הטכניון
מכון טכנולוגי לישראל
TECHNION
Israel Institute of Technology

ספריות הטכניון
The Technion Libraries

בית הספר ללימודי מוסמכים ע"ש ארווין וג'ואן ג'ייקובס
Irwin and Joan Jacobs Graduate School

Leader Identification of Multi-Agents Systems Under Semi-Autonomous Consensus Protocol

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Autonomous Systems
and Robotics

Evyatar Matmon

Submitted to the Senate
of the Technion — Israel Institute of Technology
Heshvan 5786 Haifa October 2025

The Research Thesis Was Done in Autonomous Systems and Robotics Under the Supervision of Prof. Daniel Zelazo.

The author of this thesis states that the research, including the collection, processing and presentation of data, addressing and comparing to previous research, etc., was done entirely in an honest way, as expected from scientific research that is conducted according to the ethical standards of the academic world. Also, reporting the research and its results in this thesis was done in an honest and complete manner, according to the same standards.

Abstract

The consensus protocol is fundamental multi-agent systems protocol which enables synchronization of state trajectories [1]. In semi-autonomous consensus protocols, selected agents, called leaders, receive external inputs, while others, the followers, rely only on interactions with their neighbors.

This work addresses the problem of leader identification—determining the leader nodes by observing only agent trajectories. Using the system dynamics, we focus on analyzing the Fiedler vector, associated with the smallest non-zero eigenvalue of the interconnection matrix. Previous work showed a link between this vector and velocity trajectories of the agents [2]. Our approach involves analyzing a sequence of graphs with increasing number of nodes and the associated Fiedler vectors. We demonstrate that the components of the Fiedler vector converge to a quantity related to the degree of each node. We then provide a sufficient condition for sufficiently large graphs to achieve a separation of the Fiedler vector components into two distinct groups corresponding to the leaders and followers. We use this result to provide an estimation algorithm of the leader nodes by using only measurements of agent velocities. We also provide some numerical studies to demonstrate our results.

Contents

List of Figures

Abbreviations and Notations

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^{m\times n}$	Real matrix of dimension $m \times n$
$\mathbb{1}_n, n$	n-dimensional vector with all entries 1 or 0
I_n	$n \times n$ identity matrix
$\lvert x\rvert$	Absolute value of a scalar
$\lVert x\rVert$	Norm of a vector (Euclidean norm unless specified)
$x^\top$	Transpose of vector or matrix x
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$
$n = \lvert\mathcal{V}\rvert$	Number of nodes (vertices) in the graph
$m = \lvert\mathcal{E}\rvert$	Number of edges in the graph
$A(\mathcal{G})$	Adjacency matrix of graph $\mathcal{G}$
$D(\mathcal{G})$	Degree matrix of graph $\mathcal{G}$
$L(\mathcal{G})$	Laplacian matrix of graph $\mathcal{G}$, $L = D - A$
L_B	Grounded Laplacian matrix (obtained by removing leader rows/columns)
$\lambda_i(M)$	i-th eigenvalue of matrix M (ordered increasingly)
$\mathbf{v}_i(M)$	Eigenvector associated with $\lambda_i(M)$
$\lambda_{\min}(M), \lambda_{\max}(M)$	Smallest and largest eigenvalues of M
λ_F, v_F	Fiedler eigenvalue and eigenvector of grounded Laplacian L_B
$\mathcal{N}(i)$	Neighborhood of node i (all adjacent nodes)
$\mathcal{V}_\ell, \mathcal{V}_f$	Leader and follower agent groups
$x_i(t)$	State of agent i at time t
$u_i(t)$	Control input applied to agent i at time t

Chapter 1

Introduction

Multi-agent systems refer to complex configurations of independent dynamic units that can interact with each other through designated network protocols. In such a networked system, agents interact with each other to fulfill a higher-level purpose. The rationale for employing such a system lies in the potential to enhance energy efficiency, cost-effectiveness, and resilience through the utilization of distributed protocols among agents. There are several examples of employing multi-agent systems, and we will outline some key applications in this domain: Applications of multi-agent coordination span a wide range of domains. One main example is **formation flight**, which focuses on establishing spatial coordination among agents to accomplish system-level goals. This approach has proven useful in surveillance, reconnaissance, search operations, and environmental monitoring [3]. More recently, it has also found application in spaceborne optical interferometry, where distributed spacecraft carrying optical instruments are coordinated to enhance the efficiency and robustness of space interferometry missions [4, 5]. Another important area is **sensor networks**, where spatially distributed sensors are deployed to monitor diverse physical or environmental conditions, enabling applications in areas such as environmental sensing, infrastructure monitoring, and precision agriculture [6, 7]. Similarly, **energy networks** illustrate the role of distributed coordination in the distribution of energy from generators to consumers. This field has gained increasing attention with the rise of smart grids, which aim to improve the efficiency, resilience, and sustainability of energy distribution systems [8, 9].

Among the fundamental coordination strategies within multi-agent frameworks lies the *consensus* protocol, or sometime called *agreement*, aiming to accomplish complete or partial synchronization of state trajectories across all agents. The basic mechanism of the consensus protocol is that every agent can sense some others agents, usually within some pre-specified range or *ad hoc* information exchange network. This information is then using to compute a control command aimed at achieving the consensus objective [1]. Numerous studies have addressed the consensus problem, including investigations of consensus under switching topologies and communication delays [10], agreement over random networks [11], finite-time consensus [12], and stochastic consensus with noisy

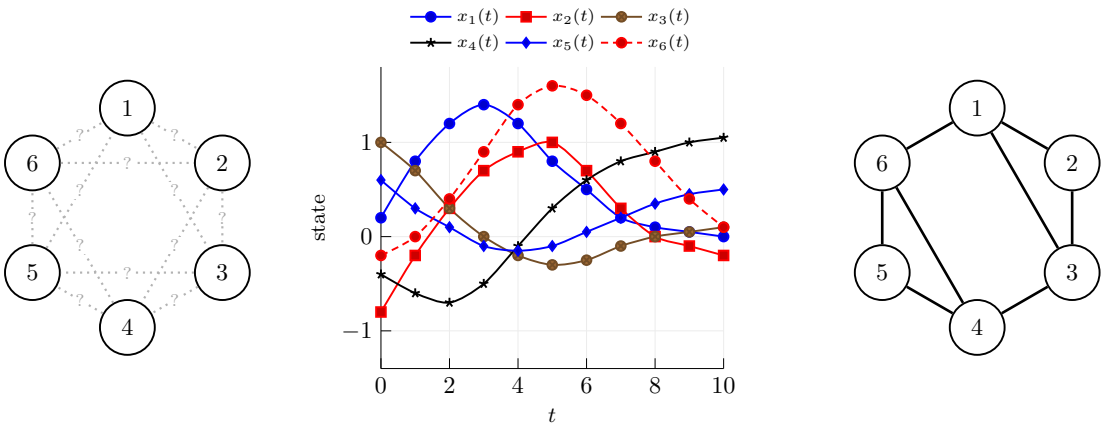
communication [13].

The term *semi-autonomous consensus*, or sometimes called *leader-follower consensus*, characterizes scenarios in which select agents, often denoted as the *leaders*, receive external inputs. The usage of an external input to the system can, for example, be used to maneuver or manipulate the system to some desired trajectory or configuration. In [14], the use of leader agents is examined to guarantee prescribed performance in the system's transient behavior. In other works, such as [15], external noise inputs are injected into certain agents to evaluate the network's performance. In [16], the performance of a human-in-the-loop system is analyzed, where some agents are directly controlled by human operators, and the work in [2] addresses the selection of leader agents in the network to enhance system performance.

The widespread use of these systems raises concerns about their security from malicious attack. If a malicious attacker strategically directs an attack towards the most critical agents, the entire system could be disrupted, or even eliminated, with relatively minimal effort [17]. If an external observer can infer the structure of the network topology, they may also be able to identify which agents receive external inputs—an aspect that is critical to the operation of the system.

The **network identification problem** concerns the task of reconstructing or inferring the underlying interaction topology of a multi-agent system from observed data. In other words, given measurements of the agents' states or outputs, one seeks to determine which agents exchange information and with what strength. This problem is fundamental in areas such as systems biology [18], neuroscience [19], social networks [20], and engineering systems [21]. A central challenge is that the observed collective behavior does not uniquely determine the network structure without additional assumptions or prior knowledge. Accordingly, approaches to network identification can be viewed along a spectrum: at one end, there are methods that rely on extensive prior knowledge or the ability to actively perturb the network (e.g., through controlled inputs), while at the other end there are data-driven approaches that operate with minimal prior information and limited or no interaction with the network [22, 23, 24]. In the context of multi-agent coordination, identifying the interaction graph is crucial for understanding how local rules give rise to global behaviors, as well as for designing control strategies that are robust to uncertainties in the communication topology. An illustraion of network topology identification is presented in Figure 1.1.

Traditional input–output identification techniques generally assume that both the external inputs and the system dynamics are known, which allows a straightforward reconstruction of the network model. In adversarial contexts, however, such assumptions are unrealistic: an attacker may not observe the control inputs at all. Li et al. [17] demonstrate that even under this information deficit, an adversary can still learn the interaction rules of a multi-robot formation purely from state observations, and then strategically exploit this knowledge to disrupt the system through a so-called "sneak attack." This perspective emphasizes that security analyses must account not only



(a) Unknown graph. (b) Observed trajectories. (c) Recovered graph.

Figure 1.1: Conceptual overview of the network identification problem: (a) the interaction graph is unknown; (b) trajectories are observed; (c) after identification, the edges are recovered.

for adversaries with full system knowledge, but also for those who incrementally infer network structure from minimal information before launching an attack.

In this study, we investigate the problem of network topology identification from the perspective of an external observer, assuming limited prior knowledge about the network. Our objective is to identify a specific subset of the topology—namely, the leader agents—which play a critical role in the network's operation.

1.1 Literature Review

In this section we review the literature on the network identification problem, with a focus on methods proposed for reconstructing interaction topologies from observed data. Existing approaches can be grouped into several categories, reflecting different assumptions about data availability and system structure:

- **Full observation:** Methods where the states of all agents in the network are fully observed. These approaches typically assume complete access to the system's state evolution and use it to reconstruct the network [25, 23].
- **Partial observation:** Techniques that rely on observations from only a subset of agents. These problems are more challenging and often involve inferring the unobserved part of the network using indirect information [26, 27].
- **Leader-follower identification:** Focuses on identifying leader nodes or external input points in the network. These leaders often influence the dynamics of follower agents and are critical for understanding system behavior [17].
- **Spectral approaches:** Use the spectral properties (e.g., eigenvalues and eigenvectors) of matrices like the Laplacian or adjacency matrix to infer global or local

network structure. These methods are often analytical and provide insights into connectivity and topology [28, 29].

- **Structural identification:** Aims to reconstruct the underlying graph topology based on system input-output data, interaction patterns, or transfer functions. This category includes both exact and approximate recovery of the adjacency or Laplacian matrix [26, 21].

In our work, we focus on identifying leader agents under a full observation setting, based on input-output relationships, where the input signals are unknown. We begin by discussing studies that address the full observation setting, where network identification is based solely on the system's dynamic response to initial conditions, without any external input applied.

The work discussed in [30] is focused on the estimation of an undirected weighted consensus network in the presence of noise, using sparse inverse covariance estimation. This method involves utilizing the sample covariance matrix of the network's state to solve an optimization problem with a regularization term, aiming to achieve a sparse network topology. It's important to note that this research focuses on sparse network topology. In the case of sparse network topology (network with small number of edges), it has been demonstrated that exact topology identification is attainable for certain factors of the regularization term in the optimization problem. Conversely, for dense topology, the research achieves a sparse representation approximation of the network.

The study detailed in [31] focused on the identification of undirected, weighted consensus dynamics, with presence of noise. They adopt the Wiener filter methodology, which require time series measurements of the agents states. In the first phase, Wiener filtering is constructed by time series measurements of the agents states. This stage reveal network topology with potentially spurious edges. In the second phase, they eliminate these spurious edges through an examination of the frequency response of the Wiener filters. In the limit of large number of samples, this method allows them to accurately recover the network topology structure (i.e., the edges set, without necessarily the precise weights).

In [32], they deal with weighted network structure identification of discrete-time consensus dynamics. The problem consists of multiple independent consensus dynamics with the same network topology. In this context, "independent" signifies that each consensus dynamic possesses distinct initial conditions. By observing single snapshot of the dynamics at some common time T_i for each, and assuming some statistical properties of the initial conditions, they extract certain spectral properties of the network Laplacian matrix. Subsequently, they transform the problem into a convex optimization problem, with some desire properties like sparsity, to find the optimal estimate of the network. This method does not guarantee exact recovery of the structure of the network topology (i.e., the edges set of the graph).

In the study outlined in [33], the focus lies in network identification for diffusively-coupled networks. The authors present algorithms designed to address the challenge of network identification in the general, non-linear case, utilizing the global convergence properties of the network. They provide an error bound for the identified network in this general context and further demonstrate that their proposed algorithm is optimal with respect to time complexity.

Next, we will introduce research that involves partial observation methods. In [34], the focus is on identifying network topology from partial observations. The protocol employed is a consensus protocol with a known external input. Additionally, they assume knowledge regarding how the input interacts with the system, specifically identifying which agents directly receive the input signal. They utilize subspace techniques derived from system identification principles, grounded in the input-output interactions of the system. This leads to the formulation of an optimization problem aimed at identifying the network's topology. Numerical examples presented in the study demonstrate the exact recovery of the structure of the topology directly influenced by the input signal.

The study presented in [35] addresses the challenge of identifying the network topology within an undirected, unweighted consensus network that incorporates an external input. In this context, they make assumptions regarding knowledge of the input signal and its interaction with the network, specifically identifying which agents are directly affected by the input. Their approach relies on a method that employs the transfer function between input and output to identify network parameters. Moreover, they consider scenarios where the observable agents are also the observed agents. Notably, the study offers an algorithm capable of retrieving the network's structural topology efficiently within polynomial time complexity.

Now, let's introduce studies that delve into the utilization of the *Fiedler vector* within both autonomous and semi-autonomous consensus protocols. In the field of network structure analysis, many works rely on the spectral properties of the network, like examining the properties of the Fiedler vector. This vector is the eigenvector corresponding to the second smallest eigenvalue of a network's Laplacian matrix [36]. Under the context of consensus network protocols, the components of the Fiedler vector contain valuable information regarding the network's structure [36]. In semi-autonomous scenarios works such as [37] and [38], researchers explore the relationship between the second smallest eigenvalue and the corresponding eigenvector concerning graph structure. Their findings reveal structural properties analogous to Fiedler's classical results for the Laplacian: in particular, the eigenvector associated with the second smallest eigenvalue of the perturbed Laplacian exhibits monotonicity across cut vertices and induces sign–based partitions of the graph, features that are also fundamental in consensus scenarios. Additionally, studies such as [39] and [40] analyze the spectral properties of the grounded Laplacian matrix in consensus systems with stubborn agents. In particular, they show that the smallest eigenvalue of the grounded Laplacian determines the convergence rate to steady state, while the associated eigenvector reveals how the

influence of stubborn agents is distributed across the network. These works provide bounds on the smallest eigenvalue, characterize its dependence on graph structure, and establish connections to notions such as grounding centrality.

In [2], the authors investigate a scenario involving a single leader, a fixed input, and a semi-autonomous context. They didn't investigate the problem of leader identification directly, but rather the relationship between the components of the Fiedler vector and the manner in which the input signal interacts with the network. More precisely, they introduce the term *relative tempo* between two subgroups of agents, defined as the ratio of their state differentials which is equal to the corresponding division of the Fiedler vector components. By establishing this connection between the dynamic behavior of the system and the Fiedler vector components underlying the network graph, they assess various aspects of network performance. An intriguing finding from this study is that, in such scenarios characterized by a fixed input and a single leader, the leader agent corresponds to the smallest value in the Fiedler vector components.

1.2 Thesis Contribution

Our approach investigates the Fiedler vector components of the Laplacian in a semi-autonomous consensus scenario. As shown in [2], these components are related to the transient dynamics (before reaching consensus) in response to a constant external input sent to the leaders. In our research, we explore conditions, particularly in dense networks, that lead to a separation in the Fiedler vector components. This separation distinguishes between the leaders and followers, with leaders having the smallest Fiedler vector values and followers the largest. This division mirrors the transient dynamics of the agents, separating them into slow and fast dynamics—where the leaders exhibit slow dynamics and the followers fast dynamics. This separation enables leader identification by measuring the agents' velocity during the transient response to the external input. The main results can summarize as follow:

- We provide a sufficient condition for sufficiently large graphs to achieve separation of the Fiedler vector components into two distinct groups corresponding to the leaders and followers.
- We provide an estimation algorithm for the leader nodes by using only measurements of agent velocities.
- We provide some numerical studies to demonstrate our results.

