weight-balanced and doubly stochastic digraphs. *European Journal of Control*, 18(6):539 – 557, 2012.
- [22] Paolo Robuffo Giordano, Antonio Franchi, Cristian Secchi, and Heinrich H Blthoff. A passivity-based decentralized strategy for generalized connectivity maintenance. *The International Journal of Robotics Research*, 32(3):299–323, 2013.
- [23] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Transactions on Automatic Control*, 50(11):1867–1872, Nov 2005.
- [24] Meng Ji and Magnus Egerstedt. Distributed coordination control of multi-agent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, 2007.
- [25] Fangcui Jiang and Long Wang. Finite-time information consensus for multi-agent systems with fixed and switching topologies. *Physica D: Nonlinear Phenomena*, 238(16):1550 – 1560, 2009.
- [26] Y. Karayiannidis, D. V. Dimarogonas, and D. Kragic. Multi-agent average consensus control with prescribed performance guarantees. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 2219–2225, Dec 2012.
- [27] M. Kefayati, M. S. Talebi, B. H. Khalaj, and H. R. Rabiee. Secure consensus averaging in sensor networks using random offsets. In *2007 IEEE International Conference on Telecommunications and Malaysia International Conference on Communications*, pages 556–560, May 2007.
- [28] S. S. Kia, J. Corts, and S. Martnez. Singularly perturbed algorithms for dynamic average consensus. In *2013 European Control Conference (ECC)*, pages 1758–1763, July 2013.

- [29] Solmaz S. Kia, Jorge Cortes, and Sonia Martinez. Distributed event-triggered communication for dynamic average consensus in networked systems. *Automatica*, 59:112 – 119, 2015.
- [30] Byung-Hun Lee and Hyo-Sung Ahn. Distributed formation control via global orientation estimation. *Automatica*, 73:125 – 129, 2016.
- [31] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*, volume 33. Princeton University Press, 2010.
- [32] H. Moradian and S. S. Kia. On robustness analysis of a dynamic average consensus algorithm to communication delay. *IEEE Transactions on Control of Network Systems*, pages 1–1, 2018.
- [33] C. Nowzari and J. Cortes. Zeno-free, distributed event-triggered communication and control for multi-agent average consensus. In *2014 American Control Conference*, pages 2148–2153, June 2014.
- [34] Cameron Nowzari and Jorge Cortes. Distributed event-triggered coordination for average consensus on weight-balanced digraphs. *Automatica*, 68:237 – 244, 2016.
- [35] Erfan Nozari, Pavankumar Tallapragada, and Jorge Cortes. Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design. *Automatica*, 81:221 – 231, 2017.
- [36] R. Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 8179–8184, Dec 2005.
- [37] R. Olfati-Saber. Ultrafast consensus in small-world networks. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 2371–2378 vol. 4, June 2005.
- [38] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6698–6703, Dec 2005.
- [39] Sergey E. Parsegov, Anton V. Proskurnikov, Roberto Tempo, and Noah E. Friedkin. Novel Multidimensional Models of Opinion Dynamics in Social Networks. *arXiv e-prints*, page arXiv:1505.04920, May 2015.

- [40] Wei Ren. Consensus based formation control strategies for multi-vehicle systems. In *2006 American Control Conference*, pages 6–, 06 2006.
- [41] Wei Ren. Distributed leaderless consensus algorithms for networked eulerlagrange systems. *International Journal of Control*, 82(11):2137–2149, 2009.
- [42] Wei Ren and R. W. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, 05 2005.
- [43] N. Rezazadeh and S. S. Kia. Privacy preservation in a continuous-time static average consensus algorithm over directed graphs. In *2018 Annual American Control Conference (ACC)*, pages 5890–5895, June 2018.
- [44] R. O. Saber and R. M. Murray. Consensus protocols for networks of dynamic agents. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 2, pages 951–956, 06 2003.
- [45] R. Olfati Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 01 2007.
- [46] R. Olfati Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.
- [47] A. Tahbaz-Salehi and A. Jadbabaie. A necessary and sufficient condition for consensus over random networks. *IEEE Transactions on Automatic Control*, 53(3):791–795, April 2008.
- [48] Y. Tian and C. Liu. Consensus of multi-agent systems with diverse input and communication delays. *IEEE Transactions on Automatic Control*, 53(9):2122–2128, Oct 2008.
- [49] M. H. Trinh, M. Ye, H. Ahn, and B. D. O. Anderson. Matrix-weighted consensus with leader-following topologies. In *2017 11th Asian Control Conference (ASCC)*, pages 1795–1800, 12 2017.
- [50] Minh Hoang Trinh, Chuong Van Nguyen, Young-Hun Lim, and Hyo-Sung Ahn. Matrix-weighted consensus and its applications. *Automatica*, 89:415 – 419, 2018.

- [51] S. E. Tuna. Conditions for synchronizability in arrays of coupled linear systems. *IEEE Transactions on Automatic Control*, 54(10):2416–2420, Oct 2009.
- [52] S. E. Tuna. Observability through a matrix-weighted graph. *IEEE Transactions on Automatic Control*, 63(7):2061–2074, July 2018.
- [53] S. Emre Tuna. Synchronization under matrix-weighted laplacian. *Automatica*, 73:76 – 81, 2016.
- [54] Xiaobo Wang, Juelong Li, Jianchun Xing, Ronghao Wang, Liqiang Xie, and Xiaocheng Zhang. A novel finite-time average consensus protocol for multi-agent systems with switching topology. *Transactions of the Institute of Measurement and Control*, 40(2):606–614, 2018.
- [55] F. Xiao and L. Wang. State consensus for multi-agent systems with switching topologies and time-varying delays. *International Journal of Control*, 79(10):1277–1284, 2006.
- [56] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 63–70, April 2005.
- [57] G. Xie, H. Liu, L. Wang, and Y. Jia. Consensus in networked multi-agent systems via sampled control: Fixed topology case. In *2009 American Control Conference*, pages 3902–3907, June 2009.
- [58] G. Xie, H. Liu, L. Wang, and Y. Jia. Consensus in networked multi-agent systems via sampled control: Switching topology case. In *2009 American Control Conference*, pages 4525–4530, June 2009.
- [59] Y. Xu, W. Liu, and J. Gong. Stable multi-agent-based load shedding algorithm for power systems. *IEEE Transactions on Power Systems*, 26(4):2006–2014, Nov 2011.
- [60] S. Yang, S. Tan, and J. Xu. Consensus based approach for economic dispatch problem in a smart grid. *IEEE Transactions on Power Systems*, 28(4):4416–4426, Nov 2013.

- [61] Mehran Zareh, Carla Seatzu, and Mauro Franceschelli. Consensus on the average in arbitrary directed network topologies with time-delays. *arXiv e-prints*, page arXiv:1502.04364, February 2015.
- [62] Michael M Zavlanos and George J Pappas. Potential fields for maintaining connectivity of mobile networks. *IEEE Transactions on robotics*, 23(4):812–816, 2007.
- [63] Ya Zhang and Yu-Ping Tian. Consentability and protocol design of multi-agent systems with stochastic switching topology. *Automatica*, 45(5):1195 – 1201, 2009.
- [64] S. Zhao and D. Zelazo. Bearing-based formation stabilization with directed interaction topologies. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 6115–6120, 12 2015.
- [65] Shiyu Zhao and Daniel Zelazo. Localizability and distributed protocols for bearing-based network localization in arbitrary dimensions. *Automatica*, 69:334 – 341, 2016.