Politecnico di Torino

Politecnico di Torino

Master of Science in Computer Engineering

Academic Year 2025/2026

Graduation Session March/April 2026

Control of Cyber-Physical Systems under attack using a Leader-Follower network model

Supervisors:
Sophie Marie Fosson
Diego Regruto Tomalino

Candidate:
Lorenzo Scorrano

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Context and Motivation

Over the past few decades, Cyber-Physical Systems (CPS) have assumed a central role in the development of autonomous technologies, finding widespread applications in sensor networks, drone swarms, smart grids, and cooperative transportation systems[1, preface, first chapter introduction]. At the core of many of these applications is the *leader-follower dynamics*, a cooperative control architecture where a designated autonomous agent (the leader) dictates a global reference trajectory, and the remaining agents (the followers) must synchronize their states to this trajectory. A key advantage of this architecture is that it does not require a centralized authority nor global all-to-all communication, allowing agents to coordinate scalably by exchanging information exclusively with a limited set of local neighbors.

Within this field, consensus protocols are generally classified based on the agents' intrinsic dynamics. *First-order consensus* typically refers to networks of agents modeled as single integrators or, through specific coordinate transformations, frameworks where the tracking problem is mathematically reduced to reaching an agreement on a static value (such as a projected initial state). Conversely, *higher-order consensus* addresses agents governed by more complex, generalized linear dynamics—such as double integrators or general state-space models—where the nodes must continuously synchronize a full, time-evolving state vector.

However, the heavy reliance on network connectivity exposes these cooperative control architectures to significant vulnerabilities: the injection of malicious data or the compromise of individual nodes (cyber-attacks) can propagate through the network in a cascading manner, preventing the achievement of global consensus and destabilizing the entire system dynamics. Guaranteeing the resilience of leader-follower tracking dynamics in the presence of adversarial agents represents,

therefore, one of the most critical challenges in modern control theory. While standard linear consensus protocols offer elegant and scalable synthesis methodologies, they lack mechanisms to correct tracking dynamics against potential attacks on communication channels. This renders the network structurally defenseless against constant additive attacks, necessitating the exploration of nonlinear filtering architectures and the optimization of network topologies.

1.2 Thesis Objectives

The primary goal of this thesis is to analyze novel approaches to address the security of cyber-physical systems described by higher-order leader-follower models and subjected to adversarial attacks. Starting from the distributed architecture proposed by Lewis et al.—which provides a highly effective and straightforward approach to synchronization—this research initially aims to evaluate the feasibility of error-correction strategies based on algorithmic observers, specifically Online Gradient Descent (OGD).

Having demonstrated the ineffectiveness of such approaches in the face of the recursive accumulation of attack signals, the focus shifts toward drawing inspiration from recent works on resilient network synchronization dynamics. Specifically, we integrate a continuous-time non-linear input filter, the Asymptotic Resilient Consensus Protocol (ARC-P), designed to identify and discard malicious information locally at each follower node.

The ultimate goal is twofold: first, to present a method for constructing network topologies that guarantee robustness against a specific number of attacks, minimizing communication overhead while maximizing tracking speed; second, to develop an enhanced, resilient, higher-order consensus framework capable of tolerating agents characterized by intrinsically unstable dynamics. This results in a robust tracking strategy that successfully preserves the design simplicity of the original architecture.

1.3 Achieved Results

Compared to the state-of-the-art analyzed in the literature, this thesis introduces two fundamental contributions:

1. **Synthesis of a scalable modular topology (DAG):** The analysis of standard robust architectures (fully connected networks and circulant graphs) revealed structural vulnerabilities related to information bottlenecks and topological inertia caused by backward-pointing edges. To overcome these limitations, a novel class of networks based on Directed Acyclic Graphs (DAGs)

was designed. This topology eliminates the cascading propagation of attacks, guarantees resilience under the most severe local threat model (the F-local malicious model), and maintains a local connectivity strictly bounded to $2F + 1$. Numerical simulations demonstrated that synchronization is highly achievable with this optimized network topology, drastically accelerating convergence and yielding a 271% performance improvement compared to traditional resilient circulant topologies, thereby minimizing communication overhead while maximizing tracking speed.

2. **Resilient consensus for unstable dynamics:** The reference framework for first-order resilient consensus (Scardovi [2]/LeBlanc[3]) imposes stringent restrictions, requiring a dynamic local controller and limiting its application to systems with at most marginally stable state matrices A. We propose the direct integration of the ARC-P filter within the static Lewis control protocol in the state space to achieve a higher-order resilient consensus. Although this approach involves an inevitable trade-off in convergence speed (due to the absence of a dynamic controller forcing a monotonic error decrease), it successfully removes the marginal stability constraint and preserves overall design simplicity. The results demonstrate the capability of the new framework to resiliently synchronize nodes characterized by strictly unstable and divergent dynamics (e.g., ramp dynamics driven by a double integrator) under uninformed attacks.

1.4 Thesis Structure

The thesis is organized as follows:

- **Chapter 1** introduces the discrete-time Leader-Follower network model. It establishes the theoretical foundations of distributed control and rigorously models the propagation of additive attacks. The chapter mathematically demonstrates the structural limitations of gradient-descent-based (OGD) observers in handling error accumulation, thereby motivating the shift toward continuous filtering strategies.

- **Chapter 2** analyzes the first-order resilient control framework based on modal projections and the ARC-P algorithm. An in-depth topological analysis of robustness conditions is conducted, highlighting the inertia constraints of circulant networks. Finally, the synthesis of the proposed acyclic modular topology is presented and validated.

- **Chapter 3** extends the concepts of resilience by formulating a higher-order consensus. The ARC-P filter is integrated into the continuous-time Lewis

framework to overcome the marginal stability limitation. The chapter concludes with a comparative performance evaluation and the simulative validation of the system applied to unstable dynamics subjected to various types of attacks.

Chapter 2

Leader-Follower Network Model

The network control system proposed by Lewis et al. in [1] provides a solid and mathematically rigorous foundation for the design of leader-follower dynamics in multi-agent systems. Its primary advantage resides in the node-level control synthesis: by leveraging Algebraic Riccati Equations (ARE), it allows for a fine-tuned balance between tracking performance and control effort, offering a straightforward methodology for distributed control design.

The main objective of this chapter is to thoroughly analyze this foundational framework and, most importantly, to rigorously mathematically model its vulnerabilities when subjected to malicious attacks on the communication channels. Specifically, we aim to demonstrate how constant additive attacks injected by compromised nodes propagate through the network, recursively altering the system's state and rendering standard error-correction mechanisms ineffective.

While the subsequent chapters of this thesis will address the problem of resilience in the *continuous-time* domain, this specific chapter adopts a *discrete-time* formulation. The rationale behind this modeling choice is deeply tied to our initial research objective: implementing an algorithmic error-correction strategy based on an Online Gradient Descent (OGD) global observer. Since optimization algorithms and algorithmic state estimators inherently operate iteratively on sampled data, evaluating the feasibility of an OGD-based resilient framework strictly required a discrete-time dynamic model.

However, as the mathematical derivations presented in the following sections will reveal, the recursive accumulation of the attack signals within the discrete-time closed-loop dynamics fundamentally violates the standard measurement models required by gradient-based observers. Proving this structural limitation is a core objective of this chapter, as it justifies the abandonment of the OGD approach

and mathematically motivates the shift toward the novel, continuous-time resilient filtering strategies proposed in the remainder of this thesis.

The chapter is organized as follows: Section 2.1 establishes the graph-theoretic foundations and details the synthesis of the distributed control and observer architectures according to Lewis et al. Section 2.2 introduces the threat model, injecting constant additive attacks into the communication channels of the various topological configurations. Finally, Section 2.3 derives the augmented error dynamics to model the sparse attack propagation, leading to Section 2.4 containig the concluding remarks on the limitations of OGD-based estimation in this specific adversarial scenario.

2.1 Distributed Control System Design

2.1.1 General Considerations

Before describing the distributed control system design we need to define what is an agent node and how it is related to the leader and to the overall topology.

Given a communication graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ that describes the topology of the agent interactions, where $\mathcal{V} = \{1,2,\dots,N\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges representing the communication links between them , let each of its N nodes be endowed with a state vector $x_i \in \mathbb{R}^n$, an output $y_i \in \mathbb{R}^p$ and a control input $u_i \in \mathbb{R}^m$. We consider at each node the discrete-time linear time-invariant dynamics

$$\begin{cases} x_i(k+1) = Ax_i(k) + Bu_i(k) \\ y_i(k) = Cx_i(k) \end{cases} \tag{2.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$. It is assumed that (A, B) is stabilizable, B has full column rank m, (A, C) is detectable, and C has full row rank p. The leader node 0 has the autonomous drift dynamics with no input

$$\begin{cases} x_0(k+1) = Ax_0(k) \\ y_0(k) = Cx_0(k) \end{cases} \tag{2.2}$$

The state, input and output are $x_i, x_0 \in \mathbb{R}^n$, $u_i \in \mathbb{R}^m$, $y_i, y_0 \in \mathbb{R}^p$. Note that A can be either be stable, marginally stable or even unstable, as long as (A, B) is stabilizable.

The leader is essentially an autonomous agent whose initial conditions are utilized to dictate a trajectory to the entire network.

The *synchronization* problem is to select the control signal u_i, using only measurements from the neighbors of node i, such that all nodes synchronize to the state of the leader, that is $\lim_{k\to\infty} \|x_i(k) - x_0(k)\|_2 = 0, \forall i$. These requirements should

be fulfilled for all initial conditions $x_i(0)$. In order to solve the synchronization problem in the general setting - specifically under the given LTI dynamics (2.1) and (2.2) and the structural assumptions of stabilizability and detectability - and where the agent nodes need some kind of observer to estimate their state variable, Lewis et al. [1] present three distinct configurations for the multi-agent system, ranging from a fully distributed architecture, where **both the controller and the observer** leverage neighborhood information for state tracking and control law computation (u_i), to hybrid solutions. Among the latter, they first propose a scheme in which agents estimate their states via **local observers** while maintaining a distributed control law; alternatively, they introduce a second hybrid approach that combines a distributed observer with a **local controller**.

2.1.2 Fundamental definitions

We first establish the graph-theoretic foundations required to design the control architectures. Given the communication graph $\mathcal{G}$, we define the following matrices:

- The **adjacency matrix** $E = [e_{ij}] \in \mathbb{R}^{N \times N}$ encodes the connectivity. A non-zero entry $e_{ij} > 0$ denotes the existence of a directed edge from node j to node i, implying that agent i receives information from neighbor j. Conversely, $e_{ij} = 0$ indicates no direct link.

- The **in-degree matrix** $D = \text{diag}(d_1, \dots, d_N) \in \mathbb{R}^{N \times N}$ is a diagonal matrix where each element $d_i = \sum_j e_{ij}$ represents the weighted sum of incoming connections for the i-th node.

- The **Laplacian matrix** $L = D - E$ models the information diffusion among agents. A crucial property of L is that its row sums are identically zero, which is fundamental for consensus protocols.

- The **pinning gain matrix** $G = \text{diag}(g_1, \dots, g_N) \in \mathbb{R}^{N \times N}$ characterizes the global coupling between the leader and the followers. Specifically, $g_i > 0$ if and only if the i-th agent has direct access to the leader's output.

Figure 2.1 illustrates a practical example of these topological definitions, highlighting the mapping between the network structure and its matrix representation.