1.3 Thesis Organization

This work is organized as follows. In Chapter 2 we introduce a basic brief of graph theory, and also presents the consensus protocol, for both autonomous and semi-autonomous cases. In Chapter 3 we deal with the leader identification problem of

semi-autonomous case, which include our main results, i.e, a theorem for leader identification. In Chapter 4 we give some simulation examples to demonstrate our finding. In Chapter 5 we conclude this work and raise some open questions for future work.

Chapter 2

Preliminaries

In this chapter, we provide an introduction to fundamental concepts in graph theory and present the consensus protocol for both autonomous and semi-autonomous scenarios.

2.1 Graph Theory

Graph theory offers a powerful way to represent systems made up of many interconnected components. Foundational works such as [41, 42] developed key concepts that remain central in understanding how the structure of a graph shapes the behavior of the system it models. These ideas have also influenced fields such as control and multi-agent systems, where graphs provide the basis for studying how agents exchange information and achieve coordinated behavior [1].

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by a set of vertices and edges. The *vertex* set denoted by $\mathcal{V}$, and has n elements, $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$. The *edge* set is denoted by $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The notation $e_{ij} = (v_i, v_j) \in \mathcal{E}$ corresponds to a directed edge from node $v_i \in \mathcal{V}$ to $v_j \in \mathcal{V}$, while $e_{ij} = \{v_i, v_j\}$ is an unordered pair and corresponds to an undirected edge from $v_i \in \mathcal{V}$ to $v_j \in \mathcal{V}$. The notation $i \sim j$ means that $e_{ij} \in \mathcal{E}$. In a *directed graph*, the edges are ordered, while in *undirected graphs*, the edges are unordered. An example of directed and undirected graphs is shown in Figure 2.1. In a *weighted graph*, weights are assigned to each edge through the map $\mathcal{W} : \mathcal{E} \to \mathbb{R}$, and the weighted graph is the triple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$. We use the notation w_{ij} to denote $\mathcal{W}(e_{ij})$.

For directed graphs, the *in-neighbours* of vertex v_i are defined by $\mathcal{N}^{in}(i) = \{v_j : e_{ji} \in \mathcal{E}\}$, and *out-neighbours* are $\mathcal{N}^{out}(i) = \{v_j : e_{ij} \in \mathcal{E}\}$. For undirected graphs, $\mathcal{N}^{in}(i) = \mathcal{N}^{out}(i)$, and we denote it simply by $\mathcal{N}(i)$. The *in-degree* of vertex v_i is defined by $d^{in}(i) = \sum_{e_{ji} \in \mathcal{E}} w_{ji}$, and *out-degree* is $d^{out}(i) = \sum_{e_{ij} \in \mathcal{E}} w_{ij}$. In case of undirected graph, $d^{in}(i) = d^{out}(i)$ and we denote it by $d(i)$.

A *walk*, Γ, is defined by an alternating sequence of vertices and edges. An edge between two sequential vertices in the sequence means there is an edge from the first

(a) An undirected graph. (b) A weighted directed graph.

Figure 2.1: An example of (a) an undirected graph and (b) a weighted directed graph.

vertex to the second. For example, $\Gamma_{ij} = v_i e_{ik_1} v_{k_1} e_{k_1 k_2} v_{k_2} ... e_{k_{m-1} j} v_j$ is a walk from node i to j. A *closed walk* is a walk where the initial and the final node is the same. The *length of a walk*, denoted by $\|\Gamma_{ij}\|$, is the total number of edges in the walk. The *weight of a walk* Γ_{ij} is defined as $W_{\Gamma_{ij}} = \Pi_{e_{ab} \in \Gamma_{ij}} w_{ab}$.

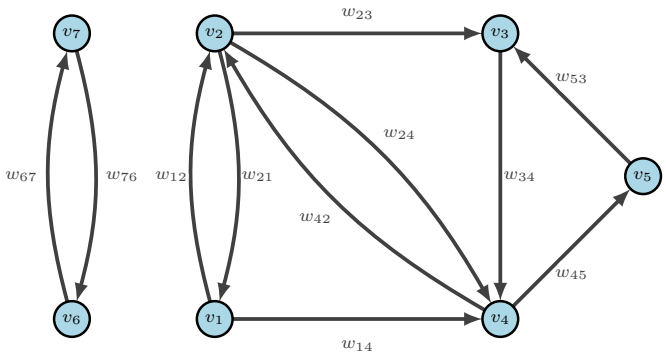


Figure 2.2: In the graph above, the vertices v_6 and v_7, along with their connecting edges, form a strongly connected subgraph. An example of a cycle is the sequence $v_4 e_{45} v_5 e_{53} v_3 e_{34} v_4$.

A *subgraph* is a graph which is contained in a larger graph. The graph is *strongly connected* if there is a walk between any two nodes in the graph. For undirected graphs, we just call it *connected*. A *connected component* is a connected subgraph of $\mathcal{G}$ that is not part of any larger connected subgraph. A *cycle* is a closed walk, where each edge is traversed only once. In Figure 2.2 the vertices $\{v_6, v_7\}$ and their incident edges form a strongly connected subgraph. Also, an example of a cycle is the closed walk $v_4 e_{45} v_5 e_{53} v_3 e_{34} v_4$. A strongly connected graph is *periodic* if the greatest common divisor of the lengths of all its simple cycles is larger than one. A digraph is *aperiodic* if it is not periodic. An example of an aperiodic graph is shown in Figure 2.3. Since there is a cycle with length 3 ($v_4 e_{45} v_5 e_{53} v_3 e_{34} v_4$) and a cycle with length 4 ($v_4 e_{41} v_1 e_{12} v_2 e_{23} v_3 e_{34} v_4$), the greatest common divisor of these lengths is 1, which means aperiodic graph.

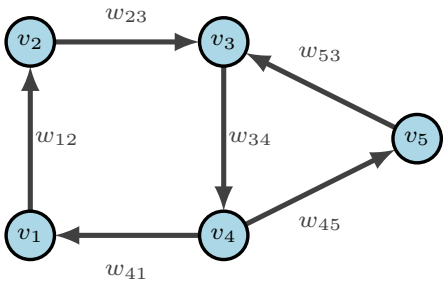


Figure 2.3: An example of aperiodic graph, where the greatest common divisor of its cycles lengths is one.

A graph can also be represented as a matrix, where the entries of the matrix represent the connections between vertices. The *adjacency matrix* of $\mathcal{G}$, denoted $A(\mathcal{G})$, is the $n \times n$ matrix, and it is defined by $[A(\mathcal{G})]_{ij} = w_{ij}$. The diagonal entries are typically set to zero, i.e., $[A(\mathcal{G})]_{ii} = 0$, unless self-loops are allowed, in which case $[A(\mathcal{G})]_{ii}$ equals the weight of the loop at vertex i. The *in-degree matrix* is defined by the $n \times n$ diagonal matrix $D^{in}(\mathcal{G})$, with $[D^{in}(\mathcal{G})]_{ii} = d_i^{in}$. The *out-degree matrix* has the same structure but with $[D^{out}(\mathcal{G})]_{ii} = d_i^{out}$. In case of undirected graph, we note the degree matrix as $D(\mathcal{G})$. The relation between adjacency matrix and degree matrix is $D(\mathcal{G}) = \mathrm{diag}(A(\mathcal{G})\mathbb{1})$. For directed graphs, the *in-Laplacian* and *out-Laplacian* matrices are defined by $L^{in}(\mathcal{G}) = D^{in}(\mathcal{G}) - A(\mathcal{G})$ and $L^{out}(\mathcal{G}) = D^{out}(\mathcal{G}) - A(\mathcal{G})$. For undirected graphs, $L^{in}(\mathcal{G}) = L^{out}(\mathcal{G})$ and we denote it by $L(\mathcal{G}) = D(\mathcal{G}) - A(\mathcal{G})$. Writing explicitly the Laplacian for undirected and unweighted graphs gives

$$[L(\mathcal{G})]_{ij} = \begin{cases} d(i), & i = j \\ -1, & i \sim j \\ 0, & \text{otherwise} \end{cases} . \tag{2.1}$$

For simplicity, we will now drop the notation $\mathcal{G}$ and use A, D and L when the context is clear. From this point forward, the notation $\mathcal{G}$ will refer to an undirected graph. When discussing directed graphs, we will introduce specific notation for each case. We now state a well known result for the Laplacian eigenvectors and eigenvalues.

Proposition 2.1.1 ([1])**.** *Let $\mathcal{G}$ be an undirected and connected graph. The Laplacian $L(\mathcal{G})$ has the following properties:*

- *$\lambda = 0$ is an eigenvalue with the corresponding eigenvector $v = \mathbb{1}$;*
- *The eigenvalues are real and can be ordered as follows: $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$.*

The term *grounded Laplacian* or *perturbed Laplacian* refers to a submatrix of L, noted as $L_B \in \mathbb{R}^{m \times m}, m < n$ after removing rows and their corresponding columns (the removed vertices are also called *grounded verteces* or *perturbed vertices*) [39].

Definition 2.1.2 (Grounded Laplacian)**.** Let $\mathcal{G}$ be an undirected graph with n nodes. Select $m < n$ nodes, referred to as the *grounded nodes*. Without loss of generality, assume that these grounded nodes correspond to the last m rows and columns of the Laplacian matrix $L(\mathcal{G})$. Partitioning the Laplacian as

$$L(\mathcal{G}) = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix}, \qquad L_{11} \in \mathbb{R}^{(n-m)\times(n-m)},$$

the matrix $L_B(\mathcal{G}) = L_{11}$ is called the *grounded Laplacian* of $L(\mathcal{G})$.

Like in the regular Laplacian case, the grounded Laplacian can be written as $L_B = D_{11} - A_{11}$ where $D_{11}, A_{11} \in R^{(n-m)\times(n-m)}$ are the corresponding upper-left block matrices of D and A.

We call the smallest non-zero eigenvalue of the grounded Laplacian matrix and the corresponding eigenvector as the *Fiedler eigenvalue* and *Fiedler vector*. We will now give another well known results that deals with the eigenvalues and eigenvectors of the grounded Laplacian.

Lemma 2.1.3 ([43])**.** *Let $\mathcal{G}$ be a connected graph with $n+1$ nodes, and let $L_B \in \mathbb{R}^{n\times n}$ be the grounded Laplacian after removing one row and corresponding column from $L(\mathcal{G})$. Let $\mathcal{G}' \subset \mathcal{G}$ be the subgraph obtained from $\mathcal{G}$ after removing the corresponding node and the connected edges from $\mathcal{G}$. If $\mathcal{G}'$ is connected, then L_B is non-singular and L_B^{-1} is a positive matrix.*

Corollary 2.1. *For the same conditions as Lemma 2.1.3, the Fiedler vector is unique (up to scaling) and is the only one that can be chosen to have positive values in all its entries.*

Proof. From Lemma 2.1.3 we know that L_B^{-1} is a positive matrix. From Perron-Frobenius Theorem [44] we know that there is a strictly largest eigenvalue (with absolute value) of L_B^{-1} and it is simple. Also, we know that the corresponding eigenvector is unique (up to scaling) and is the only one that can be chosen to have positive values in all its entries. Since L_B is a symmetric matrix, the relation between the eigenvalues of L_B and L_B^{-1} is

$$\lambda_i^{-1}(L_B) = \lambda_i(L_B^{-1}).$$

Hence, the Fiedler eigenvector of L_B (which is related to the smallest eigenvalue of L_B is unique (up to scaling) and is the only one that can be chosen to have positive values in all its entries. ■

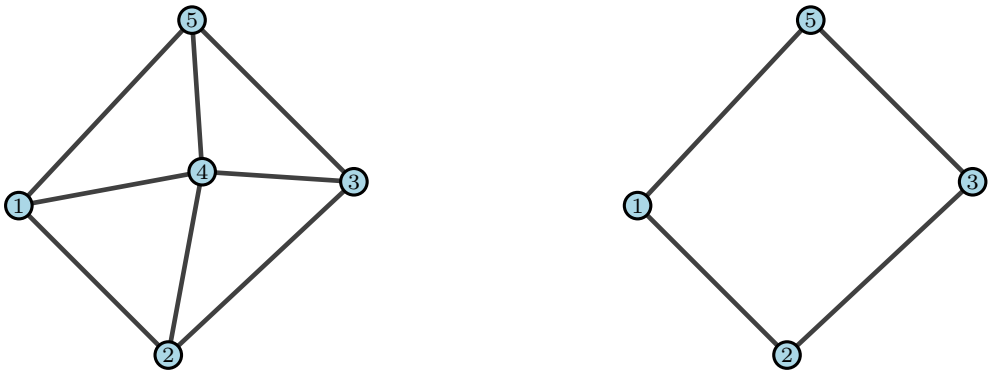
Proposition 2.1.4 ([39])**.** *For the same conditions as Lemma 2.1.3, the following properties hold for L_B:*

i) The smallest eigenvalue of L_B is positive and simple, i.e., $\lambda_1 > 0$.

ii) The upper bound of the smallest eigenvalue is $\lambda_1 \leq 1$.

iii) The upper bound for λ_1 *is achieved if and only if the removed row and column correspond to a node that is connected to all other nodes.*

An example of the case where the upper bound achieved is shown in Figure 2.4. For



(a) Graph $\mathcal{G}$ (b) Grounded graph $\mathcal{G}'$

Figure 2.4: Example graph where removing node 4 and its edges yields a grounded Laplacian with upper bound $\lambda_1 = 1$.

the graph in Figure 2.4(a), the *adjacency matrix* is

$$A(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix},$$

and the degree matrix is

$$D(\mathcal{G}) = \text{diag}(3,\, 3,\, 3,\, 4,\, 3).$$

The graph Laplacian is given by as

$$L(\mathcal{G}) = D(\mathcal{G}) - A(\mathcal{G}) = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & 0 & -1 & -1 & 3 \end{bmatrix}.$$

If we ground node 4 (remove the fourth row and column), the resulting grounded

Laplacian is

$$L_B(\mathcal{G}) = \begin{bmatrix} 3 & -1 & 0 & -1 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ -1 & 0 & -1 & 3 \end{bmatrix},$$

with rows and columns ordered as $(1, 2, 3, 5)$. The eigenvalues of L_B are $\{1, 3, 3, 5\}$. In particular, the smallest eigenvalue satisfies $\lambda_1(L_B) = 1$, which coincides with the upper bound for this case.

2.2 Consensus Protocol

Prior to introducing the protocols, we establish the groundwork for our problem. We are dealing with n dynamic agents, and we assume an integrator dynamics for each agent,

$$\dot{x}_i = u_i,\ i \in \{1, \dots, n\}, \tag{2.2}$$

where $x_i, u_i \in \mathbb{R}$ are the ith agent state and the corresponding input signal, respectively. The protocol can be distributed, meaning that each agent's control input depends on relative measurements taken from its neighbors. This can be represented by an underlying graph that describes the protocol governing how agents exchange information with one another. For example, the graph in Figure 2.1 represents a network where agents 1 and 2 exchange information, as do agents 2 and 3, and agents 3 and 4. In our study, we will deal with connected graphs.

The consensus protocol aims to achieve agreement among multiple agents regarding their states. By this we mean that the states of all agents converge to a common value, i.e., $x_i = x_j$ for all i and j.

Definition 2.2.1. Consider a group of n agents with individual states $x_i(t) \in \mathbb{R}$, collected in the vector $x(t) = [x_1(t), \dots, x_n(t)]^\top \in \mathbb{R}^n$. The agents are said to *reach agreement* if

$$\lim_{t\to\infty} |x_i(t) - x_j(t)| = 0, \quad \forall\, i, j \in \{1, \dots, n\}.$$

The notion of agreement defined above provides the foundation for the consensus problem in multi-agent systems. In essence, the goal is to design local interaction rules such that, through information exchange over the underlying communication network, all agents' states converge to a common value.