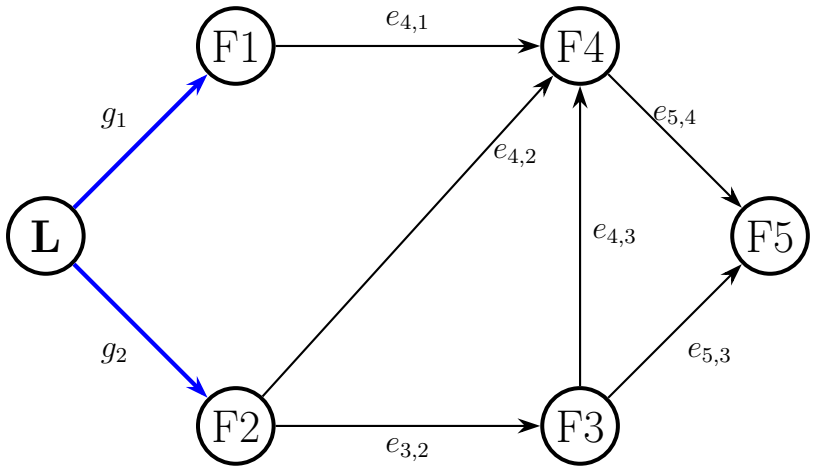


Figure 2.1: Example graph for Leader-Follower Dynamics.

$$E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & e_{3,2} & 0 & 0 & 0 \\ e_{4,1} & e_{4,2} & e_{4,3} & 0 & 0 \\ 0 & 0 & e_{5,3} & e_{5,4} & 0 \end{bmatrix}, \; D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e_{3,2} & 0 & 0 \\ 0 & 0 & 0 & e_{4,1}+e_{4,2}+e_{4,3} & 0 \\ 0 & 0 & 0 & 0 & e_{5,3}+e_{5,4} \end{bmatrix},$$

$$G = \begin{bmatrix} g_1 & 0 & 0 & 0 & 0 \\ 0 & g_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Kronecker product

The Kronecker product $\bigotimes$ of two matrices $[a_{ij}] = A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$ is the $mp \times nq$ block matrix defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \tag{2.3}$$

It will allow us to transition from the dynamics of a single agent to the global dynamics of an entire network. One of its most useful property is the **mixed Kronecker product** defined in this way:
let A,B,C and D matrices of dimensions such that the products $A \cdot C$ and $B \cdot D$ are well-defined. Then, the following identity holds:

$$(A \otimes B) \cdot (C \otimes D) = (A \cdot C) \otimes (B \cdot D) \tag{2.4}$$

2.1.3 Distributed Controller/Distributed Observer Model Design

In the fully distributed architecture, the observer model for the follower nodes is defined in this way:

$$\begin{cases} \hat{x}(k+1) = A\hat{x}(k) + B\hat{u}_i(k) - c_1(1+d_i+g_i)^{-1}F\varepsilon_i^o(k) \\ \hat{y}(k) = C\hat{x}(k) \end{cases} \tag{2.5}$$

where $\hat{u}_i$ is the input computed by looking to the observed state values and $F \in \mathbb{R}^{n\times p}$ is defined as the **observer gain matrix**.
The parameter $c_1 \in \mathbb{R}$ is the **coupling gain** related with the cooperative observer design. It is related to the topology and stabilization of the overall network dynamics. We will later discuss how to design it.
The leader node observer follows the dynamics:

$$\begin{cases} \hat{x}_0(k+1) = A\hat{x}(k) + F\tilde{y}_0 \\ \hat{y}_0(k) = C\hat{x}_0(k) \end{cases} \tag{2.6}$$

where $\tilde{y}_i = y_i - \hat{y}_i$ is the *local output estimation error*.

Assumption 2.1. $x(0)$ is known.

Then we can also assume $\hat{x}_0 = x_0$ and $\tilde{y}_0 = 0$.
To achieve synchronization, we define the **local neighborhood tracking errors**

$$\varepsilon_i = \sum_{j\in\mathcal{N}_i} e_{ij}(x_j - x_i) + g_i(x_0 - x_i) \tag{2.7}$$

and the **local neighborhood output disagreement**

$$\begin{aligned} \varepsilon_i^o &= \sum_{j\in\mathcal{N}_i} e_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(\tilde{y}_0 - \tilde{y}_i) \\ &= \sum_{j\in\mathcal{N}_i} e_{ij}(\tilde{y}_j - \tilde{y}_i) - g_i\tilde{y}_i \end{aligned} \tag{2.8}$$

where $\mathcal{N}_i = \{v_j : (v_j, v_i) \in \mathcal{E}\}$ set of in-neighbors of node v_i.
Since the state variables x_i are not assumed to be known, we need to modify the definition of the local neighborhood tracking error ε_i. We will consider its modified version $\hat{\varepsilon}_i$ instead:

$$\hat{\varepsilon}_i = \sum_{j\in\mathcal{N}_i} e_{ij}(\hat{x}_j - \hat{x}_i) + g_i(\hat{x}_0 - \hat{x}_i) \tag{2.9}$$

The input of agent i proposed by [1] is the *weighted local protocol*

$$u_i = c(1+d_i+g_i)^{-1}K\varepsilon_i \tag{2.10}$$

taking the multi agent systems dynamics to be

$$x_i(k+1) = Ax_i(k) + c(1+d_i+g_i)^{-1}BK\varepsilon_i(k) \tag{2.11}$$

where $K \in \mathbb{R}^{m\times n}$ is the **state variable feedback gain matrix** and c is the coupling gain related with the cooperative controller design.
By 'weighted', we mean that the protocol is multiplied by the weighting factor $(1+d_i+g_i)^{-1}$, which captures the local properties of the graph with respect to node i. As in the case of ε_i, we will use the alternate form of u_i

$$\hat{u}_i = c(1+d_i+g_i)^{-1}K\hat{\varepsilon}_i \tag{2.12}$$

Note that when we have the theoretical security that $\lim_{k\to\infty}(x-\hat{x}) = 0$, we can treat $\hat{u}$ as u, $\hat{x}$ as x and $\hat{\varepsilon}$ as ε in formulas.
At this stage, the local model of the leader node can be expressed as:

$$\begin{cases} x_0(k+1) = Ax_0(k) \\ \hat{x}_i(k+1) = A\hat{x}_0(k) + F\tilde{y}_0(k) \\ \quad y_0(k) = Cx_0(k) \\ \quad \hat{y}_0(k) = C\hat{x}_0(k) \end{cases} \tag{2.13}$$

Since assumption 2.1 holds, we can use the simplified version of the leader model

$$\begin{cases} x_0(k+1) = Ax_0(k) \\ \quad \tilde{y}_0 = 0 \end{cases} \tag{2.14}$$

Consequently, the local model of the i-th follower node can be expressed as:

$$\begin{cases} x_i(k+1) = Ax_i(k) + B\hat{u}_i(k) \\ \hat{x}_i(k+1) = A\hat{x}_i(k) + B\hat{u}_i(k) - c_1(1+d_i+g_i)^{-1}F\varepsilon_i^o(k) \\ \quad y_i(k) = Cx_i(k) \\ \quad \hat{y}_i(k) = C\hat{x}_i(k) \\ \quad \hat{\varepsilon}_i(k) = \sum_{j\in\mathcal{N}_i} e_{ij}(\hat{x}_j(k) - \hat{x}_i(k)) + g_i(x_0(k) - \hat{x}_i(k)) \\ \quad \varepsilon_i^o(k) = \sum_{j\in\mathcal{N}_i} e_{ij}(\tilde{y}_j(k) - \tilde{y}_i(k)) - g_i\,\tilde{y}_i(k) \\ \quad \hat{u}_i(k) = c(1+d_i+g_i)^{-1}K\hat{\varepsilon}_i(k) \end{cases} \tag{2.15}$$

where $x_i(0)$ is unknown and $\hat{x}_i(0)$ could be any value.
We choose $\hat{x}_i(0) = 0 \forall i \in \{1,\dots,N\}$.
We can note that in this setting, the nodes will share to the network their state estimate $\hat{x}_i$ and their output disagreement $\tilde{y}_i$.

At this point, the synchronization problem is solved by properly designing $\boldsymbol{c}$, $\boldsymbol{c}_1$, $\boldsymbol{K}$ and $\boldsymbol{F}$. Precisely, the design of c and K is the cooperative controller design while the design of c_1 and F is known as the cooperative observer design.
In order to design the coupling gains, we need to understand how these elements relate with the stability of the network dynamics. Therefore we define the global representations of state values, observer state values, output values, neighborhood tracking errors, neighborhood output disagreement and agents input respectively as:

$$\begin{aligned} x &= \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^{nN} \quad \hat{x} = \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_N \end{bmatrix} \in \mathbb{R}^{nN} \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^{pN} \quad \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{bmatrix} \in \mathbb{R}^{pN} \\ \hat{\varepsilon} &= \begin{bmatrix} \hat{\varepsilon}_1 \\ \vdots \\ \hat{\varepsilon}_N \end{bmatrix} \in \mathbb{R}^{nN} \quad \varepsilon^o = \begin{bmatrix} \varepsilon_1^o \\ \vdots \\ \varepsilon_N^o \end{bmatrix} \in \mathbb{R}^{pN} \quad \hat{u} = \begin{bmatrix} \hat{u}_1 \\ \vdots \\ \hat{u}_N \end{bmatrix} \in \mathbb{R}^{mN} \end{aligned} \tag{2.16}$$

We can, then, write the **global nodes dynamics**

$$x(k+1) = \begin{bmatrix} A & 0 & \dots & 0 \\ 0 & A & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A \end{bmatrix} x(k) + \begin{bmatrix} B & 0 & \dots & 0 \\ 0 & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B \end{bmatrix} \hat{u}(k) = \mathcal{M}_A x(k) + \mathcal{M}_B \hat{u}(k) \tag{2.17}$$

A clever way to rewrite 2.17 by highlighting the relationship with the matrices A and B is to employ the Kronecker product (2.3).
We can notice how $\mathcal{M}_A = \begin{bmatrix} A & 0 & \dots & 0 \\ 0 & A & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A \end{bmatrix}$ can be rewritten as $I_N \otimes A$ and $\mathcal{M}_B = \begin{bmatrix} B & 0 & \dots & 0 \\ 0 & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B \end{bmatrix}$ become $I_N \otimes B$.
Therefore, equation (2.17) becomes

$$x(k+1) = I_N \otimes A\, x(k) + I_N \otimes B\, \hat{u}(k) \tag{2.18}$$

Adopting the same approach, we can obtain the **global neighborhood tracking error** by considering

$$\hat{\varepsilon}_i = \sum_{j \in \mathcal{N}_i} e_{ij}(\hat{x}_j - \hat{x}_i) + g_i(x_0 - \hat{x}_i)$$

since if $(v_j, v_i) \notin \mathcal{E} \Rightarrow e_{ij} = 0$, we can write without loss of generality

$$\begin{aligned}\hat{\varepsilon}_i &= \sum_j e_{ij}(\hat{x}_j - \hat{x}_i) + g_i(x_0 - \hat{x}_i) = \sum_j e_{ij}\hat{x}_j - \left(\sum_j e_{ij} + g_i\right)\hat{x}_i + g_i x_0 \\ &= e_{i1}\hat{x}_1 + \ldots + e_{iN}\hat{x}_N - (d_i + g_i)\,\hat{x}_i + g_i x_0\end{aligned}$$

so, recalling the definition of Kronecker product (2.3)

$$\begin{aligned}\begin{bmatrix}\hat{\varepsilon}_1 \\ \vdots \\ \hat{\varepsilon}_N\end{bmatrix} &= \begin{bmatrix}e_{11}\hat{x}_1 + \ldots + e_{1N}\hat{x}_N \\ \vdots \\ e_{N1}\hat{x}_1 + \ldots + e_{NN}\hat{x}_N\end{bmatrix} - \begin{bmatrix}(d_1 + g_1)\hat{x}_1 \\ \vdots \\ (d_N + g_N)\hat{x}_N\end{bmatrix} + \begin{bmatrix}g_1\, x_0 \\ \vdots \\ g_N\, x_0\end{bmatrix} \\ &= \underbrace{\begin{bmatrix}e_{11}I_n & \ldots & e_{1N}I_n \\ \vdots & \ddots & \vdots \\ e_{N1}I_n & \ldots & e_{NN}I_n\end{bmatrix}}_{E\otimes I_n}\hat{x} - \underbrace{\begin{bmatrix}(d_1 + g_1)I_n & & 0 \\ & \ddots & \\ 0 & & (d_N + g_N)I_n\end{bmatrix}}_{(D+G)\otimes I_n}\hat{x} \\ &\quad + \underbrace{\begin{bmatrix}g_1 I_n & & 0 \\ & \ddots & \\ 0 & & g_N I_n\end{bmatrix}}_{G\otimes I_n}\begin{bmatrix}x_0 \\ x_0 \\ \vdots \\ x_0\end{bmatrix} \\ &= \underbrace{\begin{bmatrix}(e_{11} - d_1 - g_1)I_n & \ldots & e_{1N}I_n \\ \vdots & \ddots & \vdots \\ e_{N1}I_n & \ldots & (e_{NN} - d_N - g_N)I_n\end{bmatrix}}_{(E-D-G)\otimes I_n}\hat{x} + \underbrace{\begin{bmatrix}g_1 I_n & & 0 \\ & \ddots & \\ 0 & & g_N I_n\end{bmatrix}}_{G\otimes I_n}\begin{bmatrix}x_0 \\ x_0 \\ \vdots \\ x_0\end{bmatrix}\end{aligned}$$

By letting $\bar{x}_0 = \underline{1}_N \otimes x_0$, where $\underline{1}_N$ is the column vector of 1's, remembering the definitions of E, D, $L = D - E$ and G we can write the global neighborhood tracking error as

$$\begin{aligned}\hat{\varepsilon} &= (E - D - G) \otimes I_n\hat{x} + G \otimes I_n\bar{x}_0 \\ &= -(L + G) \otimes I_n\hat{x} + G \otimes I_n\bar{x}_0\end{aligned} \tag{2.19}$$

Equation 2.19 can be further simplified by considering that $L \cdot \underline{1}_N = 0$. Therefore, by using the mixed Kronecker product 2.4, we find that

$$\begin{aligned}&(L \otimes I_n) \cdot \bar{x}_0 = (L \otimes I_n) \cdot (\underline{1}_N \otimes x_0) = (L \cdot \underline{1}_N) \otimes (I_n \cdot x_0) = 0 \\ &\Rightarrow (L + G) \otimes I_n\, \bar{x}_0 = [L \otimes I_n + G \otimes I_n]\, \bar{x}_0 = L \otimes I_n\, \bar{x}_0 + G \otimes I_n\, \bar{x}_0 = G \otimes I_n\, \bar{x}_0 \\ &\Rightarrow G \otimes I_n\, \bar{x}_0 = (L + G) \otimes I_n\, \bar{x}_0\end{aligned} \tag{2.20}$$