Problem 1. Let the agent dynamics be given by (2.2). Design a *distributed* control protocol

$$u_i(t) = \mathcal{F}_i\big(x_i(t), \{x_j(t)\}_{j\in\mathcal{N}(i)}\big), \qquad i \in \mathcal{V},$$

such that all states x_i reach agreement.

We will now introduce the consensus protocol, a now classic solution to Problem 1.

2.2.1 Autonomous Consensus Protocol

The unweighted, undirected consensus protocol is given by [1]:

$$\dot{x}_i = \sum_{i \sim j} (x_j - x_i),\ i \in \mathcal{V}, \tag{2.3}$$

where $x_i \in \mathbb{R}$ is the ith agent state. The protocol can be written in a matrix form as

$$\dot{x} = -Lx, \tag{2.4}$$

where x is the state vector, and L is the Laplacian matrix of the underlying graph. The solution of (2.4) is given by

$$x(t) = \mathrm{e}^{-\lambda_1 t}(v_1^T x_0)v_1 + \mathrm{e}^{-\lambda_2 t}(v_2^T x_0)v_2 + \cdots + \mathrm{e}^{-\lambda_n t}(v_n^T x_0)v_n$$

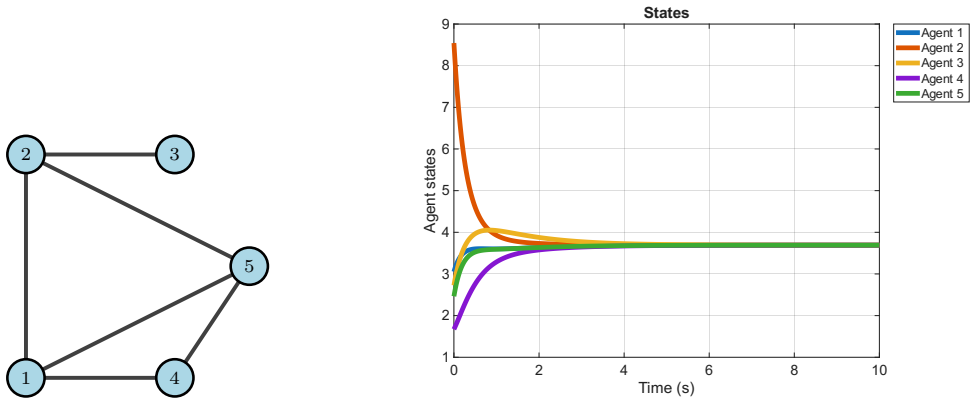
where v_i are the eigenvectors corresponding to the eigenvalues λ_i of L, and x_0 is the initial state vector.

The steady-state value of the consensus protocol (2.4) can be computed as

$$\lim_{t \to \infty} x(t) = \frac{\mathbb{1}^T x_0}{n} \mathbb{1}. \tag{2.5}$$

The states converge to a consensus value equal to the average of the initial states. The rate of convergence is determined by the smallest non-zero eigenvalue, λ_2 [1].

The consensus protocol is illustrated for five agents with initial states $x(0) = [3.0530, 8.5435, 2.7246, 1.6712, 2.4633]^\top$. Figure 2.5 shows the evolution of the state and the underlying graph. All states converge to the average of the initial conditions, which equals to 3.69.



(a) Underlying graph.

(b) Agent trajectories.

Figure 2.5: Consensus simulation of five agents. The states converge to the average of the initial values.

2.2.2 Semi-Autonomous Consensus Protocol

In the semi-autonomous protocol, the nodes are divided into 2 groups: leaders and followers, denoted by the sets $\mathcal{V}_\ell$ and $\mathcal{V}_f$ respectively, such that $\mathcal{V}_\ell \cup \mathcal{V}_f = \mathcal{V}$. The leader nodes receive an external input in addition to information from neighboring agents in the network. In this work, we assume these external inputs are constant signals. We at times refer the leader nodes in this scenario (constant input) as *grounded nodes*. Without loss of generality, we assume the nodes of $\mathcal{G}$ are ordered such that the first $|\mathcal{V}_\ell|$ nodes are leaders, and the rest are followers.

Assumption 2.2.2. The leaders and followers sets are both nonempty.

The dynamics for both leader and follower agents follow the same model as described in (2.2), with the leader nodes receiving an additional external signal,

$$\dot{x}_i = \begin{cases} u_i + u_i^{ex}, & i \in \mathcal{V}_\ell \\ u_i, & i \in \mathcal{V}_f \end{cases}. \tag{2.6}$$

The semi-autonomous consensus protocol is then given by

$$\dot{x}_i = \begin{cases} \sum_{i\sim j}(x_j - x_i) + (u_i^{ex} - x_i), & i \in \mathcal{V}_\ell \\ \sum_{i\sim j}(x_j - x_i), & i \in \mathcal{V}_f \end{cases}. \tag{2.7}$$

where $x_i \in \mathbb{R}$ is the ith agent state. The matrix form of this protocol is given by:

$$\dot{x} = -L(\mathcal{G})x + \begin{bmatrix} -I_{|\mathcal{V}_\ell|} & 0_{|\mathcal{V}_\ell|\times|\mathcal{V}_f|} \\ 0_{|\mathcal{V}_f|\times|\mathcal{V}_\ell|} & 0_{|\mathcal{V}_f|\times|\mathcal{V}_f|} \end{bmatrix} x + \begin{bmatrix} I_{|\mathcal{V}_\ell|} \\ 0_{|\mathcal{V}_f|\times|\mathcal{V}_\ell|} \end{bmatrix} \begin{bmatrix} u_1^{ex} \\ \vdots \\ u_{|\mathcal{V}_\ell|}^{ex} \end{bmatrix}. \tag{2.8}$$

We now define an augmented network $\bar{\mathcal{G}}$, in which the external control inputs are modeled as additional agents in the network, denoted as y_i. Each control agent y_i is added to the original network $\mathcal{G}$ and connected with a directed edge to the corresponding node that previously received an external input signal. Specifically, if a node x_i in $\mathcal{G}$ receives an external input u_i^{ex}, then in the augmented network $\bar{\mathcal{G}}$, there is a directed edge from y_i to x_i. Since we assume constant external inputs u_i^{ex}, this implies the following dynamics for the control agents y_i,

$$\dot{y}_i = 0 \quad \text{and} \quad y_i(0) = u_i^{ex}.$$

The augmented network, therefore, consists of $|\mathcal{V}| = n$ original agents along with $|\mathcal{V}_\ell|$ additional control agents. The augmented state is represented as: $\bar{x} = \begin{bmatrix} x^T & y^T \end{bmatrix}^T$. An example of augmented graph with two leader nodes is shown in Figure 2.6. To

express this protocol in terms of x and y, we write:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = -\begin{bmatrix}\bar{L}_{11} & \bar{L}_{12}\\ 0 & 0\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}. \tag{2.9}$$

where $\bar{L} = L(\bar{\mathcal{G}}) = \begin{bmatrix}\bar{L}_{11} & \bar{L}_{12}\\ 0 & 0\end{bmatrix}$ is the directed Laplacian of the directed graph $\bar{\mathcal{G}}$.

The explicit forms of $\bar{L}_{11}$ and $\bar{L}_{12}$ are given as follows:

$$\bar{L}_{11} = L(\mathcal{G}) + \begin{bmatrix} I_{|\mathcal{V}_\ell|} & 0_{|\mathcal{V}_\ell|\times|\mathcal{V}_f|}\\ 0_{|\mathcal{V}_f|\times|\mathcal{V}_\ell|} & 0_{|\mathcal{V}_f|\times|\mathcal{V}_f|}.\end{bmatrix} \text{ and } \bar{L}_{12} = \begin{bmatrix} -I_{|\mathcal{V}_\ell|}\\ 0_{|\mathcal{V}_f|\times|\mathcal{V}_\ell|}\end{bmatrix}. \tag{2.10}$$

Now we will show that $\bar{L}_{11}$ corresponds to the grounded Laplacian of some graph. Define a third graph, $\bar{\mathcal{G}}'$, which is the undirected version graph of $\bar{\mathcal{G}}$. This graph has the same Laplacian components $\bar{L}_{11}$ and $\bar{L}_{12}$ but in a symmetric structure.

$$L(\bar{\mathcal{G}}') = \begin{bmatrix}\bar{L}_{11} & \bar{L}_{12}\\ \bar{L}_{12}^T & \text{diag}(\bar{L}_{12}^T\mathbb{1})\end{bmatrix}. \tag{2.11}$$

We note that the submatrix $\bar{L}_{11} \in \mathbb{R}^{n\times n}$ is the grounded Laplacian L_B of $L(\bar{\mathcal{G}}')$ obtained by removing the last m rows and columns. So, from (2.10) we have,

$$L_B = \bar{L}_{11} = L(\mathcal{G}) + \begin{bmatrix} I_{|\mathcal{V}_\ell|} & 0_{|\mathcal{V}_\ell|\times|\mathcal{V}_f|}\\ 0_{|\mathcal{V}_f|\times|\mathcal{V}_\ell|} & 0_{|\mathcal{V}_f|\times|\mathcal{V}_f|}\end{bmatrix}. \tag{2.12}$$

Here, $L(\mathcal{G})$ is the regular Laplacian matrix of $\mathcal{G}$.

We can write the grounded Laplacian components in terms of the node degree in $\mathcal{G}$ using the relation $L_B = D_{11} - A_{11}$, described in Chapter 2.1. Here, D_{11}, A_{11} are the corresponding upper left submatrtix of the degree and adjacency matrices of $\bar{\mathcal{G}}'$:

$$[L_B]_{ij} = \begin{cases} d(i)+1 & i=j, i\in\mathcal{V}_\ell\\ d(i) & i=j, i\in\mathcal{V}_f\\ -1 & i\sim j\\ 0 & \text{otherwise}\end{cases}. \tag{2.13}$$

Next, we will present algebraic results for the eigenvectors and eigenvalues of the grounded Laplacian L_B and for the directed Laplacian matrix $\bar{L}$. These results will later be utilized to analyze the dynamic response in the relative tempo section.

From this point onward, we will refer to the smallest non-zero eigenvalue of L_B as λ_F (the Fiedler eigenvalue), and the corresponding eigenvector as v_F (the Fiedler eigenvector).

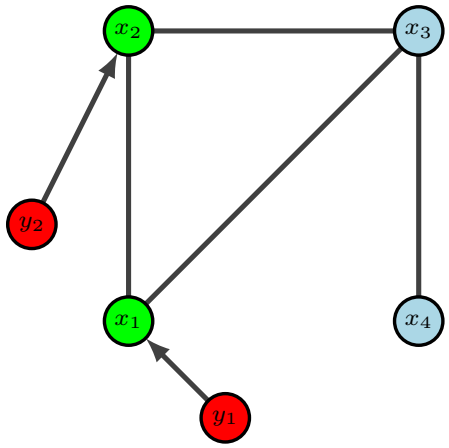


Figure 2.6: An augmented graph is shown: nodes $(x_1, \ldots, x_4)$ form the original subgraph. Red nodes are added agents, green nodes are leaders connected to external agents by directed edges, and blue nodes are followers. Original edges remain undirected.

Proof. In our setup, the grounded Laplacian matrix L_B is results of removing rows an columns corresponding to the added nodes y (recall (2.11)). Hence, after removing nodes y we just left with our original graph $\mathcal{G}$ which is connected (as outlined in Section 2.2). This satisfies the conditions stated in Corollary 2.1 and Proposition 2.1.4 which implies our results. ■

Proposition 2.2.3. *For the setup described in Section 2.2.2, the Fiedler eigenvalue of L_B (defined in (2.12)) is simple and strictly less than 1, i.e.:*

$$0 < \lambda_F < 1.$$

Also, v_F is unique up to a scalar multiple and can be chosen to have all positive entries.

Proof. We will show that L_B can be formed by removing a single row and column from some other Laplacian, then we utilize Proposition 2.1.4 to get our results. Let's define the following matrix,

$$L = \begin{bmatrix} \bar{L}_{11} & \begin{bmatrix} -\mathbb{1}_{|\mathcal{V}_\ell| \times 1} \\ 0_{|\mathcal{V}_f| \times 1} \end{bmatrix} \\ \begin{bmatrix} -\mathbb{1}_{|\mathcal{V}_\ell| \times 1} & 0_{|\mathcal{V}_f| \times 1} \end{bmatrix} & |\mathcal{V}_\ell| \end{bmatrix}, \tag{2.14}$$

where $\bar{L}_{11}$ defined from (2.10). This matrix represents a Laplacian matrix of some undirected graph $\mathcal{G}''$ where it combined from our original graph $\mathcal{G}$ and additional node s' that is connected to all leaders. From our Assumption 2.2.2, not all nodes receive an external signal, i.e., not all nodes are leaders, which imply that s' is not connected to all other nodes in $\mathcal{G}''$. Corollary 2.1 and Proposition 2.1.4 implies our results. ■

Proposition 2.2.4. *For the setup described in Section 2.2.2, The matrix L_B (defined in (2.12)) is non-singular*

Proof. We can show that the grounded Laplacian L_B can be obtained by removing a single row and column from another Laplacian, similarly to the procedure in the proof of Proposition 2.2.3. Hence, the matrix L_B is non-singular [43]. ■

We can now discuss the solution to (2.9), which can be expressed as

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} e^{-\bar{L}_{11}t} & -\int_0^t e^{-\bar{L}_{11}(t-\tau)}\bar{L}_{12}\,d\tau \\ 0 & I \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}.$$

Since $\bar{L}_{11}$ is non-singular (from Proposition 2.2.4) we can write

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} e^{-\bar{L}_{11}t} & -\bar{L}_{11}^{-1}(I - e^{-\bar{L}_{11}t})\bar{L}_{12} \\ 0 & I \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix},$$

In the limit, we have

$$\lim_{t\to\infty} x(t) = -\bar{L}_{11}^{-1}\bar{L}_{12}y_0,$$

and $y(t)$ remains constant.

Proposition 2.2.5. *Let $\{\lambda_i(\bar{L}), v_i(\bar{L})\}$, $\{\lambda_i(L_B), v_i(L_B)\}$ be the eigenvalues and corresponding eigenvectors of $\bar{L}$ and L_B respectively. Then,*

$$\lambda_i(\bar{L}) = \begin{cases} 0, & i = 1, \ldots, m \\ \lambda_{i-m}(L_B), & i = m+1, \cdots, n+m \end{cases}. \tag{2.15}$$

and

$$v_i(\bar{L}) = \begin{bmatrix} v_i(L_B)^T & \mathbf{0}_m^T \end{bmatrix}^T, \quad i = m+1, \cdots, m+n. \tag{2.16}$$

Proof. The matrix $\bar{L}$ is a square matrix with last m rows equal to zero, therefore the eigenvalue $\lambda = 0$ has algebraic multiplicity of m. The last n eigenvalues and eigenvectors is achieved by straight-forward calculation,

$$\bar{L}v_i(\bar{L}) = \begin{bmatrix} L_B v_i(L_B) \\ 0 \end{bmatrix} = \begin{bmatrix} \lambda_i(L_B)v_i(L_B) \\ 0 \end{bmatrix} = \lambda_i(L_B)v_i(\bar{L}).$$

So, we got that $\lambda_i(\bar{L}), v_i(\bar{L}), i = m+1, \cdots, m+n$, are the eigenvalues and the corresponding eigenvectors of $\bar{L}$. ■

Corollary 2.2. *The smallest non-zero eigenvalue of $\bar{L}$ is simple and positive.*

Proof. In Proposition 2.2.5 we showed that the all non-zero eigenvalues of $\bar{L}$ are the same as L_B. Since the smallest non-zero of L_B is positive and simple (from Proposition 2.2.3), then this is true also for $\bar{L}$. ■

Also, we have m eigenvector corresponding to the zero eigenvalues. It easy to observe that one of them is the all one vector: $\bar{L}\mathbb{1} = 0$. It is unnecessary to bring here the structure of the rest $m-1$ eigenvectors corresponding to the zero eigenvalues. Nevertheless, we will bring immediately a proposition that deals with the linear independence of all the eigenvectors of $\bar{L}$.

Proposition 2.2.6. *The directed Laplacian matrix, $\bar{L}$, has $n+m$ linearly independent eigenvectors.*

Proof. The last n eigenvectors of $\bar{L}$ are the same as the symmetric matrix L_B, augmented with zeros as shown in Proposition 2.2.5. Therefore, there are n linearly independent eigenvectors corresponding to the last n eigenvalues.