Accordingly, we can rewrite 2.19 as

$$\hat{\varepsilon} = -(L+G) \otimes I_n \, (\hat{x} - \bar{x}_0) = -(L+G) \otimes (\hat{x} - \bar{x}_0) \tag{2.21}$$

Extending the same procedure to the global neighborhood output disagreement ε^o(2.8), we can find that

$$\begin{aligned} \varepsilon^o &= -(L+G) \otimes \tilde{y} = -(L+G) \otimes (y - \hat{y}) \\ &= -(L+G) \otimes y + (L+G) \otimes \hat{x} \end{aligned} \tag{2.22}$$

Now, from(2.12), we can write

$$\begin{aligned} \hat{u} = c \begin{bmatrix} \hat{u}_1 \\ \vdots \\ \hat{u}_N \end{bmatrix} &= c \begin{bmatrix} (1+d_1+g_1)^{-1} K \cdot \hat{\varepsilon}_1 \\ \vdots \\ (1+d_N+g_N)^{-1} K \cdot \hat{\varepsilon}_N \end{bmatrix} \\ &= c \begin{bmatrix} (1+d_1+g_1)^{-1} K & & 0 \\ & \ddots & \\ 0 & & (1+d_N+g_N)^{-1} K \end{bmatrix} \hat{\varepsilon} \end{aligned}$$

Since $I_N + D + G$ is a diagonal matrix, its inverse $(I_N + D + G)^{-1}$ is still a diagonal matrix defined as $\text{diag}(\frac{1}{1+d_1+g_1}, \ldots, \frac{1}{1+d_N+g_N})$. We can modify the above equation writing

$$\begin{aligned} \hat{u} &= c \left[(I_N + D + G)^{-1} \otimes K \right] \cdot \hat{\varepsilon} \\ &= -c \left[(I_N + D + G)^{-1} \otimes K \right] \cdot [(L+G) \otimes (\hat{x} - \bar{x}_0)] \\ &\text{here we use the mixed Kronecker product (2.4)} \\ &= -c \left[(I_N + D + G)^{-1} \cdot (L+G) \right] \otimes [K \cdot (\hat{x} - \bar{x}_0)] \end{aligned}$$

We refer to the matrix

$$\Gamma = (I + D + G)^{-1}(L+G), \; \Gamma \in \mathbb{R}^{N \times N} \tag{2.23}$$

as the (weighted) **graph matrix**. Its eigenvalues $\Lambda_k, \; k = 1, \ldots, N$ are important for the control system design, and we refer to them as the *graph matrix eigenvalues*. At this point, we can write

$$\hat{u} = -c\Gamma \otimes K(\hat{x} - \bar{x}_0) \tag{2.24}$$

Following a procedure analogous to the derivation of (2.24), the global mapping of

$-c(1+d_i+g_i)^{-1}F\varepsilon_i^o$ in the observer dynamics (2.5) is given by

$$\begin{aligned}
&-c\begin{bmatrix}(1+d_1+g_1)^{-1}F\cdot\varepsilon_1^o\\ \vdots\\ (1+d_N+g_N)^{-1}F\cdot\varepsilon_N^o\end{bmatrix}\\
&=-c\begin{bmatrix}(1+d_1+g_1)^{-1}F & & 0\\ & \ddots & \\ 0 & & (1+d_N+g_N)^{-1}F\end{bmatrix}\varepsilon^o\\
&=-c\Big[(I_N+D+G)^{-1}\otimes F\Big]\cdot\varepsilon^o=c\Big[(I_N+D+G)^{-1}\otimes F\Big]\cdot[(L+G)\otimes\tilde{y}]\\
&=c\Big[(I_N+D+G)^{-1}\cdot(L+G)\Big]\otimes[F\cdot\tilde{y}]=c\Gamma\otimes F\tilde{y}
\end{aligned}\tag{2.25}$$

By combining all the global representations, we conclude that the global dynamics of the leader node is

$$\bar{x}_0(k+1)=I_N\otimes A\,\bar{x}_0(k)\tag{2.26}$$

the global observers dynamics is

$$\begin{aligned}
\hat{x}(k+1)&=I_N\otimes A\,\hat{x}(k)+I_N\otimes B\,\hat{u}(k)+c_1\Gamma\otimes F\,\tilde{y}(k)\\
&=I_N\otimes A\,\hat{x}(k)-c\,(I_N\otimes B)\cdot(\Gamma\otimes K)\cdot(\hat{x}-\bar{x}_0)(k)+c_1\Gamma\otimes F\,\tilde{y}(k)\\
&\quad\text{using the mixed Kronecker product(2.4)}\\
&=I_N\otimes A\,\hat{x}(k)-c\,[(I_N\cdot\Gamma)\otimes(B\cdot K)]\cdot(\hat{x}-\bar{x}_0)(k)+c_1\Gamma\otimes F\,\tilde{y}(k)\\
&=I_N\otimes A\,\hat{x}(k)-c\Gamma\otimes BK\,(\hat{x}-\bar{x}_0)(k)+c_1\Gamma\otimes F\,\tilde{y}(k)\\
&=I_N\otimes A\,\hat{x}(k)-c\Gamma\otimes BK\,(\hat{x}-\bar{x}_0)(k)+c_1\Gamma\otimes F\,(y-\hat{y})(k)\\
&=I_N\otimes A\,\hat{x}(k)-c\Gamma\otimes BK\,(\hat{x}-\bar{x}_0)(k)+c_1\Gamma\otimes F\,y(k)-c_1\Gamma\otimes F\,\hat{y}(k)\\
&=I_N\otimes A\,\hat{x}(k)-c\Gamma\otimes BK\,(\hat{x}-\bar{x}_0)(k)+c_1\Gamma\otimes F\,y(k)-c_1\Gamma\otimes F\,C\hat{x}(k)\\
&=[I_N\otimes A-c_1\Gamma\otimes FC]\,\hat{x}(k)-c\Gamma\otimes BK(\hat{x}-\bar{x}_0)(k)+c_1\Gamma\otimes F\,y(k)
\end{aligned}\tag{2.27}$$

Finally, the global states dynamics is

$$\begin{aligned}
x(k+1)&=I_N\otimes A\,x(k)+I_N\otimes B\,\hat{u}(k)\\
&=I_N\otimes A\,x(k)-c\,[I_N\otimes B]\cdot[\Gamma\otimes K]\,(\hat{x}-\bar{x}_o)(k)\\
&\quad\text{using the mixed Kronecker product(2.4)}\\
&=I_N\otimes A\,x(k)-c\,[I_N\cdot\Gamma]\otimes[B\cdot K]\,(\hat{x}-\bar{x}_o)(k)\\
&=I_N\otimes A\,x(k)-c\Gamma\otimes BK(\hat{x}-\bar{x}_o)(k)
\end{aligned}\tag{2.28}$$

We now define the **global disagreement error** $\delta=x-\bar{x}_0\in\mathbb{R}^{nN}$ and the **global observer error** $\eta=x-\hat{x}\in\mathbb{R}^{nN}$.
We say that the synchronization problem is solved if for a suitable choice of c, K,

c_1 and F we obtain

$$\lim_{k\to\infty} \delta(k) = 0 \tag{2.29}$$

To determine these suitable parameters, we have to analyze the global system dynamics involving both δ and η:

$$\begin{aligned}
\delta(k+1) &= x(k+1) - \bar{x}_0(k+1) \\
&= I_N \otimes A\, x(k) - c\Gamma \otimes BK(\hat{x} - \bar{x}_o)(k) - I_N \otimes A\, \bar{x}_0(k) \\
&= I_N \otimes A\, x(k) - c\Gamma \otimes BK(\hat{x} - x + x - \bar{x}_o)(k) - I_N \otimes A\, \bar{x}_0(k) \\
&= I_N \otimes A\, x(k) - c\Gamma \otimes BK(x - \bar{x}_0)(k) - I_n \otimes A\, \bar{x}_0(k) \\
&\quad + c\Gamma \otimes BK(x - \hat{x})(k) \\
&= [I_n \otimes A - c\Gamma \otimes BK]\,(x - \hat{x}_0)(k) + c\Gamma \otimes BK(x - \hat{x})(k) \\
&= [I_n \otimes A - c\Gamma \otimes BK]\,\delta(k) + c\Gamma \otimes BK\eta(k)
\end{aligned} \tag{2.30}$$

$$\begin{aligned}
\eta(k+1) &= x(k+1) - \hat{x}(k+1) \\
&= I_N \otimes A\, x(k) + I_N \otimes B\, \hat{u}(k) \\
&\quad - [I_N \otimes A\, \hat{x}(k) + I_N \otimes B\, \hat{u}(k) + c\Gamma \otimes F\, \tilde{y}(k)] \\
&= I_N \otimes A\, x(k) + I_N \otimes B\, \hat{u}(k) \\
&\quad - [I_N \otimes A\, \hat{x}(k) + I_N \otimes B\, \hat{u}(k) + c\Gamma \otimes FC\, (x - \hat{x})(k)] \\
&= [I_N \otimes A - c\Gamma \otimes FC]\,(x - \hat{x})(k) \\
&= [I_N \otimes A - c\Gamma \otimes FC]\,\eta(k)
\end{aligned} \tag{2.31}$$

We define the **global disagreement error system matrix** as

$$A_c \in \mathbb{R}^{nN\times nN} = I_n \otimes A - c\Gamma \otimes BK = I_n \otimes A - c(I + D + G)^{-1}(L + G) \otimes BK \tag{2.32}$$

which reflects the local agent closed-loop matrix $A - BK$ as modified on the graph structure $(L + G)$;
We will also refer to matrix $B_c \in \mathbb{R}^{nN\times nN}$ as $B_c = c\Gamma \otimes FC$.
The **global observer system matrix** is defined as

$$A_o \in \mathbb{R}^{nN\times nN} = I_N \otimes A - c\Gamma \otimes FC = I_N \otimes A - c(I + D + G)^{-1}(L + G) \otimes FC \tag{2.33}$$

Therefore, we can represent the **closed loop error system** as

$$\begin{bmatrix} \delta \\ \eta \end{bmatrix}(k+1) = \begin{bmatrix} A_c & B_c \\ 0 & A_o \end{bmatrix} \begin{bmatrix} \delta \\ \eta \end{bmatrix}(k) \tag{2.34}$$

and its stability is related with the stability of A_c and A_o. We notice how the cooperative controller design is performed by selecting c and K which will make A_c asymptotically stable and the cooperative observer design is performed by selecting c_1 and F which will make A_o asymptotically stable.

Cooperative Controller Design and Cooperative Observer Design

The design of the cooperative controller parameters relies on the solution of an Algebraic Riccati Equation (ARE), as stated in the following theorem:

Theorem 2.1 (Theorem 4.1 of [1])**.** *Suppose (A, B) is stabilizable. Assume that the interaction graph contains a spanning tree with at least one pinning gain nonzero that connects into the root node. Let $P > 0$ be a solution of the discrete-time Riccati-like equation*

$$A^T PA - P + Q - A^T PB(B^T PB)^{-1} B^T PA = 0 \tag{2.35}$$

for some prescribed $Q = Q^T > 0$. Define

$$r := \left[\sigma_{\max}\left(Q^{-1/2} A^T PB \left(B^T PB\right)^{-1} B^T PAQ^{-1/2}\right)\right]^{-1/2} \tag{2.36}$$

Then protocol (2.10) guarantees synchronization of multi-agent systems (2.11) for some K if there exists a covering circle $C(c_0, r_0)$ of the graph matrix eigenvalues $\Lambda_k,\ k = 1, \ldots N$ such that

$$\frac{r_0}{c_0} < r \tag{2.37}$$

Moreover, if condition (2.37) is satisfied then the choice of feedback matrix

$$K = \left(B^T PB\right)^{-1} B^T PA \tag{2.38}$$

and coupling gain

$$c = \frac{1}{c_0} \tag{2.39}$$

guarantee synchronization [1].

Theorem 2.1 presents a straightforward method to find the parameters K and c. A key advantage of this formulation is the structural separation between the local control design and the network topology. The feedback matrix K depends solely on the agent dynamics through the discrete-time ARE (2.35), while the coupling gain c depends exclusively on the graph eigenvalues (2.39). The interaction between these two independent design steps is entirely captured by the geometric inequality (2.37), whose satisfaction guarantees global synchronization.
Therefore, the coupling gain c is uniquely related to the graph matrix eigenvalues and the synchronization is guaranteed by the proper selection of Q. This aspect will return when we will see theorem 2.2 which is related to the cooperative observer design.

In order to solve the cooperative controller design, we can exploit the following theorem:

Theorem 2.2 (Theorem 4.2 of [1]). *Given multi-agent systems (2.1) with (A,C) detectable, assume the interaction graph contains a spanning tree with at least one pinning gain non-zero that connects into the root node. Let $P > 0$ be a solution of the discrete-time observer Riccati equation*

$$APA^T - P + Q - APC^T \left(CPC^T\right)^{-1} CPA^T = 0 \tag{2.40}$$

where $Q = Q^T > 0$. Choose the observer gain matrix as

$$F = APC^T \left(CPC^T\right)^{-1} \tag{2.41}$$

Define

$$r_{obs} := \left[\sigma_{\max}\left(Q^{-1/2} APC^T \left(CPC^T\right)^{-1} CPA^T Q^{-1/2}\right)\right]^{-1/2} \tag{2.42}$$

Then the observer error dynamics (2.31) are stable if there exists a covering circle $\bar{C}(c_0, r_0)$ of the graph matrix eigenvalues Λ_k, $k = 1 \dots N$ such that

$$\frac{r_0}{c_0} < r_{obs} \tag{2.43}$$

If (2.43) is satisfied then taking the coupling gain

$$c_1 = \frac{1}{c_0} \tag{2.44}$$

makes the observer dynamics (2.31) stable [1].

The above theorem presents a similar method of theorem 2.1 to solve the cooperative observer design. We can notice how following the procedure in theorem 2.2 will held to obtain $c_1 = \frac{1}{c_0}$. Therefore, assuming that the cooperative controller design is made by following the procedure in theorem 2.1 and that the cooperative observer design is made by following the procedure in theorem 2.2 and assuming that both problems are solved for the same Γ, if we use the same method to find c and c_1, we can impose

$$c = c_1 = \frac{1}{c_0} \tag{2.45}$$

A general way to satisfy both (2.37) and (2.43) is to find c_0 and r_0 which minimize $\frac{r_0}{c_0}$. However, the book [1] propose a straightforward solution to this problem only in case Γ has all real and positive eigenvalues:

Corollary 2.3 (Corollary 4.1 of [1]). *Let graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ have a spanning tree with pinning into the root node, and all eigenvalues of Γ be real and positive, that is $\Lambda_k > 0$ for all k.*

Define $0 < \Lambda_{\min} \leq \ldots \leq \Lambda_k \leq \ldots \leq \Lambda_{\max}$. *A converging circle of* Λ_k *that also minimizes* r_0/c_0, *is* $C(c_0, r_0)$ *with*

$$\frac{r_0}{c_0} = \frac{\Lambda_{\max} - \Lambda_{\min}}{\Lambda_{\max} + \Lambda_{\min}} \tag{2.46}$$

Then, condition (2.37) in Theorem 2.1 becomes

$$\frac{\Lambda_{\max} - \Lambda_{\min}}{\Lambda_{\max} + \Lambda_{\min}} < r \tag{2.47}$$

and condition (2.43) in Theorem 2.2 becomes

$$\frac{\Lambda_{\max} - \Lambda_{\min}}{\Lambda_{\max} + \Lambda_{\min}} < r_{obs}. \tag{2.48}$$

Moreover, given that this condition is satisfied, the coupling gain choices (2.39) and (2.44) reduce to

$$c = c_1 = \frac{2}{\Lambda_{\max} + \Lambda_{\min}}. \tag{2.49}$$

A crucial aspect is how the AREs in Theorems 2.1 and 2.2 allow us to decouple the design of the local feedback protocol from the details of the communication graph structure, which are managed by the coupling gain c.
From a geometric perspective, solving the local ARE provides a guaranteed *synchronization region*—specifically, a stability circle $\mathcal{C}(1, r)$ in the complex plane. Consequently, the cooperative control problem is reduced to a simple geometric condition: ensuring that the region covering the eigenvalues of the graph matrix can be mapped inside this guaranteed stability circle via the coupling gain.
This design approach inherently confers **robustness** to the system. If the conditions of the theorems are met, there exists an open interval of admissible values for the coupling gain c. This implies that synchronization is maintained even in the presence of small perturbations or variations in c, provided the gain remains within this interval.
Finally, since the design relies on local agent dynamics, the proposed method is **highly scalable**. The dimension of the Riccati equation corresponds to the state dimension n of a single agent and is independent of the total number of nodes N in the graph. This allows the same design to be applied regardless of the network size.

2.1.4 Distributed Controller and Local Observer

This setting is analogous to the fully distributed one, except for the local observer dynamics formulation which becomes

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + B\hat{u}(k) + F\tilde{y}_i(k) \tag{2.50}$$

and its global representation becomes

$$\begin{aligned}\hat{x}(k+1) &= I_N \otimes A\ \hat{x}(k) + I_N \otimes B\ \hat{u}(k) + I_N \otimes F\ \tilde{y}(k) \\ &= \ldots = I_N \otimes A\ \hat{x}(k) - c\Gamma \otimes BK(\hat{x} - \bar{x}_0)(k) + I_N \otimes F\ (y - \hat{y}) \\ &= I_N \otimes A\ \hat{x}(k) - c\Gamma \otimes BK(\hat{x} - \bar{x}_0)(k) + I_N \otimes F\ y(k) - I_N \otimes F\ \hat{y}(k) \\ &= I_N \otimes A\ \hat{x}(k) - c\Gamma \otimes BK(\hat{x} - \bar{x}_0)(k) + I_N \otimes F\ y(k) - I_N \otimes FC\ \hat{x}(k) \\ &= [I_N \otimes (A - FC)]\ \hat{x}(k) - c\Gamma \otimes BK(\hat{x} - \bar{x}_0)(k) + I_N \otimes FC\ y(k)\end{aligned} \tag{2.51}$$

and the information shared to the network by a node is just its state estimate $\hat{x}_i$. The closed-loop local observer error dynamics η_i becomes

$$\begin{aligned}\eta_i(k+1) &= x_i(k+1) - \hat{x}_i(k+1) \\ &= Ax_i(k) + B\hat{u}_i(k) - [A\hat{x}(k) + B\hat{u}_i(k) + F\tilde{y}_i(k)] \\ &= Ax_i(k) + B\hat{u}_i(k) - A\hat{x}_i(k) - B\hat{u}_i(k) - F\tilde{y}_i(k) \\ &= Ax_i(k) - A\hat{x}_i(k) - FC(x_i - \hat{x}_i)(k) \\ &= (A - FC)(x_i - \hat{x}_i)(k) = (A - FC)\eta_i(k)\end{aligned} \tag{2.52}$$

and its global representation is

$$\eta(k+1) = I_N \otimes (A - FC)\ \eta(k) \tag{2.53}$$

Since the global disagreement error dynamics $\delta(k+1) = x(k=1) - \bar{x}_0(k+1)$ remains the one defined in (2.30), the closed loop error system changes in

$$\begin{bmatrix}\delta \\ \eta\end{bmatrix}(k+1) = \begin{bmatrix}A_c & B_c \\ 0 & I_N \otimes (A - FC)\end{bmatrix}\begin{bmatrix}\delta \\ \eta\end{bmatrix}(k). \tag{2.54}$$

We can see how the stability of (2.54) is determined by the stability of A_c and $I_N \otimes (A - FC)$. Therefore, in order to guarantee $\lim_{k\to\infty} \delta(k) = 0$, we need to solve a cooperative controller design (already discussed in theorem 2.1) and to find a F which stabilize $I_N \otimes (A - FC)$.
To solve the latter problem, [1] propose to simply select F using, for instance, Riccati design so that $(A - FC)$ is asymptotically stable.

2.1.5 Local Controller and Distributed Observer

The last setting we analyze still remains similar to the full distributed one, except for the fact that the controller is local of the form

$$\hat{u}_i = -K\hat{x}_i \tag{2.55}$$

and the local neighborhood output disagreement changes its definition to

$$\varepsilon_i^o = \sum_{j \in \mathcal{N}_i} e_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(\boldsymbol{y}_0 - \tilde{y}_i) \tag{2.56}$$

Note that instead of $\tilde{y}_0$ in (2.8), which is assumed identically equal to zero (2.1), here one uses the control node output y_0. In global form this yields

$$\begin{aligned} x(k+1) &= I_N \otimes A\, x(k) - I_N \otimes BK\, \hat{x}(k) \\ \hat{x}(k+1) &= I_N \otimes (A - BK)\, \hat{x}(k) + c_1\Gamma \otimes FC\, \eta(k) - c_1\Gamma \otimes FC\, \bar{x}_0(k) \end{aligned}. \tag{2.57}$$

Expressing x, $\hat{x}$ through x, $\eta = x - \hat{x}$ gives

$$\begin{aligned} x(k+1) &= I_N \otimes (A - BK)\, x(k) + I_N \otimes BK\, \eta(k) \\ \eta(k+1) &= I_N \otimes (A - BK)\, x(k) + I_n \otimes BK\, \eta(k) \\ &\quad - I_N \otimes (A - BK)\hat{x}(k) - c_1\Gamma \otimes\ FC\eta(k) + c_1\Gamma \otimes FC\, \bar{x}_0(k) \\ &= I_N \otimes (A - BK)\eta(k) + I_N \otimes BK\, \eta(k) - c_1\Gamma \otimes FC(\eta - \bar{x}_0)(k) \\ &= (I_N \otimes A - c_1\Gamma \otimes FC)\eta(k) + c_1\Gamma \otimes FC\, \bar{x}_0(k) \\ &= A_o\eta(k) + c_1\Gamma \otimes FC\bar{x}_0(k) \end{aligned}. \tag{2.58}$$

Global tracking error dynamics now follows as

$$\delta(k+1) = (I_N \otimes A)\delta(k) - (I_N \otimes BK)\hat{x}(k). \tag{2.59}$$

Using that $\hat{x} = x - \eta = x - \bar{x}_0 - \eta + \bar{x}_0 = \delta - (\eta - \bar{x}_0)$ one finds

$$\delta(k+1) = I_N \otimes (A - BK)\delta(k) + (I_N \otimes BK)(\eta(k) - \bar{x}_0(k)). \tag{2.60}$$

We ca see how the global error system is not an autonomous one since an exogenous input is present in form of control node state $\bar{x}_0$. However if one looks at the dynamics of $\vartheta(k) = \eta(k) - \bar{x}_0(k)$ it follows that $\vartheta(k+1) = A_o\vartheta(k)$, and $\delta(k+1) = (I_N \otimes (A - BK)\delta(k) + (I_N \otimes BK)\vartheta(k)$. Or, mode clearly written in matrix form

$$\begin{bmatrix} \delta \\ \vartheta \end{bmatrix}(k+1) = \begin{bmatrix} I_N \otimes (A - BK) & I_N \otimes BK \\ 0 & A_o \end{bmatrix} \begin{bmatrix} \delta \\ \vartheta \end{bmatrix}(k) \tag{2.61}$$

The control gain K is simply selected using, *e.g.* Riccati design, so that $(A - BK)$ is asymptotically stable. The observer gain is selected by Theorem 2.2. Then, under the hypotheses of Theorem 2.2, synchronization $\delta \to 0$ is guaranteed.
Note that the observer in (2.57) is biased since $\vartheta \to 0$ implies $\eta \to \bar{x}_0$. Thus, the observers effectively estimate the tracking errors, converging to $\hat{x}_i(k) = x_i(k) - x_0(k) = \delta_i(k)$.

At this point, a quick analysis of the systems described in the previous paragraphs shows how the three proposed observer-controller architectures differ significantly in terms of information exchange and computational effort.
The Neighborhood Controller and Neighborhood Observer (fully distributed) architecture requires a significant amount of computation and communication. Specifically, to produce an estimate of its own state, each agent must measure the outputs of its neighbors, and state or output estimates must be exchanged between neighbors. Furthermore, the calculation of the control input necessitates access to the estimated states of all neighboring agents.
In contrast, the Neighborhood Controller and Local Observer architecture entails a reduced computational and communication burden compared to the fully distributed approach. Since the observer is local, each agent estimates its state using solely its own inputs and outputs. Consequently, only the state estimates need to be transmitted between neighbors to facilitate the distributed control law.
Finally, the Local Controller and Neighborhood Observer architecture presents a distinct scenario where, for control purposes, the agents operate without needing to communicate with their neighborhood. However, communication remains a requisite for the distributed estimation process within the neighborhood observer.

2.2 Constant Attacks on Communication Channels

In this section, we analyze the impact of additive attacks targeting the communication channels of both the followers and the leader node. Specifically, we model the attack as an injection of malicious signals into the transmitted information rather than on the intrinsic system output y. Consequently, an attack on a specific node does not directly alter its own dynamics but propagates to the neighbors receiving its corrupted data.

Figure 2.2 depicts a representative attack scenario, which can be compared with the nominal topology in Figure 2.1.

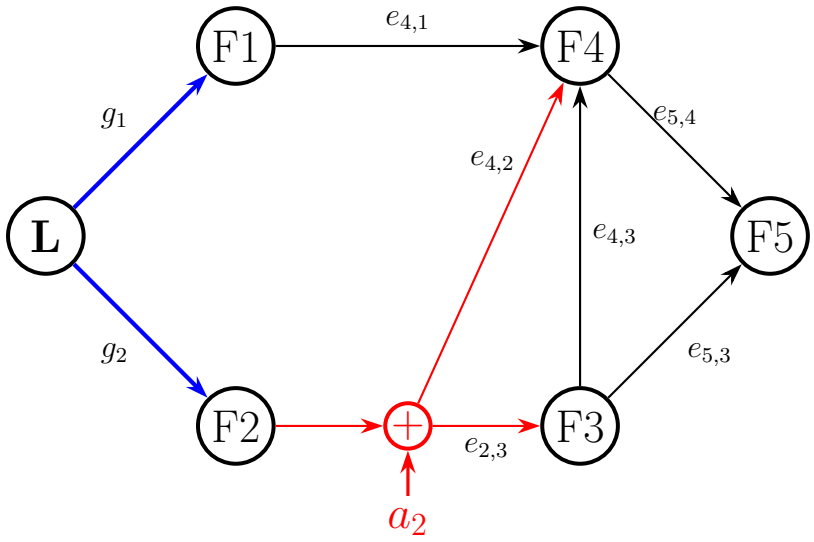


Figure 2.2: Example attacked graph for Leader-Follower Dynamics.