The first m eigenvectors corresponding to the eigenvalue zero with algebraic multiplicity of m (from Proposition 2.2.5). Since L_B is an $n \times n$ full-rank matrix (no zero eigenvalue), the rank of $\bar{L}$ is n. Hence, from the rank-nullity theorem in linear algebra, for any matrix $\bar{L}$ of size $(n+m) \times (n+m)$ we have $\dim(\ker(L)) + \operatorname{rank}(L) = n+m$. The geometric multiplicity of the zero eigenvalue of $\bar{L}$ is defined as the dimension of its nullspace, i.e., $GM(0) = \dim(\ker(L))$. Since $\operatorname{rank}(\bar{L}) = n$, then the rank-nullity theorem gives $GM(0) = \dim(\ker(\bar{L})) = (n+m) - \operatorname{rank}(\bar{L}) = (n+m) - n = m$. Since the algebraic and geometric multiplicities of zero eigenvalues are equal to m, we have m independent eigenvectors corresponding to zero eigenvalue.

We therefore have n linearly independent eigenvectors corresponding to the strictly negative eigenvalues, and m linearly independent eigenvectors corresponding to the zero eigenvalues. Due to the fact that eigenvectors corresponding to different distinct eigenvalues are linearly independent, we conclude that there are $n+m$ linearly independent eigenvectors. ■

As shown in Proposition 2.2.6, the matrix $\bar{L}$ has $n+m$ linearly independent eigenvectors, so we can write the eigen-decomposition of $\bar{L}$ as $\bar{L} = V\Lambda V^{-1}$, where V is a matrix whose ith column is the eigenvector v_i of $\bar{L}$. Hence, the solution of (2.9) using eigenvalues can be written as follows:

$$\bar{x}(t) = \mathrm{e}^{-\lambda_1(\mathcal{G})t}(q_1\bar{x}_0)v_1 + \mathrm{e}^{-\lambda_2(\mathcal{G})t}(q_2\bar{x}_0)v_2 + \cdots + \mathrm{e}^{-\lambda_{n+m}(\mathcal{G})t}(q_{n+m}\bar{x}_0)v_{n+m} \qquad (2.17)$$

where q_i is the ith row of V^{-1}, and v_i is the ith column of V. We remind that the external control inputs are constant, so the $u_i(t)$ part of $\bar{x}(t)$ is just $u_i(t) = u_i^{ex}$.

2.2.3 Relative Tempo

Next, we will introduce the concept of the *relative tempo*, which essentially captures the ratio of state derivatives between two agents. The intention behind introducing this term is to establish a connection between real-time measurable values, the state derivative of the agents, and the Fiedler vector, which is related to the graph's spectral

properties. We will now explicitly illustrate this relationship. Mathematically, the relative tempo between any two arbitrary agents can be defined as [2],

$$\tau_{ij} = \frac{\dot{x}_i}{\dot{x}_j}. \tag{2.18}$$

Here, $\dot{x}_i$ is the velocity of node i. To establish the relationship between the relative tempo and the Fiedler vector, we introduce the *selector vector* $c_k \in \mathbb{R}^{n+m}$, defined component-wise as

$$[c_k]_i = \begin{cases} 1, & i = k \\ 0, & \text{otherwise} \end{cases}. \tag{2.19}$$

By substituting the derivative of the solution from (2.17) into the expression for the relative tempo, we arrive at the following results:

$$\begin{aligned} \tau_{ij} &= \frac{\dot{\bar{x}}_i}{\dot{\bar{x}}_j} \\ &= \frac{\sum_{k=1}^{m} \lambda_k c_i^T (q_k \bar{x}_0) v_k e^{-\lambda_k t} + \lambda_F c_i^T (q_{m+1} \bar{x}_0) v_F e^{-\lambda_F} + \sum_{k=m+2}^{m+n} \lambda_k c_i^T (q_k \bar{x}_0) v_k e^{-\lambda_k t}}{\sum_{k=1}^{m} \lambda_k c_j^T (q_k \bar{x}_0) v_k e^{-\lambda_k t} + \lambda_F c_j^T (q_{m+1} \bar{x}_0) v_F e^{-\lambda_F} + \sum_{k=m+2}^{m+n} \lambda_k c_j^T (q_k \bar{x}_0) v_k e^{-\lambda_k t}}, \\ & i, j \in \{1, \ldots, n+m\}. \end{aligned} \tag{2.20}$$

Here, we substitute the derivative of the states by taking the derivative of (2.17). The first m eigenvalues are just zero, then we have the Fiedler eigenvalue which is the largest and simple (negative value). Then we have the remaining $n-1$ negative eigenvalues. We can now utilize our knowledge about the eigenvalues and eigenvectors of $\bar{L}$ to give an expression in term of Fiedler vector v_F.

The expression in (2.17) is defined in an $(n+m)$-dimensional space, corresponding to the dimension of $\bar{L}$. However, when expressing the relative tempo in terms of the Fiedler vector components $[v_F]_i$, where v_F belong to the n-dimensional subspace associated with L_B, the indices i must be restricted accordingly. This simplification is valid because the remaining m eigenvalues of $\bar{L}$ are zero, implying that the corresponding states y (recall that $\bar{x} = \begin{bmatrix} x^T & y^T \end{bmatrix}^T$) have constant derivatives, i.e., $\dot{y} = 0$.

In this direction we have,

$$\tau_{ij} = \frac{\lambda_F (p_{m+1}^T \bar{x}_0)[v_F]_i e^{-\lambda_F t} + \sum_{k=m+2}^{m+n} \lambda_k (p_k^T \bar{x}_0)[v_k]_i e^{-\lambda_k t}}{\lambda_F (p_{m+1}^T \bar{x}_0)[v_F]_j e^{-\lambda_F t} + \sum_{k=m+2}^{m+n} \lambda_k (p_k^T \bar{x}_0)[v_k]_j e^{-\lambda_k t}}, \quad i, j \in \{1, \ldots, n\}. \tag{2.21}$$

The movement from (2.20) to (2.21) is by eliminate the zero eigenvalue expression. Also, because the largest non-zero eigenvalue, λ_F, is non-negative and unique, as shown in Proposition 2.2.3, for a sufficiently long time period T, we can approximate the

relative tempo as follows,

$$\tau_{ij} \approx \frac{[v_F]_i}{[v_F]_j}, \quad t > T, \quad i, j \in [1, \ldots, n] \tag{2.22}$$

It is important to note that once consensus is reached, the relative tempo becomes undefined, as both the numerator and denominator approach zero. However, the analysis is performed at a sufficiently large time T, before consensus occurs, where the relative dynamics are still meaningful. This is possible due to the spectral gap between the Fiedler eigenvalue λ_F and the next largest nonzero eigenvalue.

Using (2.22), the Fiedler vector (up to a scaling factor) can be estimated by selecting one agent as a reference and normalizing the other agents' velocity ratios with respect to it. Specifically, by choosing an agent j^* as the reference, the Fiedler vector can be expressed as

$$v_F \in \text{span}\{\bar{\tau}\}, \tag{2.23}$$

where each component is defined as $[\bar{\tau}]_i = \tau_{ij^*}$, and the relation holds for $t > T$.

To extend the semi-autonomous consensus protocol to two dimensions, each agent's state is now represented by a vector $x_i = [x_{i,1},\, x_{i,2}]^\top \in \mathbb{R}^2$, where $x_{i,1}$ and $x_{i,2}$ denote its coordinates in the 2D space. The interaction structure and control logic remain identical to the one-dimensional case; that is, each agent updates its position based on the relative differences with its neighbors, while the leader agents additionally track their external reference inputs. The resulting dynamics are obtained by applying the same protocol component-wise to each coordinate, leading to a two-dimensional semi-autonomous consensus system that can be compactly written in matrix form as

$$\dot{x} = -(\bar{L} \otimes I_2)x + \left(\begin{bmatrix} I_{|\mathcal{V}_\ell|} \\ 0_{|\mathcal{V}_f| \times |\mathcal{V}_\ell|} \end{bmatrix} \otimes I_2 \right) u^{ex}, \tag{2.24}$$

where $x = [x_1^\top,\, x_2^\top,\, \ldots,\, x_n^\top]^\top \in \mathbb{R}^{2(n+m)}$ is the stacked state vector of all agents, $\bar{L}$ is the augmented Laplacian matrix corresponding to the interaction topology, I_2 is the 2×2 identity matrix, and u^{ex} is the vector of external control inputs applied to the leader agents. This formulation allows the consensus behavior in both spatial dimensions to evolve simultaneously under the same communication graph.

Example 2.2.7. The following 2D example illustrate the relation between relative tempo and Fiedler vector. Let define a graph (Figure 2.7) with $n = 6$ nodes, where nodes 1

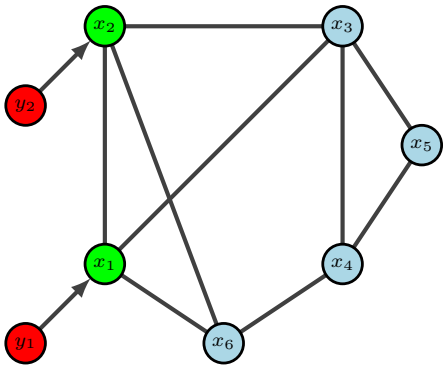


Figure 2.7: An example of augmented graph that represents a network with 2 leaders (green), 2 constant inputs (red), and 4 followers (blue).

and 2 are the leaders. The external input to the leaders is $u_i^{ex} = [20, 20]^T$ for $i = 1, 2$.

$$\bar{L} = \begin{bmatrix} 4 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & 4 & -1 & 0 & 0 & -1 & 0 & -1 \\ -1 & -1 & 4 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 2 & 0 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The matrix L_B is just the upper left 6×6 submatrix of $\bar{L}$. The Fiedler vector of $\bar{L}$, normalized by the first entry of the vector, is given as,

$$v_F = \begin{bmatrix} 0.33 & 0.29 & 0.48 & 0.41 & 0.51 & 0.37 \end{bmatrix}^T.$$

The protocol was simulated in Matlab for $t = 20$ seconds and involves 2D dynamics. In the figures below, the green curve related to leader agents, and blue related to follower agents. The trajectories of the agents are depicted in Figure 2.8, while the relative tempo is shown in Figure 2.9. The relative tempo is calculated using (2.2.3), with the velocity of the first agent in the denominator. It can be observed that agreement is achieved with the x, y coordinates of value 20. The relative tempo of the transient dynamics reaches a constant value in less than 20 seconds, indicating convergence to the corresponding components of the Fiedler vector.

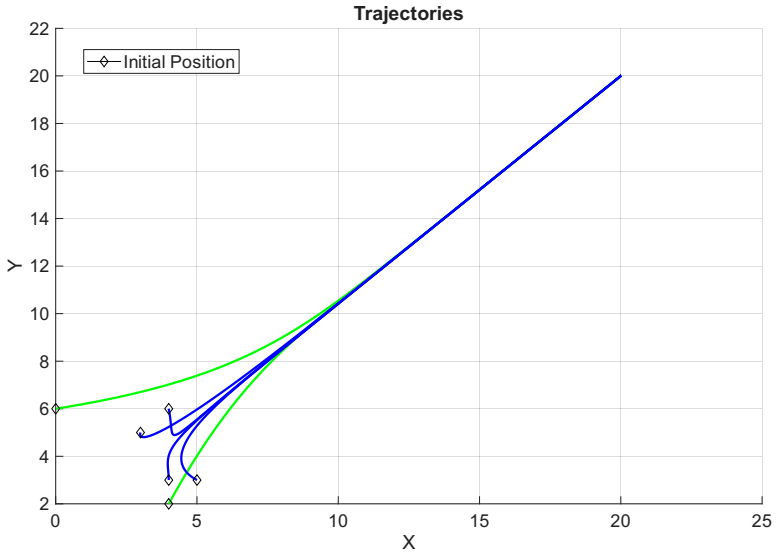


Figure 2.8: State values of the agents. The last 2 agents (7,8) are the constant external input nodes. The green and blue lines correspond to the leader and follower agents, respectively.

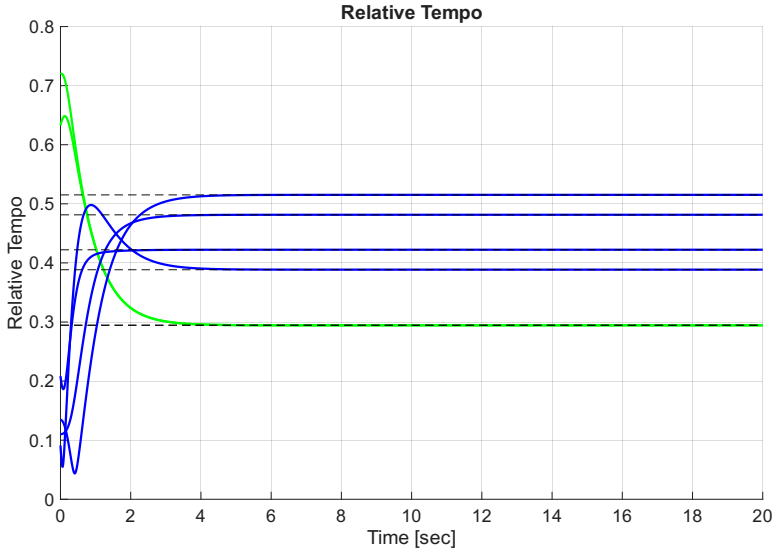


Figure 2.9: Relative tempo values for each agent. The dashed black lines indicate the individual components of the Fiedler vector. The green and blue lines correspond to the leader and follower agents, respectively.

Chapter 3

Leader Identification in Semi-Autonomous Consensus Protocol

In this chapter, we will delve into the problem of leader identification, which encompasses our primary results. We will provide theoretical analysis and present simulation examples.

3.1 Problem Setup

Consider the setup of the semi-autonomous protocol given in Section 2.2.2. Each leader receives an unknown external constant input signal. The dynamics of the system are observed from an unknown initial state. In mathematical terms, we have recorded the state of nodes, denoted as $x(t_i)$, at time interval $[t_0, t_f]$, where $x(t_i)$ corresponds to time t_i. Our objective is to identify the leader nodes within this network.

Problem 2. Consider a network operating under a semi-autonomous consensus protocol, where both the initial states and the external input signals are unknown. Suppose that the system states $x(t)$ are observed within the interval $[t_0, t_f]$. The objective is to identify the leader nodes in the network.

In our next step, we will examine the relation between the Fiedler vector v_F and the original graph structure $\mathcal{G}$. More specific, we will define a normalized adjacency matrix, and then we will examine the behavior of the corresponding eigenvectors and its relation to the Fiedler vector.

3.2 Fiedler Vector of the Semi-Autonomous Consensus protocol Laplacian

In this section, we will delve into the relationship between the Fiedler vector v_F (recall that in our work v_F is related to the Fiedler vector of L_B) and the structure of our original graph $\mathcal{G}$. Our methodology is as follow: (i) in previous chapter we define the augmented graph $\bar{\mathcal{G}}$ to develop the connection between the relative tempo (discussed in Section 2.2.3) and the Fiedler vector v_F, and (ii) now we want to define another graph which will help us to examine the relation between v_F and the original graph structure $\mathcal{G}$. Finally we will combine (i) and (ii) to a theorem that gives sufficient conditions on the structure of $\mathcal{G}$ to identify the leaders based on the relative tempo (which can be measured).

To start, we will express the Fiedler vector components through walks on a normalized graph, a concept we will define shortly. Specifically, we define the *semi-normalized adjacency matrix* as

$$\hat{A} = \hat{D}_{\lambda_F}^{-1} A \in \mathbb{R}^{n\times n}, \tag{3.1}$$

where $\hat{D}_{\lambda_F} \in \mathbb{R}^{n\times n}$ is given by

$$[\hat{D}_{\lambda_F}]_{ii} = \begin{cases} d(i) + 1 - \lambda_F & i \in \mathcal{V}_\ell \\ d(i) - \lambda_F & i \in \mathcal{V}_f \end{cases}. \tag{3.2}$$

We recall that $d(i)$ is the degree of node i in $\mathcal{G}$. The invertibility of the matrix $\hat{D}_{\lambda_F}$ follows from our knowledge, as stated in Proposition 2.2.3, that the Fiedler eigenvalue satisfies $0 < \lambda_F < 1$. The matrix $\hat{A}$ is diagonalizable since it is similar to the symmetric matrix $\hat{D}_{\lambda_F}^{-1/2} A \hat{D}_{\lambda_F}^{-1/2}$, as

$$\hat{A} = \hat{D}_{\lambda_F}^{-1/2} (\hat{D}_{\lambda_F}^{-1/2} A \hat{D}_{\lambda_F}^{-1/2}) \hat{D}_{\lambda_F}^{1/2}. \tag{3.3}$$

Proposition 3.2.1. *The symmetric matrix $\hat{D}_{\lambda_F}^{-1/2} A \hat{D}_{\lambda_F}^{-1/2} \in \mathbb{R}^{n\times n}$ shares the same eigenvalues as $\hat{A}$.*

Proof. From (3.3) we know that both matrices are similar, and similar matrices share the same eigenvalues, which imply the proposition statement. ■

Based on the normalized adjacency matrix $\hat{A}$, we can define a weighted graph, denoted as $\hat{\mathcal{G}}$. The graph $\hat{\mathcal{G}}$ contains the same vertices and edges as $\mathcal{G}$, but with weights on the edges.