For the sake of simplicity, we assume a constant attack model where a scalar value a_j is added to every entry of the j-th node's output vector. We define the sparse attack vector affecting the agents' communication channels as $a = [a_1, \ldots, a_N]^T \in \mathbb{R}^N$, and the attack on the leader's channel as the scalar $a_0 \in \mathbb{R}$.

2.2.1 Attack on the Fully Distributed Model

In this setting the node output vector is $\begin{bmatrix} \hat{x}_j \\ \tilde{y}_j \end{bmatrix}$ and the leader output vector is just x_0. After the attack, the information transferred through the network by a follower becomes $\begin{bmatrix} \hat{x}'_j \\ \tilde{y}'_j \end{bmatrix} = \begin{bmatrix} \hat{x}_j \\ \tilde{y}_j \end{bmatrix} + \begin{bmatrix} a_j \underline{1}_n \\ a_j \underline{1}_p \end{bmatrix}$ and the information shared by the leader becomes $x'_0 = x_0 + \underline{1}_n a_0$.
We define the local attack notations

$$\begin{aligned} a_{x_i} &= a_i \underline{1}_n,\ a_{x_i} \in \mathbb{R}^n \\ a_{y_i} &= a_i \underline{1}_p,\ a_{y_i} \in \mathbb{R}^n \end{aligned} \tag{2.62}$$

and the global attack notations

$$a_x = \begin{bmatrix} a_1\underline{1}_n \\ a_2\underline{1}_n \\ \vdots \\ a_N\underline{1}_n \end{bmatrix} = \begin{bmatrix} a_{x_1} \\ a_{x_2} \\ \vdots \\ a_{x_N} \end{bmatrix} = a \otimes \underline{1}_n \in \mathbb{R}^{nN}, \quad a_y = \begin{bmatrix} a_1\underline{1}_p \\ a_2\underline{1}_p \\ \vdots \\ a_N\underline{1}_p \end{bmatrix} = \begin{bmatrix} a_{y_1} \\ a_{y_2} \\ \vdots \\ a_{y_N} \end{bmatrix} = a \otimes \underline{1}_p \in \mathbb{R}^{pN}$$
$$\bar{a}_{x_0} = \begin{bmatrix} a_{x_0} \\ a_{x_0} \\ \vdots \\ a_{x_0} \end{bmatrix} = a_0 \cdot \underline{1}_{nN} \in \mathbb{R}^{nN}, \quad \bar{a}_{y_0} = \begin{bmatrix} a_{y_0} \\ a_{y_0} \\ \vdots \\ a_{y_0} \end{bmatrix} = a_0 \cdot \underline{1}_{pN} \in \mathbb{R}^{pN} \tag{2.63}$$

It follows the analysis on how the nodes dynamics changes because of the attacks.

Attack on $\hat{x}_j$

Attack on $\hat{x}_j$ will affect the i-th agent dynamics by altering its input

$$\hat{u}_i \to \hat{u}'_i = c(1 + d_i + g_i)^{-1} K \hat{\varepsilon}'_i \tag{2.64}$$

where

$$\begin{aligned} \hat{\varepsilon}'_i &= \sum_{j \in \mathcal{N}_i} e_{ij}(\hat{x}'_j - \hat{x}_i) + g_i(x'_0 - \hat{x}_i) \\ &= \sum_{j \in \mathcal{N}_i} e_{ij}(\hat{x}_j + a_{x_{j\neq i}} - \hat{x}_i) + g_i(x_0 + a_{x_0} - \hat{x}_i) \\ &= \sum_{j \in \mathcal{N}_i} e_{ij}(\hat{x}_j - \hat{x}_i) + g_i(x_0 - \hat{x}_i) + \sum_{j \in \mathcal{N}_i} e_{ij} a_{x_{j\neq i}} + g_i a_{x_0} \\ &= \hat{\varepsilon}_i + \sum_{j \in \mathcal{N}_i} e_{ij} a_{x_{j\neq i}} + g_i a_{x_0} \end{aligned} \tag{2.65}$$

By defining $E' = E - \text{diag}(E)$, $E' \in \mathbb{R}^{N \times N}$, the distributed description of $\hat{\varepsilon}'$ is

$$\begin{aligned} \hat{\varepsilon}' &= \hat{\varepsilon} + (E' \otimes I_n)a_x + (G \otimes I_n)\bar{a}_{x_0} \\ &\quad \text{for the same reason in (2.20)} \\ &= \hat{\varepsilon} + (E' \otimes I_n)a_x + ((L + G) \otimes I_n)\bar{a}_{x_0} \\ &\quad \text{by assuming no self loop in the graph } (\text{diag}(E) = 0 \Rightarrow E' = E) \\ &= \hat{\varepsilon} + (E \otimes I_n)a_x + ((L + G) \otimes I_n)\bar{a}_{x_0} \end{aligned} \tag{2.66}$$

Therefore the global description of the input values become

$$\begin{aligned} \hat{u}' &= c\left[(I_N + D + G)^{-1} \otimes K\right] \cdot \hat{\varepsilon}' \\ &= c\left[(I_N + D + G)^{-1} \otimes K\right] \cdot [\hat{\varepsilon} + (E \otimes I_n)a_x + ((L + G) \otimes I_n)\bar{a}_{x_0}] \\ &= \hat{u} + c\left[(I_N + D + G)^{-1} \otimes K\right] \cdot [(E \otimes I_n)a_x + ((L + G) \otimes I_n)\bar{a}_{x_0}] \end{aligned} \tag{2.67}$$

Attack on $\tilde{y}_j$

Attack on $\tilde{y}_j$ will affect the i-th neighborhood observer dynamics by altering $\varepsilon^o \to \varepsilon^{o'}$ where

$$\begin{aligned}\varepsilon_i^{o'} &= \sum_{j\in\mathcal{N}_i} e_{ij}(\tilde{y}_j' - \tilde{y}_i) - g_i\,\tilde{y}_i \\ &= \sum_{j\in\mathcal{N}_i} e_{ij}(\tilde{y}_j + a_{y_{j\neq i}} - \tilde{y}_i) - g_i\,\tilde{y}_i \\ &= \sum_{j\in\mathcal{N}_i} e_{ij}(\tilde{y}_j - \tilde{y}_i) - g_i\,\tilde{y}_i + \sum_{j\in\mathcal{N}_i} e_{ij} a_{y_{j\neq i}}\end{aligned} \quad . \tag{2.68}$$

The attacked global neighborhood output disagreement is

$$\varepsilon^{o'} = \varepsilon^o + (E\otimes I_p)a_y \tag{2.69}$$

Attack Effects

The states' global dynamics becomes

$$\begin{aligned}x(k+1) &= I_N\otimes A\,x(k) + I_N\otimes B\,\hat{u}'(k) \\ &= I_N\otimes A\,x(k) + c(I_N+D+G)^{-1}\otimes BK\hat{\varepsilon}'(k) \\ &= I_N\otimes A\,x(k) \\ &\quad + c(I_N+D+G)^{-1}\otimes BK\,[\hat{\varepsilon} + (E\otimes I_n)a_x + ((L+G)\otimes I_n)\bar{a}_{x_0}] \\ &= I_N\otimes A\,x(k) + c(I_N+D+G)^{-1}\otimes BK\,\hat{\varepsilon} \\ &\quad + c(I_N+D+G)^{-1}\otimes BK\,[(E\otimes I_n)a_x + ((L+G)\otimes I_n)\bar{a}_{x_0}]\end{aligned}$$

from (2.66)

$$\begin{aligned}&= I_N\otimes A\,x(k) - c\Gamma\otimes BK(\hat{x}-\bar{x}_0)(k) \\ &\quad + c(I_N+D+G)^{-1}E\otimes BKa_x + c\Gamma\otimes BK\bar{a}_{x_0} \\ &= I_N\otimes A\,x(k) + I_N\otimes B\,\hat{u}(k) \\ &\quad + c(I_N+D+G)^{-1}E\otimes BKa_x + c\Gamma\otimes BK\bar{a}_{x_0}\end{aligned} \tag{2.70}$$

and the distributed observer dynamics becomes

$$\begin{aligned}\hat{x}(k+1) &= I_N\otimes A\,\hat{x}(k) + I_N\otimes B\,\hat{u}'(k) - c(I_N+G+D)^{-1}\otimes F\varepsilon^{o'}(k) \\ &= I_N\otimes A\,\hat{x}(k) + c(I_N+D+G)^{-1}\otimes BK\hat{\varepsilon}'(k) \\ &\quad - c(I_N+G+D)^{-1}\otimes F\varepsilon^{o'}(k) \\ &= I_N\otimes A\,x(k) \\ &\quad + c(I_N+D+G)^{-1}\otimes BK\,[\hat{\varepsilon} + (E\otimes I_n)a_x + ((L+G)\otimes I_n)\bar{a}_{x_0}] \\ &\quad - c(I_N+D+G)^{-1}\otimes F\,[\varepsilon^o + (E\otimes I_p)a_y]\end{aligned} \tag{2.71}$$

from (2.27) and (2.70)

$$\begin{aligned}\hat{x}(k+1) = &\left[I_N \otimes A - c\Gamma \otimes FC\right]\hat{x}(k) - c\Gamma \otimes BK(\hat{x} - \bar{x}_0)(k) + c\Gamma \otimes F\ y(k) \\ &+ c(I_N + D + G)^{-1}E \otimes BKa_x + c(I_N + D + G)^{-1}G \otimes BK\bar{a}_{x_0} \\ &- c(I_N + D + G)^{-1}E \otimes Fa_y\end{aligned} \tag{2.72}$$

The global disagreement error dynamics modified by the attack is

$$\begin{aligned}\delta(k+1) &= x(k+1) - \bar{x}_0(k+1) \\ &= \left[I_N \otimes A - c\Gamma \otimes BK\right](x - \bar{x}_0)(k) + c\Gamma \otimes BK(x - \hat{x})(k) \\ &\quad + c(I_N + D + G)^{-1}E \otimes BKa_x + c(I_N + D + G)^{-1}G \otimes BK\bar{a}_{x_0} \\ &= A_c\delta(k) + B_c\eta(k) \\ &\quad + \begin{bmatrix} c(I_N + D + G)^{-1}E \otimes BK & c(I_N + D + G)^{-1}G \otimes BK \end{bmatrix} \begin{bmatrix} a_x \\ \bar{a}_{x_0} \end{bmatrix}\end{aligned} \tag{2.73}$$

The global observer error dynamics affected by the attack is

$$\begin{aligned}\eta(k+1) &= x(k=1) - \hat{x}(k=1) \\ &= \left[I_N \otimes A - c\Gamma \otimes FC\right](x - \hat{x})(k) \\ &\quad + c(I_N + D + G)^{-1}E \otimes Fa_y \\ &= A_o\eta(k) + c(I_N + D + G)^{-1}E \otimes Fa_y\end{aligned} \tag{2.74}$$

We can represent the closed loop error system with exogenous attack as

$$\begin{aligned}&\begin{bmatrix}\delta \\ \eta\end{bmatrix}(k+1) = \begin{bmatrix} A_c & B_c \\ 0 & A_o \end{bmatrix}\begin{bmatrix}\delta \\ \eta\end{bmatrix}(k) \\ &+ \begin{bmatrix} c(I_N + D + G)^{-1}E \otimes BK & c\Gamma \otimes BK & 0 \\ 0 & 0 & c(I_N + D + G)^{-1}E \otimes F \end{bmatrix}\begin{bmatrix} a_x \\ \bar{a}_{x_0} \\ a_y \end{bmatrix}\end{aligned} \tag{2.75}$$

Since the attack remains constant, we can evaluate the steady-state values of δ and η by exploiting the fact that the matrix $\begin{bmatrix} A_c & B_c \\ 0 & A_o \end{bmatrix}$ is asymptotically stable in the non-malicious setting. Therefore, by defining $\delta^\infty = \lim_{k\to\infty}\delta(k)$ and $\eta^\infty = \lim_{k\to\infty}\eta(k)$

$$\begin{aligned}\delta^\infty &= (I_{nN} - A_c)^{-1}\Big[B_c\eta^\infty + c(I_N + D + G)^{-1}E \otimes BKa_x \\ &\quad + c(I_N + D + G)^{-1}G \otimes BK\bar{a}_{x_0}\Big] \\ \eta^\infty &= c\left(I_{nN} - A_o\right)^{-1}\Big[(I_N + D + G)^{-1}E \otimes F\Big]\, a_y\end{aligned} \tag{2.76}$$