In the following, we present a series of lemmas and corollaries that establish a connection between weighted walks of length k in the graph $\hat{\mathcal{G}}$ and the components of the Fiedler eigenvector v_F. In the next chapter, we will focus on identifying graphs for which length-1 walks alone suffice to provide a good approximation of the Fiedler vector, leading to simple and analytically tractable expressions for its components.

The following lemma establishes a connection between a walk on the graph $\hat{\mathcal{G}}$ and the corresponding entry in a power of its adjacency matrix.

Proposition 3.2.2. *If $\mathcal{G}$ is connected, then $\hat{\mathcal{G}}$ is strongly connected.*

Proof. The matrix $\hat{A}$ is derived by normalizing the rows of A by some positive numbers in $\hat{D}_{\lambda_F}^{-1}$. Therefore, $\hat{A}$ has the same pattern as A (i.e., non-zero entries of A correspond to non-zero entries of $\hat{A}$, and likewise for zero-entries). In particular, if there is an undirected edge between nodes i and j in $\mathcal{G}$, then there are directed edges (i,j) and (j,i) (with potentially different weights) in $\hat{\mathcal{G}}$. Therefore $\hat{\mathcal{G}}$ is strongly connected whenever $\mathcal{G}$ is connected. ■

Now, we will introduce lemmas that establish a connection between the Fiedler vector and graph walks. These lemmas will form the basis for stating and proving our main theorem in the upcoming sections.

Lemma 3.2.3 ([45])**.** *The entry $[\hat{A}^k]_{ij}$ is equal to the sum of weights of all walks with length k from node i to j, i.e.,*

$$[\hat{A}^k]_{i,j} = \sum_{\|\Gamma_{ij}\|=k} W_{\Gamma_{ij}}. \tag{3.4}$$

Next, we present two lemmas that show the largest eigenvalue of $\hat{A}$ is 1, with the corresponding eigenvector v_F. These results will later be used to examine the limiting value as the walk length approaches infinity.

Lemma 3.2.4. *The semi-normalized adjacency matrix $\hat{A}$ has real, simple, greatest eigenvalue equal to* 1*. Furthermore, the corresponding eigenvector is v_F (recall that v_F is the corresponding Fiedler vector of L_B).*

Proof. Define the following diagonal matrix,

$$[\hat{D}_0]_{ii} = \begin{cases} d(i)+1 & i \in \mathcal{V}_\ell \\ d(i) & i \in \mathcal{V}_f \end{cases}.$$

First, we will show that v_F is an eigenvector with corresponding eigenvalue $\lambda = 1$. Based on the definition of L_B, we can write the following relation (recall the components expression for L_B from (2.13)),

$$L_B = \hat{D}_0 - A.$$

Subtract $\lambda_F I$ from both sides and multiply both side by $(\hat{D}_0 - \lambda_F I)^{-1}$ from left to obtain

$$(\hat{D}_0 - \lambda_F I)^{-1}(L_B - \lambda_F I) = (\hat{D}_0 - \lambda_F I)^{-1}(\hat{D}_0 - \lambda_F I) - (\hat{D}_0 - \lambda_F I)^{-1}A.$$

Rearranging above gives

$$(\hat{D}_0 - \lambda_F I)^{-1} A = I - (\hat{D}_0 - \lambda_F I)^{-1}(L_B - \lambda_F I).$$

Note that $\hat{D}_0 - \lambda_F I = \hat{D}_{\lambda_F}$ defined in (3.2), giving

$$\hat{A} = I - (\hat{D}_{\lambda_F})^{-1}(L_B - \lambda_F I).$$

By multiply both sides by v_F we get:

$$\hat{A}v_F = v_F - (\hat{D}_{\lambda_F})^{-1}(L_B - \lambda_F I)v_F = v_F - (\hat{D}_{\lambda_F})^{-1}(\lambda_F v_F - \lambda_F v_F) = v_F.$$

Therefore, $\lambda = 1$ is an eigenvalue of $\hat{A}$ with corresponding eigenvector v_F.

We now show that $\lambda = 1$ is simple and is also the largest eigenvalue.

The matrix $\hat{A}$ is non-negative, and the corresponding graph is strongly connected which follows from Proposition 3.2.2. From the Perron-Frobenius Theorem [44] we get that $\hat{A}$ has simple greatest eigenvalue, and the corresponding eigenvector is the only one which has all entries positive. We showed that $\lambda = 1, v_F$ are eigenvalue and corresponding eigenvector, and also that v_F can be determined all entry positive (Proposition 2.2.3), so we get that $\lambda = 1$ is simple eigenvalue of $\hat{A}$. ■

Lemma 3.2.5. *Let $\hat{\mathcal{G}}$ be strongly connected and aperiodic. Then the corresponding $\hat{A}$ has largest eigenvalue equal to* 1*, and all other eigenvalues have absolute value that is strictly less than* 1*.*

Proof. From Perron-Frobenius theorem we know that if the graph is strongly connected and aperiodic, the matrix is called *primitive*, and the real, positive, largest eigenvalue is strictly greater than all other in absolute value. From Lemma 3.2.4, we know that $\lambda = 1$ is the spectral radius of $\hat{A}$. ■

We can now state the corollary that establishes the relationship between the limit of $\hat{A}^k x_0$ as $k \to \infty$ and the Fiedler vector, for some arbitrary x_0. This result will later be used to connect walks on $\hat{\mathcal{G}}$ with the Fiedler vector.

Corollary 3.1. *Let $\hat{\mathcal{G}}$ be strongly connected and aperiodic. Then*

$$\lim_{k\to\infty} \hat{A}^k x_0 \in \text{span}(v_F),$$

for any vector x_0.

Proof. Since $\hat{A}$ is a diagonalizable matrix, we can write it as

$$\hat{A} = V\hat{D}V^{-1},$$

where $\hat{D}$ is a diagonal matrix with eigenvalues of $\hat{A}$ on the diagonal. The ith column of V is the eigenvector corresponding to the eigenvalue λ_i. The i row of V^{-1} is the

left eigenvector of λ_i. Without loss of generality, let's assume the eigenvalues ordered in a descending order from the top left of $\hat{D}$. From Lemma 3.2.5, we know that the largest eigenvalue is strictly larger than all others, with the corresponding eigenvector v_F. Let's take the limit value:

$$\begin{aligned}
&\lim_{k\to\infty} \hat{A}^k x_0 = \lim_{k\to\infty} V\hat{D}^k V^{-1} x_0 \\
&= \lim_{k\to\infty} \begin{bmatrix} v_F & v_2 & v_3 & \cdots & v_n \end{bmatrix} \begin{bmatrix} \lambda_1^k & 0 & 0 & 0 & 0 \\ 0 & \lambda_2^k & 0 & 0 & 0 \\ 0 & 0 & \lambda_3^k & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \lambda_n^k \end{bmatrix} \begin{bmatrix} w_1^T \\ w_2^T \\ w_3^T \\ \vdots \\ w_n^T \end{bmatrix} x_0 \\
&= v_F(w_1^T x_0)
\end{aligned}$$

where $(w_1^T x_0)$ is some scalar value. The final equivalence holds true due to the fact $|\lambda_i| < 1, i = 2 \dots n$. ■

We can now use the previous results to establish a connection between walks on $\hat{\mathcal{G}}$ and the Fiedler vector.

Lemma 3.2.6. *Let $\hat{\mathcal{G}}$ be a strongly connected and aperiodic. Then, the following vector, $\hat{v}$:*

$$[\hat{v}]_i = \lim_{k\to\infty} \sum_j \sum_{\|\Gamma_{ij}\|=k} W_{\Gamma_{ij}}, \quad i,j=1\dots n \tag{3.5}$$

is proportional to the Fiedler vector v_F (recall Fiedler vector of L_B), i.e., $\hat{v} \in \text{span}(v_F)$.

Proof. Using Lemma 3.2.3, we can multiply $\hat{A}^k$ by $\mathbb{1}$ to obtain

$$[\hat{A}^k \mathbb{1}]_i = \sum_j \sum_{\|\Gamma_{ij}\|=k} W_{\Gamma_{ij}}.$$

Then, taking $k \to \infty$,

$$\lim_{k\to\infty} [\hat{A}^k \mathbb{1}]_i = \lim_{k\to\infty} \sum_j \sum_{\|\Gamma_{ij}\|=k} W_{\Gamma_{ij}} = [\hat{v}]_i.$$

Using Corollary 3.1 with $x_0 = \mathbb{1}$, we know that the left side of the equation is in the span of v_F, i.e.,

$$\hat{v} \in \text{span}(v_F). \qquad \blacksquare$$

In Lemma 3.2.6, we provide an expression for the Fiedler vector by considering the limit of traversing the graph through walks. Now, our aim is to establish a bound for the approximated Fiedler vector when finite-length walks are taken into account. To do this, we'll define the error between the approximated Fiedler vector and the actual

Fiedler vector. In more mathematical terms, our goal is to bound the error between the vectors

$$[\hat{v}]_i = \lim_{k\to\infty} [\hat{A}^k \mathbb{1}]_i, \text{ and} \tag{3.6}$$

$$[\hat{v}]_i^k = [\hat{A}^k \mathbb{1}]_i. \tag{3.7}$$

We define the error as the norm of the difference between the vectors, i.e.,

$$\|v_e^k\| := \|\hat{v} - \hat{v}^k\|. \tag{3.8}$$

In the following, we aim to achieve an error bound based on the graph structure.

Proposition 3.2.7. *The error $\|v_e^k\|$ is bounded by:*

$$\|v_e^k\| \leq \lambda_2^k \frac{\max_i(\sqrt{[\hat{D}_{\lambda_F}]_{ii}})}{\min_i(\sqrt{[\hat{D}_{\lambda_F}]_{ii}})}\sqrt{n}.$$

Proof. We will utilize Proposition 3.2.1 to bound the error. We can represent the symmetric matrix $\hat{D}_{\lambda_F}^{-1/2} A \hat{D}_{\lambda_F}^{-1/2}$ in terms of its eigenvalues and eigenvectors as follows:

$$\hat{D}_{\lambda_F}^{-1/2} A \hat{D}_{\lambda_F}^{-1/2} = \sum_{i=1}^{n} \lambda_i \phi_i \phi_i^T,$$

where λ_i and ϕ_i are the eigenvalues and normalized eigenvectors ($\|\phi_i\| = 1$), respectively. From Proposition 3.2.1, we know that λ_i are also eigenvalues of $\hat{A}$. From lemma 3.2.5, we know that $\lambda_1 = 1$ and also strictly larger in absolute value from all others eigenvalues. Now, let's develop an explicit expression for $\hat{v}$ and $\|v_e^k\|$ in terms of λ_i and ϕ_i,

$$\hat{v} = \lim_{k\to\infty} \hat{A}^k \mathbb{1} = \lim_{k\to\infty} \hat{D}_{\lambda_F}^{-1/2} (\hat{D}_{\lambda_F}^{-1/2} A \hat{D}_{\lambda_F}^{-1/2})^k \hat{D}_{\lambda_F}^{1/2} \mathbb{1} \tag{3.9}$$

$$= \lim_{k\to\infty} \hat{D}_{\lambda_F}^{-1/2} \left(\sum_{i=1}^{n} (\lambda_i^k \phi_i \phi_i^T) \right) \hat{D}_{\lambda_F}^{1/2} \mathbb{1} \tag{3.10}$$

$$= \hat{D}_{\lambda_F}^{-1/2} (\lambda_1 \phi_1 \phi_1^T) \hat{D}_{\lambda_F}^{1/2} \mathbb{1} \tag{3.11}$$

$$= \hat{D}_{\lambda_F}^{-1/2} (\phi_1 \phi_1^T) \hat{D}_{\lambda_F}^{1/2} \mathbb{1}. \tag{3.12}$$

Similarly, for $\|v_e^k\|$ we have,

$$\begin{aligned}\|v_e^k\| = \|\hat{A}^k\mathbb{1} - \hat{v}\| &= \|\hat{A}^k\mathbb{1} - \hat{v}\| \\ &= \|\hat{D}_{\lambda_F}^{-1/2}(\hat{D}_{\lambda_F}^{-1/2}A\hat{D}_{\lambda_F}^{-1/2})^k\hat{D}_{\lambda_F}^{1/2}\mathbb{1} - \hat{v}\| \\ &= \|\hat{D}_{\lambda_F}^{-1/2}\sum_{i=1}^{n}(\lambda_i^k\phi_i\phi_i^T)\hat{D}_{\lambda_F}^{1/2}\mathbb{1} - \hat{v}\| \\ &= \|\hat{D}_{\lambda_F}^{-1/2}\sum_{i=2}^{n}(\lambda_i^k\phi_i\phi_i^T)\hat{D}_{\lambda_F}^{1/2}\mathbb{1}\|.\end{aligned}$$

The final step is valid as it cancels out for $i = 1$ with $\hat{v}$, as shown in (3.12). We can bound this error by utilizing properties of the norm:

$$\begin{aligned}\|\hat{D}_{\lambda_F}^{-1/2}\sum_{i=2}^{n}(\lambda_i^k\phi_i\phi_i^T)\hat{D}_{\lambda_F}^{1/2}\mathbb{1}\| &\leq \|\hat{D}_{\lambda_F}^{-1/2}\|\|\sum_{i=2}^{n}(\lambda_i^k\phi_i\phi_i^T)\|\|\hat{D}_{\lambda_F}^{1/2}\|\|\mathbb{1}\| \\ &\leq \lambda_2^k\frac{\max_i(\sqrt{[\hat{D}_{\lambda_F}]_{ii}})}{\min_i(\sqrt{[\hat{D}_{\lambda_F}]_{ii}})}\sqrt{n},\end{aligned}$$

where λ_2 is the second-largest eigenvalue of $\hat{A}$. The transition to the second line is due to the fact that the norm of the matrix is less than or equal to the spectral norm, i.e., $\|\sum_{i=2}^{n}(\lambda_i^k\phi_i\phi_i^T)\| \leq \lambda_2^k$, Additionally, $\|\hat{D}_{\lambda_F}^{-1/2}\|\|\hat{D}_{\lambda_F}^{1/2}\| = \frac{\max_i(\sqrt{[\hat{D}_{\lambda_F}]_{ii}})}{\min_i(\sqrt{[\hat{D}_{\lambda_F}]_{ii}})}$. ■

3.3 Leaders Identification Method

In this section, we aim to present theoretical results for the leader identification method, which will be detailed later. Specifically, we will analyze the behavior of a sequence of graphs under certain structural conditions. We will provide sufficient conditions on the graph structure that ensure a gap between the components corresponding to leaders and followers in the Fiedler vector. We then leverage these results to develop an algorithm for identifying leaders based on this gap.

Before stating the main lemma, we introduce the notion of a *graph sequence*, which allows us to describe the structural properties required in the lemma. The idea is to consider a sequence of graphs in which the leader nodes remain fixed, while the connectivity among the follower nodes gradually increases in a controlled manner. This construction is motivated by the goal of analyzing the components of the Fiedler eigenvector associated with each graph in the sequence. In particular, by studying how the Fiedler vector changes along the sequence, we can show that it converges to a simplified expression that depends explicitly on certain structural features of the graph, providing an analytically approximation.

Definition 3.3.1 (Graph Sequence)**.** Let $\sigma : \mathbb{N} \to \mathbb{N}$ be a mapping, and let $\mathcal{G}^{\sigma(i)} = (\mathcal{V}^{\sigma(i)}, \mathcal{E}^{\sigma(i)})$ denote a sequence of graphs, where the node set is partitioned into disjoint

leader and follower sets,

$$\mathcal{V}^{\sigma(i)} = \mathcal{V}_\ell^{\sigma(i)} \cup \mathcal{V}_f^{\sigma(i)}, \text{ with } \mathcal{V}_\ell^{\sigma(i)} \cap \mathcal{V}_f^{\sigma(i)} = \emptyset.$$

Define the follower's number of follower neighbors and leader neighbors, respectively, as

$$\begin{aligned} d_F^{\sigma(i)}(j) &= \left|\{k \in \mathcal{V}_f^{\sigma(i)} \mid (k,j) \in \mathcal{E}^{\sigma(i)}\}\right| \\ d_L^{\sigma(i)}(j) &= \left|\{k \in \mathcal{V}_\ell^{\sigma(i)} \mid (k,j) \in \mathcal{E}^{\sigma(i)}\}\right|, \quad j \in \mathcal{V}_f^{\sigma(i)}. \end{aligned}$$

Also, denote the minimum follower degree as

$$\underline{d}_F^{\sigma(i)} = \min_{j \in \mathcal{V}_f^{\sigma(i)}} d_F^{\sigma(i)}(j).$$

The sequence of graphs satisfies the following properties:

i) $\mathcal{G}^{\sigma(i)}$ is connected for all i;

ii) $\mathcal{V}_\ell^{\sigma(i)} = \mathcal{V}_\ell^{\sigma(i+1)} = \mathcal{V}_\ell$, $|\mathcal{V}_\ell| \geq 1$ for all i (the leaders sets are constant);

iii) $\mathbf{d}_j := d^{\sigma(i)}(j) = d^{\sigma(i+1)}(j)$ for all i and for all $j \in \mathcal{V}_\ell$ (the leaders degree is constant);

iv) $k \notin \mathcal{N}^{\sigma(i)}(j)$ for all $k, j \in \mathcal{V}_\ell^{\sigma(i)}$ (leader nodes are not connected to each other);

v) $\underline{d}_F^{\sigma(i)} < \underline{d}_F^{\sigma(i+1)}$ for all i (minimum follower's number of followers neighbors is strictly increasing).

With this construction in place, the subsequent lemma can refer to the graph sequence $\{\mathcal{G}^{\sigma(i)}\}$ without repeating all of the above conditions. An example of such a sequence is illustrated in Example 3.3.2, showing the evolution of follower connectivity while maintaining a constant leader set.

Example 3.3.2. To illustrate Definition 3.3.1, consider the graph sequence shown in Figure 3.1. We start the sequence with a 4-node connected graph, where nodes 1 and 2 are leaders, and nodes 3 and 4 are followers. Each leader has degree 2, and the leaders are not directly connected to each other. In this graph, $\underline{d}_F^{\sigma(i)} = 1$, since removing the leader nodes and their adjacent edges leaves a 2-node graph, where the minimum degree is 1. Next, we move to the second graph in the sequence by adding an additional follower node. According to Definition 3.3.1, $\underline{d}_F^{\sigma(i)}$ is strictly increasing along the sequence. We also verify that the leader set and the leader degrees remain constant. The sequence continues by adding more follower nodes while maintaining all the properties described in the definition.

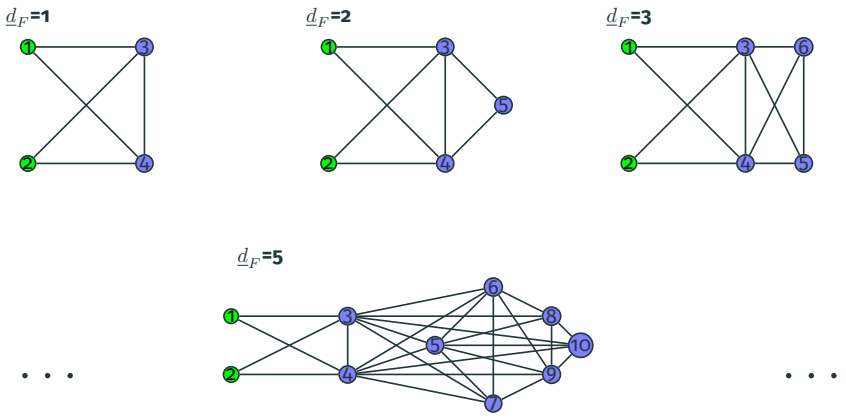


Figure 3.1: Sequence of graphs illustrating the addition of nodes and edges. The green nodes indicate the leader nodes and the blue are the follower nodes.

Lemma 3.3.3. *Let $\mathcal{G}^{\sigma(i)}$ be a sequence of graphs satisfying the conditions in Definition 3.3.1. Then,*

$$\lim_{i\to\infty}[\hat{A}^{\sigma(i)}\bar{v}_F^{\sigma(i)}]_j = \lim_{i\to\infty}\bar{v}_F^{\sigma(i)}$$

where

$$[\bar{v}_F^{\sigma(i)}]_j = \begin{cases} 1, & j \in \mathcal{V}_f^{\sigma(i)} \\ \frac{\mathbf{d}_j}{\mathbf{d}_j+1-\lambda_F^{\sigma(i)}}, & j \in \mathcal{V}_\ell \end{cases}. \tag{3.13}$$

where $\lambda_F^{\sigma(i)}$ is the Fiedler value of the grounded Laplacian associated with the graph $\mathcal{G}^{\sigma(i)}$ and leader set $\mathcal{V}_\ell$.

Proof. The proof is as follows. We will show that the vector $\bar{v}_F^{\sigma(i)}$ is an eigenvector corresponding to eigenvalue 1 of $\hat{A}^{\sigma(i)}$ (defined in 3.1) in the limit as i goes to infinity. Consider the sequence $\hat{A}^{\sigma(i)}$ representing the semi-normalized adjacency matrix of $\mathcal{G}^{\sigma(i)}$, and examine the multiplication of $\hat{A}^{\sigma(i)}$ with $\bar{v}_F^{\sigma(i)}$ taken from (3.13). For notational brevity, set $\lambda_i := \lambda_F^{\sigma(i)}$, $\hat{A}_i := \hat{A}^{\sigma(i)}(\mathcal{G})$, $\bar{v}_i := \bar{v}_F^{\sigma(i)}$, $\mathcal{N}_i(j) := \mathcal{N}^{\sigma(i)}(j)$, and $d_i(j) := d^{\sigma(i)}(j)$. For any node $j \in \mathcal{V}$ we have

$$[\hat{A}_i\bar{v}_i]_j = \frac{1}{d_i(j)+\delta_j-\lambda_i}\sum_{k\in\mathcal{N}_i(j)}[\bar{v}_i]_k, \tag{3.14}$$

where $\delta_j = 1$ if $j \in \mathcal{V}_\ell$ and $\delta_j = 0$ otherwise. Using (3.13) and splitting by fol-

lower/leader neighbors,

$$\sum_{k\in\mathcal{N}_i(j)}[\bar{v}_i]_k = \sum_{k\in\mathcal{N}_i(j)\cap\mathcal{V}_f} 1 + \sum_{k\in\mathcal{N}_i(j)\cap\mathcal{V}_\ell}\frac{\mathbf{d}_k}{\mathbf{d}_k+1-\lambda_i}$$
$$= d_i(j) - (1-\lambda_i)\sum_{k\in\mathcal{N}_i(j)\cap\mathcal{V}_\ell}\frac{1}{\mathbf{d}_k+1-\lambda_i},$$

where we used the identity $\frac{\mathbf{d}_k}{\mathbf{d}_k+1-\lambda_i} = 1 - \frac{1-\lambda_i}{\mathbf{d}_k+1-\lambda_i}$. Substituting into (3.14) and separating the terms yields compact form

$$[\hat{A}_i\bar{v}_i]_j = 1 - \frac{\delta_j-\lambda_i}{d_i(j)+\delta_j-\lambda_i} - \frac{(1-\lambda_i)}{d_i(j)+\delta_j-\lambda_i}\sum_{k\in\mathcal{N}^{\sigma(i)}(j)\cap\mathcal{V}_\ell}\frac{1}{\mathbf{d}_k+1-\lambda_i}. \tag{3.15}$$

We now consider (3.15) for $j\in\mathcal{V}_f$. Here $\delta_j = 0$ and $[\bar{v}_i]_j = 1$. From (3.15),

$$[\hat{A}_i\bar{v}_i]_j = 1 + \frac{\lambda_i}{d_i(j)-\lambda_i} - \frac{(1-\lambda_i)}{d_i(j)-\lambda_i}\sum_{k\in\mathcal{N}^{\sigma(i)}(j)\cap\mathcal{V}_\ell}\frac{1}{\mathbf{d}_k+1-\lambda_i}.$$

By Definition 3.3.1(v), $\underline{d}_F^{\sigma(i)}\to\infty$, while $\lambda_i\in(0,1)$, and the leader degrees $\mathbf{d}_k$ are constant by item (iii). Hence both correction terms vanish, and therefore

$$\lim_{i\to\infty}[\hat{A}_i\bar{v}_i]_j = 1 = \lim_{i\to\infty}[\bar{v}_i]_j.$$

For leader nodes $j\in\mathcal{V}_\ell$, by Definition 3.3.1(iv), leaders are not directly connected, so $\mathcal{N}^{\sigma(i)}(j)\cap\mathcal{V}_\ell = \emptyset$. Equation (3.15) then reduces to

$$[\hat{A}_i\bar{v}_i]_j = 1 - \frac{1-\lambda_i}{d_i(j)+1-\lambda_i} = \frac{d_i(j)}{d_i(j)+1-\lambda_i}$$
$$= \frac{\mathbf{d}_j}{\mathbf{d}_j+1-\lambda_i} = [\bar{v}_i]_j.$$

Hence equality holds for all i. To conclude, we have in both cases $\lim_{i\to\infty}[\hat{A}_i\bar{v}_i]_j = \lim_{i\to\infty}[\bar{v}_i]_j$. ■

We now use Lemma 3.3.3 to establish the convergence of the corresponding Fiedler vectors for the graphs in the sequence.

Corollary 3.2. *For the same conditions stated in Lemma 3.3.3, the Fiedler vector of*

$\mathcal{G}^{\sigma(i)}$ *converges to the following values:*

$$\lim_{i\to\infty}[v_F^{\sigma(i)}]_j = c\lim_{i\to\infty}\begin{cases}1, & j\in\mathcal{V}_f^{\sigma(i)}\\ \frac{\mathbf{d}_j}{\mathbf{d}_j+1-\lambda_F^{\sigma(i)}}, & j\in\mathcal{V}_\ell\end{cases}, \quad c\in\mathbb{R}.$$

Proof. From 3.2.4 we know that $\hat{A}$ has eigenvalue $\lambda = 1$ with the corresponding eigenvector v_F which is the Fidler vector. Combined this with Lemma 3.3.3 gives our results. ■

The result of the last corollary shows that the Fiedler vector converges to the sum of the rows of $\hat{A}$. This is equivalent to the term we previously developed in (3.7) for the case when $k = 1$. The following theorem leverages our previous results to provide a sufficient condition on the graph structure that ensures a separation between the Fiedler vector components corresponding to leaders and followers. This separation will later enable us to identify the leaders.

Theorem 3.3. *Let $\mathcal{G}$ be graph where the nodes are separated into two groups, leaders $\mathcal{V}_\ell$ and followers $\mathcal{V}_f$. Denote by $d_F(j) = |\{k\in\mathcal{V}_f \,|\, j\in\mathcal{V}_f, (k,j)\in\mathcal{E}\}|$ (the follower's number of followers neighbors), and $\underline{d}_F = \min_j d_F(j)$ (the minimum follower's number of followers neighbors). If the following conditions are met:*

i) $\mathcal{G}$ is connected;

ii) $k\notin\mathcal{N}(j)$ for all $k,j\in\mathcal{V}_\ell$ (leader nodes are not connected to each other);

iii) $1-\max_{j\in\mathcal{V}_\ell}\frac{d(j)}{d(j)+1-\lambda_F} > \max_{j\in\mathcal{V}_\ell}\min_{k\in\mathcal{V}_\ell}|\frac{d(j)}{d(j)+1-\lambda_F}-\frac{d(k)}{d(k)+1-\lambda_F}|$;

iv) $\underline{d}_F$ is sufficient large,

where λ_F is the Fiedler eigenvalue of $\mathcal{G}$, then

$$\min_{i\in\mathcal{V}_f}[v_F]_i - \max_{i\in\mathcal{V}_\ell}[v_F]_i > \max_{i\in\mathcal{V}_\ell}\min_{j\in\mathcal{V}_\ell}|[v_F]_i-[v_F]_j|.$$

Proof. Denote

$$\epsilon_d = 1-\max_{j\in\mathcal{V}_\ell}\frac{d(j)}{d(j)+1-\lambda_F} - \max_{j\in\mathcal{V}_\ell}\min_{k\in\mathcal{V}_\ell}|\frac{d(j)}{d(j)+1-\lambda_F}-\frac{d(k)}{d(k)+1-\lambda_F}| > 0.$$

We define a graph sequence $\sigma:\mathbb{N}\to\mathbb{N}$, $\mathcal{G}^{\sigma(i)} = (\mathcal{V}^{\sigma(i)},\mathcal{E}^{\sigma(i)})$ that satisfies the following conditions:

i) $\mathcal{G}^{\sigma(i)}$ is connected for all i;

ii) $\mathcal{V}_\ell^{\sigma(i)} = \mathcal{V}_\ell^{\sigma(i+1)} = \mathcal{V}_\ell$, $|\mathcal{V}_\ell|\geq 1$ for all i (the leaders sets are constant);

iii) $d^{\sigma(i)}(j) = d^{\sigma(i+1)}(j) = \mathbf{d}_j$ for all i and for all $j\in\mathcal{V}_\ell$ (the leaders degree is constant);

iv) $k \notin \mathcal{N}^{\sigma(i)}(j)$ for all $k, j \in \mathcal{V}_\ell^{\sigma(i)}$ (leader nodes are not connected to each other);

v) $\underline{d}_F^{\sigma(i)} < \underline{d}_F^{\sigma(i+1)}$ for all i (minimum follower's number of followers neighbors is strictly increasing).

From Corollary 3.2, we know that

$$\lim_{i\to\infty} \|c^{\sigma(i)} v_F^{\sigma(i)} - \bar{v}_F^{\sigma(i)}\|^2 = 0,$$

where

$$[\bar{v}_F^{\sigma(i)}]_j = \begin{cases} 1, & j \in \mathcal{V}_f^{\sigma(i)} \\ \frac{d_j}{d_j+1-\lambda_F^{\sigma(i)}}, & j \in \mathcal{V}_\ell^{\sigma(i)} \end{cases}$$

and the scaling factor $c^{\sigma(i)}$ is chosen such that $\|c^{\sigma(i)} v_F^{\sigma(i)}\| = \|\bar{v}_F^{\sigma(i)}\|$.

Now, let $S^{\sigma(i)} = \|\bar{v}_F^{\sigma(i)} - c^{\sigma(i)} v_F^{\sigma(i)}\|_\infty$ where $\|x\|_\infty = \max_i |x|_i$. The sequence S^i converges to zero, and hence it is also a Cauchy sequence. Therefore, for any arbitrarily small $\epsilon > 0$, there exists an index i^* such that for all $i > i^*$, we have $S^i < \epsilon$.

Now, choose i^* such that $\underline{d}_F^{\sigma(i^*)} = m$, and pick ϵ small enough to satisfy

$$\epsilon < \frac{\epsilon_d}{4}.$$

This choice ensures that the subsequent analysis holds for the desired level of accuracy. It follows that,

$$\begin{aligned}
\min_{j\in\mathcal{V}_f}[c^{\sigma(i)} v_F^{\sigma(i)}]_j - \max_{j\in\mathcal{V}_\ell}[c^{\sigma(i)} v_F^{\sigma(i)}]_j &> 1 - \max_{j\in\mathcal{V}_\ell} \frac{d_j}{d_j+1-\lambda_F} - 2\epsilon \\
&= \max_{j\in\mathcal{V}_\ell} \min_{k\in\mathcal{V}_\ell} |\frac{d_j}{d_j+1-\lambda_F} - \frac{d_k}{d_k+1-\lambda_F}| + \epsilon_d - 2\epsilon \\
&> \max_{j\in\mathcal{V}_\ell} \min_{k\in\mathcal{V}_\ell} c^{\sigma(i)} |[v_F^{\sigma(i)}]_j - [v_F^{\sigma(i)}]_k| + \epsilon_d - 4\epsilon \\
&> \max_{j\in\mathcal{V}_\ell} \min_{k\in\mathcal{V}_\ell} c^{\sigma(i)} |[v_F^{\sigma(i)}]_j - [v_F^{\sigma(i)}]_k|
\end{aligned}$$

In the first line we use $S^{\sigma(i)} < \epsilon$. This bound is applied twice: once to the leaders' components and once to the followers'. The transition from first to second line is by using the definition of ϵ_d. In the third line we are using again $S^{\sigma(i)} < \epsilon$. The transition from third to forth line is by using $\epsilon < \frac{\epsilon_d}{4}$.