2.2.2 Attack on the Distributed Controller and Local Observer Model

In this setting the node output vector is $\hat{x}_j$ and the leader output vector is x_0. After the attack, the information transferred through the network by a follower becomes $\hat{x}'_j = \hat{x}_j + a_j \underline{1}_n = \hat{x}_j + a_{x_j}$ and the information shared by the leader becomes $x'_0 = x_0 + \underline{1}_n a_0 = x_0 + a_{x_0}$'. Therefore, the attacks affect only $\hat{\varepsilon}_i \to \hat{\varepsilon}'_i$ and $\hat{u}_i \to \hat{u}'_i$ as already seen in (2.64), (2.65), (2.66) and (2.67).
The states' global dynamics will changes as in (2.70) and the distributed observer dynamics will be modified in

$$\begin{aligned}\hat{x}(k+1) &= I_N \otimes A\, \hat{x}(k) + I_N \otimes B\, \hat{u}'(k) + I_N \otimes F\, \tilde{y}(k) \\ &= I_N \otimes A\, \hat{x}(k) + I_N \otimes B\, \hat{u}(k) + I_N \otimes F\, \tilde{y}(k) \\ &\quad + c(I_N + D + G)^{-1} E \otimes BK a_x + c\Gamma \otimes BK \bar{a}_{x_0} \\ &= [I_N \otimes (A - FC)]\, \hat{x}(k) - c\Gamma \otimes BK(\hat{x} - \bar{x}_0)(k) + I_N \otimes FC\, y(k) \\ &\quad + c(I_N + D + G)^{-1} E \otimes BK a_x + c\Gamma \otimes BK \bar{a}_{x_0}\end{aligned} \tag{2.77}$$

The global disagreement error dynamics will be the same as in (2.73) while we can observe how the local observer error dynamics will not be affected by the attacks:

$$\begin{aligned}\eta_i(k+1) &= x_i(k+1) - \hat{x}_i(k+1) \\ &= Ax_i(k) + B\hat{u}'(k) - [A\hat{x}_i(k) + B\hat{u}'(k) + F\tilde{y}_i(k)] \\ &= Ax_i(k) - A\hat{x}_i(k) - F\tilde{y}_i(k) \\ &= Ax_i(k) - A\hat{x}_i(k) - FC(x_i - \hat{x}_i)(k) \\ &= (A - FC)\eta_i(k) \Rightarrow \eta(k+1) = I_N \otimes (A - FC)\eta(k)\end{aligned} \tag{2.78}$$

Therefore, if F is designed to make $A - FC$ asymptotically stable, $\lim_{k\to\infty} \eta(k) = 0\ \forall a \in \mathbb{R}^N, a_0 \in \mathbb{R}$.
The closed loop error system with exogenous attack can be represented by

$$\begin{aligned}\begin{bmatrix}\delta \\ \eta\end{bmatrix}(k+1) &= \begin{bmatrix}A_c & B_c \\ 0 & I_N \otimes (A - FC)\end{bmatrix}\begin{bmatrix}\delta \\ \eta\end{bmatrix}(k) \\ &+ \begin{bmatrix}c(I_N + D + G)^{-1} E \otimes BK & c\Gamma \otimes BK \\ 0 & 0\end{bmatrix}\begin{bmatrix}a_x \\ \bar{a}_{x_0}\end{bmatrix}\end{aligned} \tag{2.79}$$

We can evaluate the steady-state values of δ by exploiting the fact the the attack is constant and that matrix $\begin{bmatrix}A_c & B_c \\ 0 & I_N \otimes (A - FC)\end{bmatrix}$ is asymptotically stable in the non-malicious setting:

$$\delta^\infty = (I_{nN} - A_c)^{-1}\left[c(I_N + D + G)^{-1} E \otimes BK a_x + c(I_N + D + G)^{-1} G \otimes BK \bar{a}_{x_0}\right] \tag{2.80}$$

2.2.3 Attack on the local Controller and Distributed Observer Model

In this setting the node output vector is $\tilde{y}_j$ and the leader output vector is y_0. After the attack, the information transferred through the network by a follower becomes $\tilde{y}'_j = \tilde{y}_j + a_j \underline{1}_p = \tilde{y}_j + a_{y_j}$ and the information shared by the leader becomes $y'_0 = y_0 + a_{y_0}$.
The attacks affect only the neighborhood output disagreement $\varepsilon_i^o \to \varepsilon_i^{o'}$ where, from (2.56) we obtain

$$\begin{aligned}
\varepsilon_i^{o'} &= \sum_{j\in\mathcal{N}_i} e_{ij}(\tilde{y}'_j - \tilde{y}_i) + g_i(y'_0 - \tilde{y}_i) \\
&= \sum_{j\in\mathcal{N}_i} e_{ij}(\tilde{y}_j + a_{y_j} - \tilde{y}_i) + g_i(y_0 + a_{y_0} - \tilde{y}_i) \\
&= \sum_{j\in\mathcal{N}_i} e_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(y_0 - \tilde{y}_i) + \sum_{j\in\mathcal{N}_i} e_{ij} a_{y_j} + g_i a_{y_0} \\
&= \varepsilon_i^o + \sum_{j\in\mathcal{N}_i} e_{ij} a_{y_j} + g_i a_{y_0}
\end{aligned} \tag{2.81}$$

By defining $\bar{a}_{y_0} = \begin{bmatrix} a_{y_0} \\ a_{y_0} \\ \vdots \\ a_{y_0} \end{bmatrix} \in \mathbb{R}^{nN}$, the global neighborhood output disagreement becomes

$$\varepsilon^{o'}(k) = \varepsilon^o(k) + (E \otimes I_p) a_y + ((L+G) \otimes I_p)\bar{a}_{y_0} \tag{2.82}$$

The distributed observer dynamics will change in

$$\begin{aligned}
\hat{x}(k+1) &= I_N \otimes A\, \hat{x}(k) + I_N \otimes B\, \hat{u}(k) - c(I_N + G + D)^{-1} \otimes F\varepsilon^{o'}(k) \\
&\quad \text{recalling (2.57) and (2.72)} \\
&= I_N \otimes (A - BK)\, \hat{x}(k) + c\Gamma \otimes FC\, \eta(k) - c\Gamma \otimes FC\, \bar{x}_0(k) \\
&\quad - c(I_N + D + G)^{-1} E \otimes F a_y - c\Gamma \otimes F\, \bar{a}_{y_0}
\end{aligned} \tag{2.83}$$

Recalling the algebraic definition of η in (2.58) and the passages in (2.83) , we can say that

$$\begin{aligned}
\eta(k+1) &= x(k+1) - \hat{x}(k+1) = \\
&= A_o \eta(k) + c_1 \Gamma \otimes FC\, \bar{x}_0(k) \\
&\quad - c(I_N + D + G)^{-1} E \otimes F a_y - c\Gamma \otimes F\, \bar{a}_{y_0}
\end{aligned} \tag{2.84}$$

Using that $\hat{x} = x - \eta = x - \bar{x}_0 - \eta + \bar{x}_0 = \delta - (\eta - \bar{x}_0)$ and $\vartheta = \eta - \bar{x}_0$, one finds

$$\begin{aligned}
\delta(k+1) &= I_N \otimes (A - BK)\delta(k) + (I_N \otimes BK)(\eta(k) - \bar{x}_0(k)) \\
\vartheta(k+1) &= \eta(k+1) - \bar{x}_0(k+1) = \\
&= A_o \vartheta(k) + c(I_N + D + G)^{-1} E \otimes F a_y + c\Gamma \otimes F\, \bar{a}_{y_0}
\end{aligned} \tag{2.85}$$

The closed loop error system with exogenous attack will change in

$$\begin{aligned}\begin{bmatrix}\delta\\ \vartheta\end{bmatrix}(k+1) = &\begin{bmatrix}I_N \otimes (A-BK) & I_N \otimes BK\\ 0 & A_o\end{bmatrix}\begin{bmatrix}\delta\\ \vartheta\end{bmatrix}(k)\\ &+\begin{bmatrix}0 & 0\\ c(I_N+D+G)^{-1}E\otimes F & c\Gamma\otimes F\end{bmatrix}\begin{bmatrix}a_y\\ \bar{a}_{y_0}\end{bmatrix}\end{aligned} \tag{2.86}$$

Since the attack remains constant, we can evaluate the steady-state values of δ by exploiting the fact that the matrix $\begin{bmatrix}I_N \otimes (A-BK) & I_N \otimes BK\\ 0 & A_o\end{bmatrix}$ is asymptotically stable in the non-malicious setting. Defining $\vartheta^\infty = \lim_{k\to\infty}\vartheta(k)$,

$$\begin{aligned}\delta^\infty &= \left[I_{nN} - I_N \otimes (A-BK)\right]^{-1}(I_N\otimes BK)\vartheta^\infty\\ \vartheta^\infty &= (I_{nN}-A_o)^{-1}\left[c(I_N+D+G)^{-1}E\otimes F\, a_y + c\Gamma\otimes F\,\bar{a}_{y_0}\right]\end{aligned} \tag{2.87}$$

2.3 Modeling Sparse Attack Propagation in Distributed Observers

Our first objective is to establish a general framework for a distributed observer capable of reconstructing the states of nodes subject to malicious attacks across all analyzed models. To this end, we first recast the distributed dynamics into a representation consistent with classical discrete-time control theory. This specific formulation is intended to facilitate the application of a gradient-descent-based algorithm, enabling the online estimation of the true states directly from the corrupted measurements:

$$\begin{cases}x(k+1) = Ax(k) = A^k x(0)\\ y(k) = Cx(k) + a\end{cases} \tag{2.88}$$

where a is the general sparse attack vector on the output measurement.
The issues with this description are:

1. the entire system is considered autonomous
2. the sparse attack vector is added to the outputs $y(k)$ and thus by observing the outputs, a sparse attack vector can be derived

To solve the first issue, we can consider the distributed dynamic system as autonomous by considering an extended state vector

$$x_{ext} = \begin{bmatrix}x_o\\ x\\ \hat{x}\end{bmatrix} \in \mathbb{R}^{n(2N+1)}$$

Its dynamics, without attacks, follows the pattern

$$x_{ext}(k+1) = \begin{bmatrix} x_0 \\ x \\ \hat{x} \end{bmatrix}(k+1) = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} x_o \\ x \\ \hat{x} \end{bmatrix}(k) = \mathcal{A}x_{ext}(k) \tag{2.89}$$

$$x_{ext}(k) = \mathcal{A}^k x_{ext}(0),\ \mathcal{A} \in \mathbb{R}^{n(2N+1)\times n(2N+1)} \tag{2.90}$$

We can construct $\mathcal{A}$ for the various models analyzed starting from the dynamic equations established in the previous sections and by observing that, by exploiting the mixed Kronecker product

$$c\Gamma \otimes BK\bar{x}_0 = c(\Gamma \otimes BK)\cdot(\underline{\mathbf{1}}_N \otimes x_0) = c\Gamma \cdot \underline{\mathbf{1}}_N \otimes BKx_0 = c(\Gamma \cdot \underline{\mathbf{1}}_N \otimes BK)x_0 \tag{2.91}$$

In the fully distributed setting, we can define $\mathcal{A}_{dist.dist}$ as

$$\mathcal{A}_{dist.dist} = \begin{bmatrix} A & \underline{\mathbf{0}}_{n\times nN} & \underline{\mathbf{0}}_{n\times nN} \\ c\Gamma\underline{\mathbf{1}}_N \otimes BK & I_N \otimes A & -c\Gamma \otimes BK \\ c\Gamma\underline{\mathbf{1}}_N \otimes BK & c\Gamma \otimes FC & I_N \otimes A - c\Gamma \otimes (BK + FC) \end{bmatrix} \tag{2.92}$$

For the model with the local observer, we define $\mathcal{A}_{loc.dist}$

$$\mathcal{A}_{loc.dist} = \begin{bmatrix} A & \underline{\mathbf{0}}_{n\times nN} & \underline{\mathbf{0}}_{n\times nN} \\ c\Gamma\underline{\mathbf{1}}_N \otimes BK & I_N \otimes A & -c\Gamma \otimes BK \\ c\Gamma\underline{\mathbf{1}}_N \otimes BK & I_N \otimes FC & I_N \otimes (A - FC) - c\Gamma \otimes BK \end{bmatrix}$$

For the last model, the one with the a local controller, we can define $\mathcal{A}_{dist.loc}$

$$\mathcal{A}_{dist.loc} = \begin{bmatrix} A & \underline{\mathbf{0}}_{n\times nN} & \underline{\mathbf{0}}_{n\times nN} \\ \underline{\mathbf{0}}_{nN\times n} & I_N \otimes A & -I_N \otimes BK \\ -c\Gamma\underline{\mathbf{1}}_N \otimes FC & c\Gamma \otimes FC & I_N \otimes (A - BK) - c\Gamma \otimes FC \end{bmatrix}$$

At this stage, an autonomous description of the system has been derived. We now analyze how the attacks influence the outputs $y(k)$ in accordance with the target model (2.88).

By assuming a sparse attack vector

$$a_{ext} \in \mathbb{R}^{N+1} = \begin{bmatrix} a_0 \\ a \end{bmatrix}, \tag{2.93}$$

and defining the global measurement matrix as $\mathcal{C} = I_{2N+1} \otimes C$, the global output vector $y_{ext}(k) \in \mathbb{R}^{p(2N+1)}$ is given by:

$$y_{ext}(k) = \mathcal{C}x_{ext}(k). \tag{2.94}$$

Our objective is to reformulate the output equation to isolate the attack contribution, leading to a structure such as:

$$y_{ext}(k) = \mathcal{C}x_{ext}(k) + \mathfrak{a}, \quad \text{with } \mathfrak{a} = \mathcal{C}\mathfrak{A}a_{ext} \tag{2.95}$$

where $\mathfrak{A}$ denotes the matrix characterizing the impact of the attack as seen by the augmented state x_{ext} over time. This latter formulation is intended to highlight that, in the proposed model, the attack affects the internal states before propagating to the outputs.
We define the instantaneous attack impact matrix

$$\Psi = \begin{bmatrix} \Psi_{x_0} \in \mathbb{R}^{n\times(N+1)} \\ \Psi_x \in \mathbb{R}^{nN\times(N+1)} \\ \Psi_{\hat{x}} \in \mathbb{R}^{nN\times(N+1)} \end{bmatrix} \in \mathbb{R}^{n(2N+1)\times(N+1)} \tag{2.96}$$

as the one characterizing the effect of a sparse external attack vector a_{ext} on the augmented system state x_{ext} within a single time step. It represent the forcing term that couples the attack vector to the system dynamics, allowing the evolution of the corrupted state to be expressed as

$$x'_{ext}(k+1) = \mathcal{A}x_{ext}(k) + \Psi a_{ext} \tag{2.97}$$

where the superscript (') denotes the corrupted value.
Since the leader node is not affected by the attacks, $\Psi_{x_0} = 0$.
Due to the linearity of the augmented system, the corrupted state $x'_{ext}(k)$ can be decomposed into its nominal evolution and the accumulated contribution of the external attack. By expanding the state-space trajectory, we observe that the attack effect manifests as a power series of the system matrix $\mathcal{A}$, yielding:

$$\begin{aligned}
x'_{ext}(0) &= x_{ext}(0) \\
x'_{ext}(1) &= \mathcal{A}x_{ext}(0) + \Psi a_{ext} = x_{ext}(1) + \Psi a_{ext} \\
x'_{ext}(2) &= \mathcal{A}x'_{ext}(1) + \Psi a_{ext} \\
&= \mathcal{A}(\mathcal{A}x_{ext}(0) + \Psi a_{ext}) + \Psi a_{ext} \\
&= \mathcal{A}^2 x_{ext}(0) + (\mathcal{A}\Psi + \Psi)a_{ext} \\
&= x_{ext}(2) + (\mathcal{A} + I_{n(2N+1)})\Psi a_{ext} \\
x'_{ext}(3) &= \mathcal{A}x'_{ext}(2) + \Psi a_{ext} \\
&= \mathcal{A}(\mathcal{A}^2 x_{ext}(0) + (\mathcal{A}\Psi + \Psi)a_{ext}) + \Psi a_{ext} \\
&= \mathcal{A}^3 x_{ext}(0) + (\mathcal{A}^2 + \mathcal{A} + I_{n(2N+1)})\Psi a_{ext} \\
&= x_{ext}(3) + (\mathcal{A}^2 + \mathcal{A} + I_{n(2N+1)})\Psi a_{ext} \\
&\vdots \\
x'_{ext}(k) &= \mathcal{A}^k x_{ext}(0) + \sum_{n=0}^{k-1} \mathcal{A}^n \Psi a_{ext} = x_{ext}(k) + \sum_{n=0}^{k-1} \mathcal{A}^n \Psi a_{ext}
\end{aligned} \tag{2.98}$$

where $x_{ext}(k)$ represents the unperturbed state and the summation term characterizes the cumulative bias injected by the persistent attack a_{ext}.
Hence, we can write

$$\begin{aligned} \mathfrak{A} &= \mathfrak{A}(k) = \sum_{n=0}^{k-1} \mathcal{A}^n \Psi \\ \mathfrak{a} &= \mathfrak{a}(k) = \mathcal{C}\mathfrak{A}(k) a_{ext} = \mathcal{C} \sum_{n=0}^{k-1} \mathcal{A}^n \Psi a_{ext} \end{aligned} \tag{2.99}$$

Since the sparse attack vector is a_{ext}, we need a simple way to extract a from $a_x = a \otimes \underline{1}_n$ and $a_y = a \otimes \underline{1}_n$ and put it to the right-side of the equations. We can exploit the following property: $\forall\ \mathbf{a} \in \mathbb{R}^N,\ \mathbf{b} \in \mathbb{R}^n$ the following assertion holds true:

$$\begin{aligned} (I_N \otimes \mathbf{b}) \cdot \mathbf{a} &= \begin{bmatrix} \mathbf{b} & 0 & \cdots & 0 \\ 0 & \mathbf{b} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{b} \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = a_1 \cdot \begin{bmatrix} \mathbf{b} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + a_2 \cdot \begin{bmatrix} 0 \\ \mathbf{b} \\ \vdots \\ 0 \end{bmatrix} + \cdots + a_N \cdot \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \mathbf{b} \end{bmatrix} \\ &= \begin{bmatrix} a_1 \mathbf{b} \\ a_2 \mathbf{b} \\ \vdots \\ a_N \mathbf{b} \end{bmatrix} \doteq \mathbf{a} \otimes \mathbf{b} \end{aligned} \tag{2.100}$$

It follows that

$$a_x = \begin{bmatrix} a_1 \underline{1}_n \\ a_2 \underline{1}_n \\ \vdots \\ a_N \underline{1}_n \end{bmatrix} = (I_N \otimes \underline{1}_n) \cdot a, \quad a_y = \begin{bmatrix} a_1 \underline{1}_p \\ a_2 \underline{1}_p \\ \vdots \\ a_N \underline{1}_p \end{bmatrix} = (I_N \otimes \underline{1}_p) \cdot a \tag{2.101}$$

Exploiting (2.101) and the mixed Kronecker product (2.4), we write Ψ in a compact form for each model. In the fully distributed setting, we recall the global dynamics (2.28) and the attack effects on the global nodes dynamics (2.70):

$$\begin{aligned} x'(k+1) &= x(k+1) + c(I_N + D + G)^{-1} E \otimes BK a_x + c\Gamma \otimes BK \bar{a}_{x_0} \\ &= x(k+1) + c\,(\Gamma \otimes BK)\,\underline{1}_{nN}\, a_0 \\ &\quad + c\left[(I_N + D + G)^{-1} E \otimes BK\right] \cdot \left[I_N \otimes \underline{1}_n\right] a \end{aligned}$$

here we apply the mixed Kronecker product (2.4)

$$\begin{aligned} x'(k+1) &= x(k+1) + c\,(\Gamma \otimes BK)\,\underline{1}_{nN}\,a_0 \\ &\quad + c\left[(I_N + D + G)^{-1}E \cdot I_N\right] \otimes [BK \cdot \underline{1}_n]\,a \\ &= x(k+1) + c\,(\Gamma \otimes BK)\,\underline{1}_{nN}\,a_0 + c\left[(I_N + D + G)^{-1}E \otimes BK \cdot \underline{1}_n\right] a \\ &= x(k+1) + c\begin{bmatrix}(\Gamma \otimes BK)\underline{1}_{nN} & (I_N + D + G)^{-1}E \otimes BK \cdot \underline{1}_n\end{bmatrix}\begin{bmatrix}a_0 \\ a\end{bmatrix} \\ &= x(k+1) + \Psi_x a_{ext} \end{aligned} \tag{2.102}$$

The attacks effects on the global observer dynamics, in the single time step, can be derived by looking at (2.27) and (2.72) and by following the same passages as before:

$$\begin{aligned} \hat{x}'(k+1) &= \hat{x}(k+1) + c(I_N + D + G)^{-1}E \otimes BKa_x + c\Gamma \otimes BK\bar{a}_{x_0} \\ &\quad - c(I_N + D + G)^{-1}F \otimes BKa_y \\ &= \dots \\ &= \hat{x}(k+1) \\ &\quad + c\begin{bmatrix}(\Gamma \otimes BK)\underline{1}_{nN} & (I_N + D + G)^{-1}E \otimes (BK \cdot \underline{1}_n - F \cdot \underline{1}_p)\end{bmatrix}\begin{bmatrix}a_0 \\ a\end{bmatrix} \\ &= \hat{x}(k+1) + \Psi_{\hat{x}} a_{ext} \end{aligned} \tag{2.103}$$

Therefore, we can define the matrix Ψ for the fully distributed model as

$$\Psi_{dist.dist} = \begin{bmatrix} 0 & 0 \\ c(\Gamma \otimes BK)\underline{1}_{nN} & c(I_N + D + G)^{-1}E' \otimes BK\underline{1}_n \\ c(\Gamma \otimes BK)\underline{1}_{nN} & c(I_N + D + G)^{-1}E' \otimes (BK\underline{1}_n - F\underline{1}_p) \end{bmatrix} \tag{2.104}$$

Following the same rationale, we find matrix Ψ for the local observer model and for the local controller model, respectively

$$\begin{aligned} \Psi_{loc.dist} &= \begin{bmatrix} 0 & 0 \\ c(\Gamma \otimes BK)\underline{1}_{nN} & c(I_N + D + G)^{-1}E' \otimes BK\underline{1}_n \\ c(\Gamma \otimes BK)\underline{1}_{nN} & c(I_N + D + G)^{-1}E' \otimes BK\underline{1}_n \end{bmatrix} \\ \Psi_{dist.loc} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -c(\Gamma \otimes F)\underline{1}_{pN} & -c(I_N + D + G)^{-1}E' \otimes F\underline{1}_p \end{bmatrix} \end{aligned} \tag{2.105}$$

In the following, we describe a distributed observer that models the attacks of the individual nodes, while considering the global network dynamics for each setting analyzed:

- distributed observers and controllers model

$$\begin{cases} x'_i(k+1) = x_i(k+1) + \mathfrak{A}(k+1)_{x_i} a_{ext} \\ \qquad\qquad = Ax_i(k) + B\hat{u}_i(k) + \mathfrak{A}(k+1)_{x_i} a_{ext} \\ y'_i(k) = Cx'_i(k) = C(x_i(k) + \mathfrak{A}(k)_{x_i} a_{ext}) \\ \hat{x}'_i(k+1) = \hat{x}_i(k+1) + \mathfrak{A}(k+1)_{\hat{x}_i} a_{ext} \\ \qquad\qquad = A\hat{x}_i + B\hat{u}_i(k) - c(1+d_i+g_i)^{-1} F\varepsilon_i^o(k) + \mathfrak{A}(k+1)_{\hat{x}_i} a_{ext} \\ \hat{y}'_i(k) = C\hat{x}'_i(k) = C(\hat{x}_i(k) + \mathfrak{A}(k)_{\hat{x}_i} a_{ext}) \\ \mathfrak{A}(k) = \sum_{n=0}^{k-1} \mathcal{A}^n_{dist.dist} \Psi_{dist.dist} \end{cases} \tag{2.106}$$

- local observers and distributed controllers model

$$\begin{cases} x'_i(k+1) = x_i(k+1) + \mathfrak{A}(k+1)_{x_i} a_{ext} \\ \qquad\qquad = Ax_i(k) + B\hat{u}_i(k) + \mathfrak{A}(k+1)_{x_i} a_{ext} \\ y'_i(k) = Cx'_i(k) = C(x_i(k) + \mathfrak{A}(k)_{x_i} a_{ext}) \\ \hat{x}'_i(k+1) = \hat{x}_i(k+1) + \mathfrak{A}(k+1)_{\hat{x}_i} a_{ext} \\ \qquad\qquad = A\hat{x}_i + B\hat{u}_i(k) + F\tilde{y}_i(k) + \mathfrak{A}(k+1)_{\hat{x}_i} a_{ext} \\ \hat{y}'_i(k) = C\hat{x}'_i(k) = C(\hat{x}_i(k) + \mathfrak{A}(k)_{\hat{x}_i} a_{ext}) \\ \mathfrak{A}(k) = \sum_{n=0}^{k-1} \mathcal{A}^n_{loc.dist} \Psi_{loc.dist} \end{cases} \tag{2.107}$$

- distributed observers and local controllers model

$$\begin{cases} x'_i(k+1) = x_i(k+1) + \mathfrak{A}(k+1)_{x_i} a_{ext} \\ \qquad\qquad = Ax_i(k) + B\hat{u}_i(k) + \mathfrak{A}(k+1)_{x_i} a_{ext} \\ y'_i(k) = Cx'_i(k) = C(x_i(k) + \mathfrak{A}(k)_{x_i} a_{ext}) \\ \hat{x}'_i(k+1) = \hat{x}_i(k+1) + \mathfrak{A}(k+1)_{\hat{x}_i} a_{ext} \\ \qquad\qquad = A\hat{x}_i + B\hat{u}_i(k) - c(1+d_i+g_i)^{-1} F\varepsilon_i^o(k) + \mathfrak{A}(k+1)_{\hat{x}_i} a_{ext} \\ \hat{y}'_i(k) = C\hat{x}'_i(k) = C(\hat{x}_i(k) + \mathfrak{A}(k)_{\hat{x}_i} a_{ext}) \\ \mathfrak{A}(k) = \sum_{n=0}^{k-1} \mathcal{A}^n_{dist.loc} \Psi_{dist.loc} \end{cases} \tag{2.108}$$

where $\mathfrak{A}(k+1)_{x_i}$ and $\mathfrak{A}(k+1)_{\hat{x}_i}$ represent the local attack impact affecting respectively the state and the estimate state of the i-th node at timestep $k+1$. These terms are derived from the global accumulation matrix $\mathfrak{A}(k+1)$ by extracting the corresponding block-rows.