By normalizing both sides with $c^{\sigma(i)} > 0$ we obtain,

$$\min_{j\in\mathcal{V}_f}[v_F]_j - \max_{j\in\mathcal{V}_\ell}[v_F]_j > \max_{j\in\mathcal{V}_\ell} \min_{k\in\mathcal{V}_\ell} |[v_F]_j - [v_F]_k|.$$

Therefore, if our graph $\mathcal{G}$ has $\underline{d}_F^{\mathcal{G}} > m$, we get our results. ■

Based on the results of the theorem, there is a clear separation between the leader and follower components in the Fiedler vector. Recall that this separation is also

reflected in the agents' velocities, as discussed in Section 2.2.3. We can exploit this separation to propose an algorithm for identifying the leaders using measurements of the agents' velocities. Based on the results established in Theorem 3.3, i.e.,

$$\min_{i\in\mathcal{V}_f}[v_F]_i - \max_{i\in\mathcal{V}_\ell}[v_F]_i > \max_{i\in\mathcal{V}_\ell}\min_{j\in\mathcal{V}_\ell}|[v_F]_i - [v_F]_j|, \tag{3.16}$$

then we can identify the leaders using Algorithm 3.1 (below).

Algorithm 3.1 External Observer Based Leaders Identification

Step 1: Measure the agents velocities to an external constant input until steady state.
Step 2: Calculate the relative tempo and compute the Fiedler vector as described in Section 2.2.3.
Step 3: Sort the Fiedler vector $v_{F_s} = sort(v_F)$ where $[v_{F_s}]_i \leq [v_{F_s}]_{i+1}$.
Step 4: Calculate the number of leaders n_l with

$$n_l = |\mathcal{V}_\ell| = \arg\max_{j\in\{1,2,3,\cdots,n-1\}}\{[v_{F_s}]_{j+1} - [v_{F_s}]_j\}.$$

Step 5: The leaders correspond to the smallest n_l components in v_{F_s}.

Algorithm 3.1 provides a systematic method to identify leaders based on the Fiedler vector derived from agents' velocity measurements. Step 1 ensures the system reaches steady state so that transient dynamics do not affect the estimation. Step 2 transforms the velocity data into the relative tempo, which gives us an estimation of the Fiedler vector. Sorting the Fiedler vector in Step 3 reveals the natural separation between leaders and followers, and Step 4 identifies the number of leaders by detecting the largest gap in the sorted vector. Finally, Step 5 maps these components back to the actual leader agents.

This algorithm works under the assumptions of Theorem 3.3, where a clear separation exists between leader and follower components. Potential limitations include sensitivity to measurement noise, dynamic changes in network topology, or deviations from the assumed linear dynamics.

Chapter 4

Simulation Examples

In this chapter, we provide examples of networks that support our findings from Theorem 3.3 and Algorithm 3.1. To illustrate that our theorem conditions are sufficient and not necessary, we will present 2 examples:

1) Theorem conditions satisfied, algorithm identified successfully.

2) Theorem conditions unsatisfied, algorithm identified successfully.

The following examples demonstrate a 2D scenario, where the leaders are the green nodes. We consider a system with $n = 10$ agents, where $\{1, 2, 3\} \in \mathcal{V}_\ell$. The external input provided to the leaders are given as

$$u_1^{ex} = [40, 35]^T, \;\; u_2^{ex} = [48, 44]^T, \;\; u_3^{ex} = [16, 45]^T.$$

4.1 Algorithm Results: Theorem Conditions Met

The underlying network graph is illustrated in Figure 4.1. The protocol is simulated, and the resulting trajectories are shown in Figure 4.2. The relative tempo is depicted in Figure 4.3. As expected, the leader and follower components are clearly separated.

Next, we verify the conditions outlined in Theorem 3.3. First, let us check condition (iii):

$$\epsilon_d = 1 - \max_{j\in\mathcal{V}_\ell} \frac{d(j)}{d(j)+1-\lambda_F} - \max_{j\in\mathcal{V}_\ell} \min_{k\in\mathcal{V}_\ell} |\frac{d(j)}{d(j)+1-\lambda_F} - \frac{d(k)}{d(k)+1-\lambda_F}| = 0.4498 > 0.$$

We can see that ϵ_d has a positive value, which means condition (iii) is satisfied. Next, for condition (iv), we require a sufficiently large $\underline{d}_F$ to ensure $\epsilon < \frac{\epsilon_d}{4}$.We compute $\bar{v}_F$ for both the leader and follower components as

$$[\bar{v}_F]_j = \begin{cases} 1, & j \in \mathcal{V}_f, \\ 0.5488, & j \in \mathcal{V}_\ell. \end{cases}$$

The true Fiedler vector, normalized such that $\|v_F\| = \|\bar{v}_F\|$, is given by

$$v_F = [0.5072,\ 0.5054,\ 0.5108,\ 0.9243,\ 0.9308,\ 0.9210,\ 1.0697,\ 1.1019,\ 1.0209,\ 1.0771].$$

Therefore, we obtain

$$\epsilon = \|v_F - \bar{v}_F\|_\infty = 0.1019,$$

and

$$\epsilon_d = 1 - 0.5488 = 0.4512.$$

Hence, we have sufficiently large $\underline{d}_F$, yields that condition (iv) of the theorem is satisfied, since

$$\epsilon = 0.1019 < 0.1124 = \frac{\epsilon_d}{4}.$$

Since all conditions are satisfied, there is sufficient separation between the Fiedler vector components corresponding to leaders and followers. Therefore, Algorithm 3.1 can be used to identify the leaders. We can validate the theorem result by calculating the separation of the leader and follower components (we taking the normalized Fiedler vector such that $\|v_F\| = 1$):

$$\min_{i\in\mathcal{V}_f}[v_F]_i - \max_{i\in\mathcal{V}_\ell}[v_F]_i = 0.3276 - 0.1817 = 0.1459 > \max_{i\in\mathcal{V}_\ell}\min_{j\in\mathcal{V}_\ell}|[v_F]_i - [v_F]_j| = 0.0006.$$

The following describes how the algorithm operates in this case.

Step 1: Measure the agents' velocities to an external constant input until steady state.

Step 2: Calculate the relative tempo and compute the Fiedler vector as described in section 2.2.3:

$$v_F = [0.1804, 0.1798, 0.1817, 0.3288, 0.3311, 0.3276, 0.3805, 0.3920, 0.3631, 0.3831]$$

Step 3: Sort the Fiedler vector $v_{F_s} = sort(v_F)$ where $[v_{F_s}]_i \leq [v_{F_s}]_{i+1}$:

$$v_{F_s} = [0.1798, 0.1804, 0.1817, 0.3276, 0.3288, 0.3311, 0.3631, 0.3805, 0.3831, 0.3920].$$

Step 4: Calculate the number of leaders n_l with

$$n_l = |\mathcal{V}_\ell| = \arg\max_{j\in\{1,2,3,\cdots,n-1\}}\{[v_{F_s}]_{j+1} - [v_{F_s}]_j\} = 3.$$

Step 5: The leaders correspond to the smallest n_l components in v_{F_s},

$$\mathcal{V}_\ell = \{1, 2, 3\}.$$

We successfully identified the leader agents in this scenario. In the next example, we present a graph that does not satisfy the conditions of the theorem, yet we apply the

algorithm to identify the leaders.

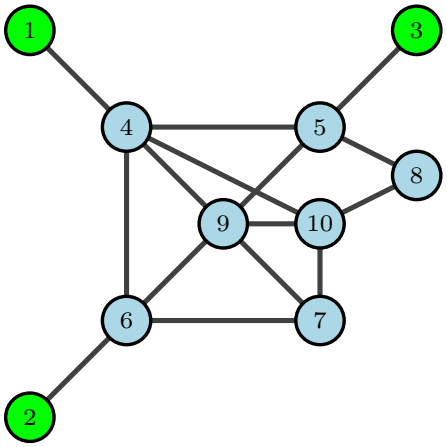


Figure 4.1: Underlying graph of Example 1.

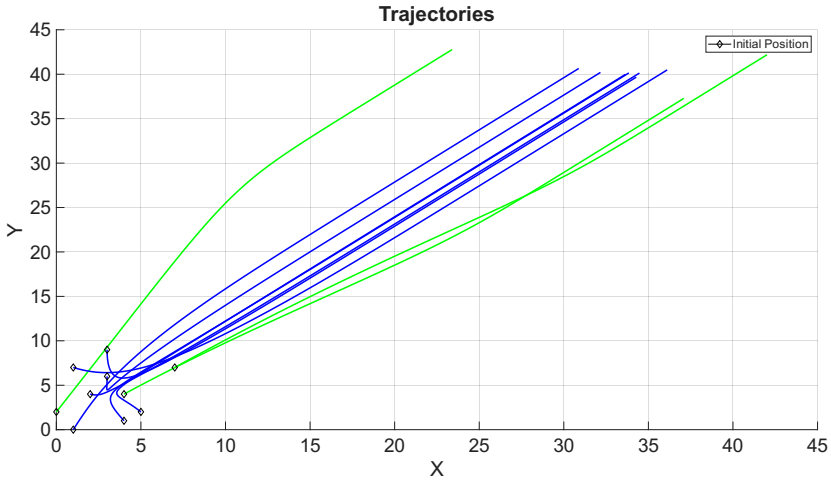


Figure 4.2: Trajectories corresponding to the underlying graph in Example 1. Diamonds mark the initial states. The blue curve corresponds to the followers, while the green curve represents the leaders.

4.2 Algorithm Results: Theorem Conditions Not Met

The underlying network graph is illustrated in Figure 4.4. Since the leaders connected between them, condition (ii) is not satisfied. Calculating condition (iii) expression yields

$$\epsilon_d = 1 - \max_{j\in\mathcal{V}_\ell} \frac{d(j)}{d(j)+1-\lambda_F} - \max_{j\in\mathcal{V}_\ell} \min_{k\in\mathcal{V}_\ell} \left|\frac{d(j)}{d(j)+1-\lambda_F} - \frac{d(k)}{d(k)+1-\lambda_F}\right| = 0.1432 > 0.$$

which satisfies the condition. As for the final condition, we require

$$\epsilon < \frac{\epsilon_d}{4}.$$

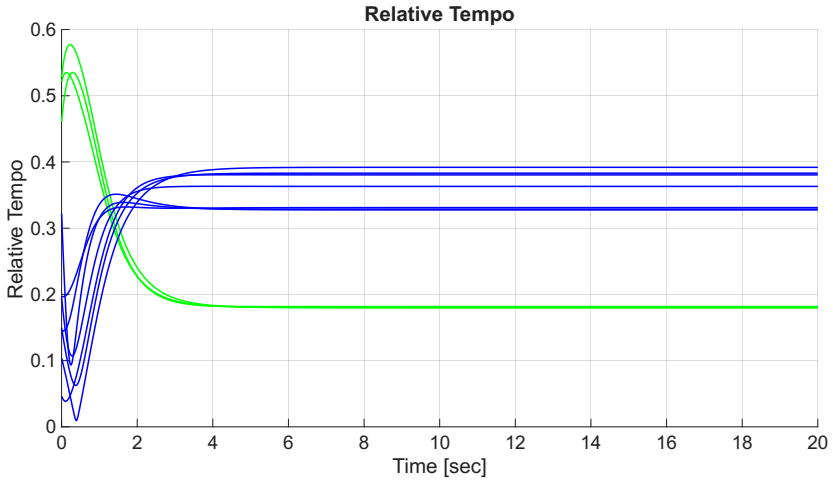


Figure 4.3: Relative tempo for the network in Example 1. Note the gap between leaders and follower components. The blue curve corresponds to the followers, while the green curve represents the leaders.

Numerically,

$$\epsilon = 0.47 \quad \text{and} \quad \frac{\epsilon_d}{4} = 0.0358,$$

so the inequality does **not** hold (indeed, $0.47 > 0.0358$). Therefore, the condition is not satisfied. Although the conditions are not satisfied, the algorithm still successfully estimate the leader agents.

Below is a description of how Algorithm 3.1 operates in this case.

Step 1: Measure the agents velocities to an external constant input until steady state.

Step 2: Calculate the relative tempo and compute the Fiedler vector as described in Section 2.2.3:

$$v_F = [0.14, 0.15, 0.15, 0.43, 0.5, 0.32, 0.38, 0.25, 0.34, 0.29].$$

Step 3: Sort the Fiedler vector $v_{F_s} = sort(v_F)$ where $[v_{F_s}]_i \leq [v_{F_s}]_{i+1}$:

$$v_{F_s} = [0.14, 0.15, 0.15, 0.25, 0.29, 0.32, 0.34, 0.38, 0.43, 0.5].$$

Step 4: Calculate the number of leaders n_l with

$$n_l = |\mathcal{V}_\ell| = \arg \max_{j \in \{1,2,3,\cdots,n-1\}} \{[v_{F_s}]_{j+1} - [v_{F_s}]_j\} = 3.$$

Step 5: The leaders are corresponding to the smallest n_l components in v_{F_s}.

$$\mathcal{V}_\ell = \{1, 2, 3\}.$$

We also obtained an accurate identification of the leaders in this case, demonstrating that the conditions stated in our theorem are sufficient but not necessary.

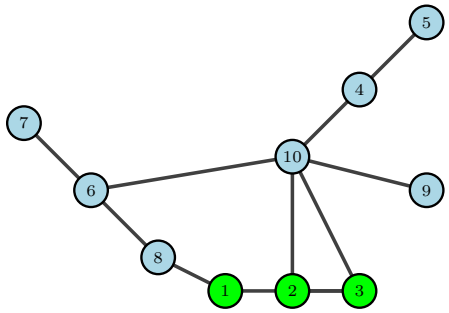


Figure 4.4: Underlying graph of Example 2.

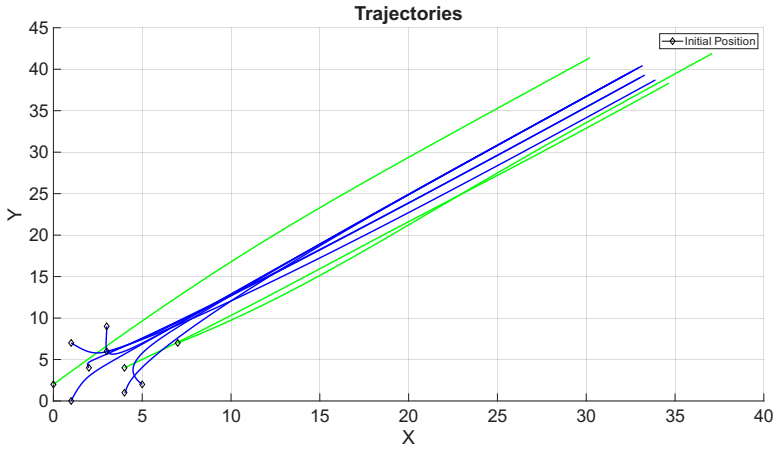


Figure 4.5: Trajectories corresponding to the underlying graph in Example 2. The blue curve corresponds to the followers, while the green curve represents the leaders.

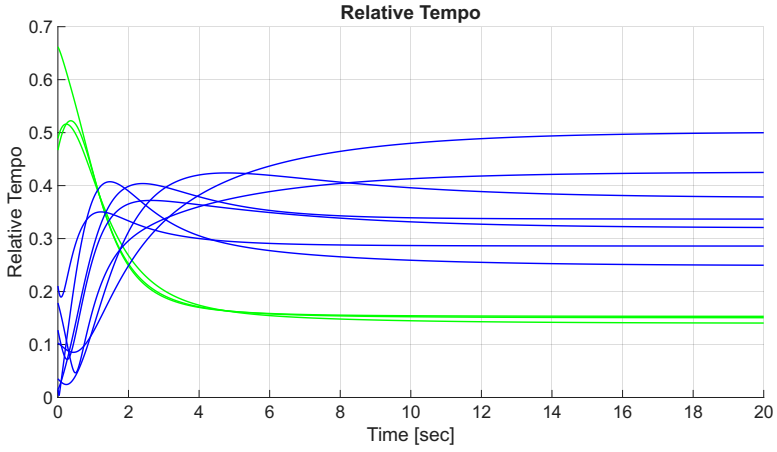


Figure 4.6: Relative tempo for the network in Example 2; note the gap between leaders and follower components. The blue curve corresponds to the followers, while the green curve represents the leaders. We observe that although the theorem conditions are not satisfied, there is still a gap between the relative tempo components associated with the leaders and the followers, illustrating that the theorem conditions are sufficient but not necessary.

Chapter 5

Conclusion and Future Work

In this chapter, we conclude the work conducted for this thesis and also suggest some future orientation following our results.

5.1 Conclusion

Our findings indicate that certain graph structures are more likely to exhibit a clear separation between the components of the Fiedler vector. Specifically, the graphs identified in this research tend to have relatively dense connectivity—characterized by a high mean node degree—while the leader nodes themselves maintain comparatively lower degrees.

This structural property enables effective leader identification through external observation, particularly in scenarios involving constant external input. In such cases, the agents' steady-state velocities are directly related to the corresponding components of the Fiedler vector. This relationship allows us to reliably distinguish and identify the leader agents within the network.

5.2 Future Work

Future research can extend this work in several directions. One promising direction is to investigate scenarios involving non-constant external input signals, or inputs that remain constant only over specific time intervals—corresponding to signals with discrete frequency updates. This can be related to exploring the connection between the convergence rate of the relative tempo and the graph structure. Another important avenue is the development of methods for identifying not only the leader nodes but the complete underlying network structure. Also, the case of directed and weighted graphs can be investigated. Finally, exploring a broader range of graph topologies that exhibit component separation in the Fiedler vector could provide deeper insight into how structural properties influence leader–follower dynamics.

Bibliography

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

[2] H. Shao and M. Mesbahi, "Degree of relative influence for consensus-type networks," in *2014 American Control Conference*, p. 2676–2681, IEEE, June 2014.

[3] R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs," *Proceedings of the IEEE*, vol. 94, p. 1306–1324, July 2006.

[4] J. D. Monnier, G. Rau, *et al.*, "A realistic roadmap to formation flying space interferometry," White Paper / Technical Report NASA NTRS 20205000692, NASA Technical Reports Server (NTRS), Dec. 2019. Acquired April 9, 2020.

[5] P. Wang, J. Yee, C. Xia, M. Mokuno, and F. Hadaegh, "Cooperative control of a magnetically levitated interferometer," in *Proceedings of the 2004 IEEE International Conference on Control Applications, 2004.*, CCA-04, p. 18–25, IEEE.

[6] J. Wu, K. Chen, L. Wang, K. Ren, and M. Han, "Overview of wireless sensor network technology and its applications," in *2025 IEEE 12th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, p. 44–48, IEEE, May 2025.

[7] J. Hu, L. Shen, Y. Yang, and R. Lv, "Design and implementation of wireless sensor and actor network for precision agriculture," in *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, p. 571–575, IEEE, June 2010.

[8] D. Zelazo, R. Dai, and M. Mesbahi, "An energy management system for off-grid power systems," *Energy Systems*, vol. 3, p. 153–179, Jan. 2012.

[9] D. Hoa Nguyen, H. Ngoc Tran, T. Narikiyo, and M. Kawanishi, *A Distributed Optimization Method for Optimal Energy Management in Smart Grid*. IntechOpen, Jan. 2020.

[10] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, p. 1520–1533, Sept. 2004.

[11] Y. Hatano and M. Mesbahi, "Agreement over random networks," *IEEE Transactions on Automatic Control*, vol. 50, p. 1867–1872, Nov. 2005.

[12] L. Wang and F. Xiao, "Finite-time consensus problems for networks of dynamic agents," *IEEE Transactions on Automatic Control*, vol. 55, p. 950–955, Apr. 2010.

[13] M. Huang and J. Manton, "Stochastic consensus seeking with noisy and directed inter-agent communication: Fixed and randomly varying topologies," *IEEE Transactions on Automatic Control*, vol. 55, p. 235–241, Jan. 2010.

[14] F. Chen and D. V. Dimarogonas, "Consensus control for leader-follower multi-agent systems under prescribed performance guarantees," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, Dec. 2019.

[15] A. Chapman and M. Mesbahi, "Semi-autonomous consensus: Network measures and adaptive trees," *IEEE Transactions on Automatic Control*, vol. 58, p. 19–31, Jan. 2013.

[16] Z. Qin, H.-N. Wu, and J.-L. Wang, "Proactive cooperative consensus control for a class of human-in-the-loop multi-agent systems with human time-delays," *Neurocomputing*, vol. 581, p. 127485, May 2024.

[17] Q. Li, X. Wang, M. Sun, W. Liu, and J. Wu, "Sneak attack against mobile robotic networks under formation control," in *Proceedings of the 37th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 2190–2195, 2017.

[18] P. Vera-Licona, A. Jarrah, L. Garcia-Puente, J. McGee, and R. Laubenbacher, "An algebra-based method for inferring gene regulatory networks," *BMC Systems Biology*, vol. 8, no. 37, 2014.

[19] K. J. Friston, "Functional and effective connectivity: A review," *Brain Connectivity*, vol. 1, p. 13–36, Jan. 2011.

[20] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010.

[21] D. Materassi and G. Innocenti, "Topological identification in networks of dynamical systems," *IEEE Transactions on Automatic Control*, vol. 55, p. 1860–1871, Aug. 2010.

[22] M. Timme and J. Casadiego, "Revealing networks from dynamics: an introduction," *Journal of Physics A: Mathematical and Theoretical*, vol. 47, p. 343001, Aug. 2014.

[23] S. Shahrampour and V. M. Preciado, "Reconstruction of directed networks from consensus dynamics," in *2013 American Control Conference*, p. 1685–1690, IEEE, June 2013.

[24] C. Liu, J. He, S. Zhu, and C. Chen, "Dynamic topology inference via external observation for multi-robot formation control," in *2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, p. 1–6, IEEE, Aug. 2019.

[25] S. G. Shandilya and M. Timme, "Inferring network topology from complex dynamics," *New Journal of Physics*, vol. 13, no. 1, p. 013004, 2011.

[26] J. Goncalves and S. Warnick, "Necessary and sufficient conditions for dynamical structure reconstruction of LTI networks," *IEEE Transactions on Automatic Control*, vol. 53, p. 1670–1674, Aug. 2008.

[27] R.-Q. Su, W.-X. Wang, and Y.-C. Lai, "Detecting hidden nodes in complex networks from time series," *Physical Review E*, vol. 85, June 2012.

[28] V. M. Preciado and A. Jadbabaie, "Moment-based spectral analysis of large-scale networks using local structural information," *IEEE/ACM Transactions on Networking*, vol. 21, p. 373–382, Apr. 2013.

[29] V. M. Preciado and A. Jadbabaie, "Spectral analysis of virus spreading in random geometric networks," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, p. 4802–4807, IEEE, Dec. 2009.

[30] S. Hassan-Moghaddam, N. K. Dhingra, and M. R. Jovanovic, "Topology identification of undirected consensus networks via sparse inverse covariance estimation," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, p. 4624–4629, IEEE, Dec. 2016.

[31] S. Talukdar, D. Deka, S. Attree, D. Materassi, and M. Salapaka, "Learning the exact topology of undirected consensus networks," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, p. 5784–5789, IEEE, Dec. 2017.

[32] S. Segarra, M. T. Schaub, and A. Jadbabaie, "Network inference from consensus dynamics," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, Dec. 2017.

[33] M. Sharf and D. Zelazo, "Network identification for diffusively coupled networks with minimal time complexity," *IEEE Transactions on Control of Network Systems*, vol. 10, p. 1616–1628, Sept. 2023.

[34] M. Coutino, E. Isufi, T. Maehara, and G. Leus, "State-space network topology identification from partial observations," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, p. 211–225, 2020.

[35] X. Zhang, G. Wang, T. Cai, and J. Sun, "Network topology identification under the multi-agent agreement protocol," *Journal of the Franklin Institute*, vol. 358, p. 6759–6774, Sept. 2021.

[36] D. A. Spielman and S.-H. Teng, "Spectral partitioning works: Planar graphs and finite element meshes," *Linear Algebra and its Applications*, vol. 421, p. 284–305, Mar. 2007.

[37] I. Rocha and V. Trevisan, "A fiedler-like theory for the perturbed Laplacian," *Czechoslovak Mathematical Journal*, vol. 66, pp. 717–735, Sept. 2016.

[38] R. B. Bapat, S. J. Kirkland, and S. Pati, "The perturbed laplacian matrix of a graph," *Linear and Multilinear Algebra*, vol. 49, p. 219–242, Dec. 2001.

[39] M. Pirani and S. Sundaram, "Spectral properties of the grounded laplacian matrix with applications to consensus in the presence of stubborn agents," in *2014 American Control Conference*, IEEE, June 2014.

[40] M. Pirani and S. Sundaram, "On the smallest eigenvalue of grounded laplacian matrices," *IEEE Transactions on Automatic Control*, vol. 61, no. 2, pp. 509–514, 2016.

[41] D. B. West, *Introduction to Graph Theory*. Prentice Hall, 2nd ed., 2001.

[42] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer New York, 2001.

[43] U. Miekkala, "Graph properties for splitting with grounded laplacian matrices," *BIT*, vol. 33, p. 485–495, Sept. 1993.

[44] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, Oct. 2012.

[45] V. Noferini and M. C. Quintana, "Generating functions of non-backtracking walks on weighted digraphs: Radius of convergence and ihara's theorem," *Linear Algebra and its Applications*, vol. 699, p. 72–106, Oct. 2024.

תקציר

זיהוי מובילים בפרוטוקולי קונצנזוס חצי-אוטונומיים

פרוטוקול הקונצנזוס הוא אחד המרכיבים המרכזיים והבסיסיים במערכות מרובות־סוכנים, והוא ממלא תפקיד מרכזי בהבטחת תיאום, שיתוף פעולה וסנכרון בין סוכנים הפועלים בסביבה משותפת. מטרתו של הפרוטוקול היא לאפשר לסוכנים המרכיבים את המערכת – אשר כל אחד מהם פועל באופן מקומי, עם גישה מוגבלת למידע גלובלי – להגיע להסכמה או לסנכרון של מצבי המערכת באמצעות החלפת מידע עם שכניהם בלבד. כך, ללא צורך במרכז בקרה יחיד או בתקשורת ישירה עם כל שאר הסוכנים, מתאפשרת קבלת החלטות מבוזרת ויציבה, התורמת לחסינות, לגמישות וליכולת הסתגלות של המערכת לשינויים חיצוניים.

בפרוטוקולי קונצנזוס חצי־אוטונומיים, המבנה נעשה מורכב יותר: חלק מהסוכנים, המכונים מובילים, מקבלים קלטים חיצוניים ממקור חיצוני המשפיע על התנהגותם, בעוד שהיתר – העוקבים – מעדכנים את מצבם אך ורק בהתאם למידע שמתקבל מהסוכנים הסמוכים להם ברשת. סידור זה מאפשר למערכת להגיב בצורה מבוזרת לקלטים חיצוניים ולהעביר את ההשפעה של המובילים לשאר הרשת. עם זאת, מבנה זה מעורר גם קושי עקרוני ומשמעותי: כיצד ניתן לזהות מי הם הסוכנים המובילים מתוך צפייה בלבד בהתנהגות הדינמית של המערכת, ובמיוחד כאשר אין ידע מוקדם על מבנה הרשת או על האותות החיצוניים שהמערכת מקבלת.

העבודה עוסקת בבעיה זו של זיהוי סוכנים מובילים מתוך תצפיות על התנהגות המערכת בלבד. בפרט, אנו בוחנים את תגובת המעבר של המערכת לקלט חיצוני עד להתכנסות המערכת. לשם כך, נבחנת התנהגות הווקטור הפידלרי , הקשור לערך העצמי הקטן ביותר שאינו אפס של מטריצת הלפלסיאן של גרף הקשרים בין הסוכנים. וקטור זה נושא מידע חשוב על מבנה הרשת ועל האופן שבו סוכנים שונים משפיעים על הדינמיקה הכללית. מחקרים קודמים הצביעו על קשר בין רכיבי הווקטור הפידלרי לבין תגובת המערכת לקלטים חיצוניים, ובפרט בין ערכיו לבין מהירות הסוכנים בזמן התכנסות המערכת.

במחקר זה אנו מציעים גישה תאורטית המנתחת את מבנה הגרף על מנת לקבל הפרדה ברכיבי הווקטור הפידלרי בין הערכים של המובילים לבין הערכים של העוקבים. עקב קשר בין ערכי הווקטור הפידלרי לבין מהירות הסוכנים בתגובת המעבר לקלט חיצוני, הפרדה ברכיבי הווקטור מניבה גם הפרדה במהירויות הסוכנים. ניתן לנצל הפרדה זו על מנת לזהות את המובילים לפי מדידת המהירויות של הסוכנים. מטרת עבודה זו היא למצוא גרפים אשר מניבים את ההפרדה הזו בווקטור הפידלרי של מטריצת הלפלסיאן של הגרף. על מנת למצוא גרפים אלו, אנו בוחנים סדרה של גרפים הולכים וגדלים, ומראים כיצד רכיבי הווקטור הפידלרי מתנהגים ככל שמספר הצמתים במערכת גדל. בפרט, אנו מראים כי רכיבי הווקטור מתכנסים לערכים התלויים בדרגת הקישוריות המקומית של כל צומת, מה שמאפשר לנו לפתח ביטוי אנליטי יחסית מפושט לווקטור הפידלרי כתלות במבנה הגרף, עבור מקרה הגבול של הסדרה. בנוסף, על סמך התנהגות הסדרה, אנו מציגים תנאי מספיק שלפיו עבור גרפים גדולים מספיק ובעלי מבנה מסוים, מתקבלת הפרדה ברורה בין רכיבי הווקטור של המובילים לבין אלה של העוקבים .תוצאה זו מאפשרת לזהות את המובילים על בסיס מדידת מהירות הסוכנים בלבד, מבלי להזדקק למידע על נתוני הקלט החיצוניים או למידע על מבנה הגרף.

בהמשך מוצע אלגוריתם לזיהוי סוכנים מובילים ,המבוסס על מדידות של מהירות הסוכנים בלבד בתגובת המעבר של המערכת לקלט חיצוני וקבוע. האלגוריתם מנצל את ההפרדה של רכיבי הווקטור הפידלרי כדי לזהות אילו סוכנים מקבלים השפעה חיצונית ישירה, כלומר מתפקדים כמובילים. לבסוף מוצגות סימולציות מספריות המדגימות את יעילות הגישה המוצעת ואת היכולת שלה לזהות במדויק את קבוצת המובילים.

באופן כללי, העבודה תורמת להבנה מעמיקה של הקשר בין תכונות ספקטרליות של רשת לבין הדינמיקה של פרוטוקולי קונצנזוס, ומדגימה כיצד כלים אנליטיים מתחום תורת הגרפים יכולים לשמש לזיהוי תפקידים מבניים במערכות מבוזרות.
מילות מפתח :רשתות מרובות סוכנים, פרוטוקול קונצנזוס, זיהוי רשתות .

המחקר בוצע בהנחייתו של פרופסור דניאל זלזו, בפקולטה למערכות אוטונומיות ורובוטיקה. מחבר/ת חיבור זה מצהיר/ה כי המחקר, כולל איסוף הנתונים, עיבודם והצגתם, התייחסות והשוואה למחקרים קודמים וכו', נעשה כולו בצורה ישרה, כמצופה ממחקר מדעי המבוצע לפי אמות המידה האתיות של העולם האקדמי. כמו כן, הדיווח על המחקר ותוצאותיו בחיבור זה נעשה בצורה ישרה ומלאה, לפי אותן אמות מידה.

זיהוי מובילים בפרוטוקול קונצנזוס סמי-אוטונומי

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר מגיסטר למדעים ב

מערכות אוטונומיות ורובוטיקה

אביתר מטמון

הוגש לסנט הטכניון – מכון טכנולוגי לשראל

חיפה

חשוון תשפ"ו, אוקטובר 2025