2.4 Conclusions

In this chapter, it was shown that models analyzed under constant attacks on node outputs cannot be addressed using a distributed observer that estimates states via online gradient descent. This is because such systems cannot be represented as in (2.88), as the attack effects are not constant but rather cumulative over time. Furthermore, the use of an online gradient-descent-based algorithm requires a state observer that would conflict with the observer already present in the model. We can seek a solution by implementing a framework that ensures resilience in the synchronization of network dynamics.
Finally, since the local observer of the second analyzed system is unaffected by attacks, we will henceforth assume, without loss of generality, that the estimated state is equal to the agents' true state ($\hat{x} = x$). This assumption also serves to streamline the descriptions of the dynamics.

Chapter 3

Resilient First Order Consensus

The resilient control framework introduced by Scardovi et al. [2], specifically as adapted by LeBlanc et al. [3] through the integration of the **Asymptotic Resilient Consensus Protocol (ARC-P)**, provides a solid and mathematically rigorous foundation for achieving resilient synchronization in multi-agent systems—that is, the ability of the healthy cooperative nodes to asymptotically track the leader's trajectory despite the presence of compromised or malicious agents in the network. Its primary advantage resides in its distributed filtering capabilities: by leveraging a local sorting and pruning logic, ARC-P allows agents to autonomously discard extreme malicious data, offering a straightforward and strictly proven methodology for adversarial robustness.

The main objective of this chapter is to thoroughly analyze this foundational framework and to evaluate its structural consistencies with the cooperative control architecture presented by Lewis et al. [1] in the previous chapter. Both methodologies fundamentally rely on a distributed full-state feedback mechanism where the control input is synthesized via the relative state discrepancies between an agent and its neighbors. This profound architectural parallel strongly suggests that the resilience properties validated in [3] can be effectively bridged and adapted to our primary system of study.

While the previous chapter investigated discrete-time formulations to accommodate gradient-based algorithmic estimators, this specific chapter adopts the continuous-time dynamics detailed in [3]. The rationale behind this modeling choice is strictly tied to maintaining mathematical coherence with the original Scardovi framework, ensuring that the theoretical guarantees regarding modal state projections and ARC-P continuous filtering remain valid and rigorously applicable.

However, as the mathematical and simulative analyses presented in the following

sections will reveal, relying solely on algorithmic filtering is insufficient to guarantee optimal system performance. The underlying network topology fundamentally dictates the propagation of attacks and the speed of synchronization. Exposing the structural vulnerabilities of standard robust architectures—such as information bottlenecks and the detrimental "structural inertia" caused by backward-pointing edges in cyclic graphs—is a core objective of this chapter. This analysis mathematically motivates our primary innovative contribution: the design of a novel class of scalable, directed acyclic topologies that drastically accelerate tracking convergence while maintaining a strictly bounded local connectivity requirements.

The chapter is organized as follows: Section 3.1 details the synthesis of the continuous-time distributed control design based on modal projections. Section 3.2 introduces the ARC-P filter and its operational logic. Section 3.3 defines the topological conditions for network robustness and rigorously proves the resilience of fully connected and circulant digraphs. Section 3.4 presents the convergence speed analysis for leader-follower dynamics, evaluating the impact of active attacks and highlighting the structural vulnerabilities of standard architectures. Subsequently, Section 3.5 introduces our primary contribution: the synthesis of an optimized, scalable network design, demonstrating its theoretical resilience and superior tracking performance. Finally, Section 3.6 summarizes the chapter's findings.

3.1 The Scardovi/Sepulchre First Order Distributed Control Design

The framework introduced by Scardovi et al. [2] addresses the synchronization problem for N identical continuous-time linear agents, reducing it to a static consensus problem via a suitable coordinate transformation.
Under the assumption of full state measurement ($C = I$), the continuous-time dynamics of each agent $i \in \mathcal{N}$ is governed by:

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t) \tag{3.1}$$

where $x_i \in \mathbb{R}^n$ is the state vector, $u_i \in \mathbb{R}^m$ is the control input, and the pair (A, B), with $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, is assumed to be stabilizable.
Synchronization to a common autonomous trajectory $x_0(t) = e^{At}x_0(0)$ is achieved by continuously comparing the backward projection of the i-th node's current state, defined as $x_i(0)(t) = e^{-At}x_i(t)$, with the corresponding projections of its neighbors [2].
Building on this concept, LeBlanc et al. [3] apply a projection into the modal coordinate space. This approach allows the control dynamics to operate directly on the system's modes. By assuming that A is marginally stable, it can be decomposed

via an invertible transformation matrix Q such that $A = QRQ^{-1}$, where R is a block-diagonal matrix capturing the decoupled integrator and oscillatory modes.

Specifically, since all eigenvalues of a marginally stable matrix lie on the imaginary axis and are simple roots of the minimal polynomial, its real Jordan canonical form R is structurally defined as:

$$R = \begin{bmatrix} 0_{n_0 \times n_0} & 0 & \cdots & 0 \\ 0 & \Omega_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Omega_p \end{bmatrix}, \quad \text{with } \Omega_k = \begin{bmatrix} 0 & \omega_k \\ -\omega_k & 0 \end{bmatrix} \tag{3.2}$$

where n_0 represents the multiplicity of the zero eigenvalue (corresponding to the integrator modes), and $\pm j\omega_k$ (for $k = 1, \ldots, p$) are the pure imaginary conjugate pairs representing the network's oscillatory modes.

Consequently, the exponential state transition matrix in the modal space, e^{Rt}, perfectly preserves this decoupled architecture, yielding a block-diagonal matrix composed of static identity blocks and harmonic rotation matrices:

$$e^{Rt} = \begin{bmatrix} I_{n_0} & 0 & \cdots & 0 \\ 0 & e^{\Omega_1 t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\Omega_p t} \end{bmatrix}, \quad \text{with } e^{\Omega_k t} = \begin{bmatrix} \cos(\omega_k t) & \sin(\omega_k t) \\ -\sin(\omega_k t) & \cos(\omega_k t) \end{bmatrix} \tag{3.3}$$

The crucial step, which involves projecting the states to their equivalent initial conditions, is performed in the space of the decoupled dynamic modes by defining the transformed state vector $z_i(t) \in \mathbb{R}^n$ as $z_i(t) = e^{-Rt}Q^{-1}x_i(t)$. By leveraging the projection of the agents' initial states into the modal space, the synchronization problem is reduced to achieving consensus on a static vector z_0, which represents the initial condition of the synchronized trajectory.

The significance of this decoupling is twofold: first, it enables the control law to address the various spectral components (*e.g.*, rigid-body motions and harmonic oscillations) independently; second, it ensures that the inter-agent interactions solely correct trajectory deviations without interfering with the intrinsic internal dynamics of the system. Consequently, the agents asymptotically converge to a safe and stable zero-input solution $x_0(t) = Qe^{Rt}z_0(0)$ that satisfies $\dot{x}_0(t) = Ax_0(t)$, thereby aligning their amplitudes and phases to a common evolution dictated by the consensus reached on their initial conditions.

The synchronization strategy is implemented through a dynamic full-state feedback controller, where the control input for each agent $i \in \mathcal{N}$ is defined as $u_i = K\eta_i$. Here, $\eta_i \in \mathbb{R}^n$ represents the internal state of the controller, which is designed to asymptotically vanish as synchronization is achieved. The feedback gain matrix $K \in \mathbb{R}^{m \times n}$ is designed in order to make $(A + BK)$ Hurwitz.

In order to guarantee synchronization, $\eta_i(0) = 0,\ \forall i \in \mathcal{N}$ (theorem 4 of [3]).
The dynamics of this controller are given by:

$$\dot{\eta}(t) = (A + BK)\eta(k) - Qe^{Rt} \sum_{j \in \mathcal{J}_i} w_{ij}(t) \left[e^{-Rt} Q^{-1} \xi_j(t) - e^{-Rt} Q^{-1} \xi_i(t) \right]. \quad (3.4)$$

In this formulation, $\xi_j(t) = x_j(t) - \eta_j(t) \in \mathbb{R}^n$ serves as the auxiliary variable transmitted to neighboring agents. The set $\mathcal{J}_i(t) = \mathcal{N}_i^{in}(t) \cup \{i\}$ represents the inclusive in-neighbors of agent i, encompassing both the agent itself and its incoming communication links. Furthermore, $w_{(j,i)}(t) > 0$ denote the positive, piecewise continuous, and uniformly bounded gain weights that agent i applies to the relative state difference $\left[e^{-Rt} Q^{-1} \xi_j(t) - e^{-Rt} Q^{-1} \xi_i(t) \right]$ with respect to neighbor j within the control law. For simplicity we will consider the weights w_{ij} as static.
At this point, the control dynamics of the i-th agent can be described as

$$\begin{cases} \dot{x}_i(t) = Ax(t) + Bu_i(t) \\ \dot{\eta}_i(t) = (A + BK)\eta_i(t) - Qe^{Rt} \sum_{j \in \mathcal{J}_i} w_{ij} \left[z_j(t) - z_i(t) \right] \\ u_i(t) = K\eta_i(t) \\ \xi_i(t) = x_i(t) - \eta_i(t) \\ z_i(t) = e^{-Rt} Q^{-1} \xi_i(t) \end{cases} \quad (3.5)$$

An analogous discrete-time control system can be formulated by considering the Jordan decomposition

$$\begin{aligned} A &= QJQ^{-1} \\ A^k &= QJ^kQ^{-1} \end{aligned} \quad (3.6)$$

which yields the projection of the initial state into the modal space over time, defined as $z_i(k) = J^{-k}Q^{-1}x_i(k)$, serving as the consensus variable.
The discrete-time control dynamics for the i-th agent can be described as follows:

$$\begin{cases} x_i(k+1) = Ax_i(k) + Bu_i(k) \\ \eta_i(k+1) = (A + BK)\eta_i(k) - QJ^k \sum_{j \in \mathcal{J}_i} w_{ij} \left[z_j(k) - z_i(k) \right] \\ u_i(k) = K\eta_i(k) \\ \xi_i(k) = x_i(k) - \eta_i(k) \\ z_i(k) = J^{-k}Q^{-1}\xi_i(k) \end{cases} \quad (3.7)$$

where K is the feedback gain matrix chosen such that $(A + BK)$ is asymptotically stable (*i.e.*, its eigenvalues lie strictly inside the unit circle in the complex plane), and the weights satisfy $w_{ij} \geq 0$ with $\sum_{j \in \mathcal{J}_i} w_{ij} \leq 1$ for all $i \in \mathcal{V}$.
Thus, the objective of the discrete-time system is also to synchronize the nodes'

trajectories to a common autonomous trajectory $x_0(k) = QJ^k z_0(0)$.
Consequently, several parallels can be drawn with the control system presented in the first chapter. First and foremost, the fundamental description of the agent dynamics remains unchanged. The second similarity pertains to the generation of the control input: in the framework by Lewis et al., u_i is computed as a weighted and normalized sum of the discrepancies between the node's own information and that of its neighbors (2.10), which is then applied linearly to the dynamics; similarly, Scardovi's system computes a weighted sum following the exact same logic, with the key difference that it is processed through a dynamic controller η_i. The final parallel concerns the synchronization objective itself. While in the framework by Scardovi et al. consensus is sought on a common autonomous trajectory $x_0(t) = e^{At}x_0(0)$ via a projection to the initial state, the system proposed by Lewis et al. requires the nodes to synchronize with the autonomous trajectory of a leader node, $x_0(k) = A^k x_0(0)$, by directly comparing instantaneous values without any backward projection in time.

3.2 ARC-P Filter for Resilient Consensus

We now introduce the Asymptotic Resilient Consensus Protocol (ARC-P), which constitutes the core of the resilient control framework presented by LeBlanc et al. [3].
The objective of ARC-P is to render a standard consensus protocol resilient against up to $F \in \mathbb{Z}_+$ malicious attacks by employing a *sorting and pruning* logic.
The underlying philosophy of ARC-P is that a compromised node will likely broadcast information that significantly deviates from the safe, nominal state. Consequently, the protocol compares the consensus variable of a given node with those received from its neighbors. It then discards the F highest and the F lowest values, allowing only the most closely clustered values to pass through the filter. These remaining values are thereby deemed safe and correct, ensuring the robustness of the consensus update.
In greater detail, for each node i, the protocol executes the following steps:

1. computes the differences $x_j - x_i$ for all $j \in \mathcal{N}_i^{in}$;
2. sorts the computed differences $x_j - x_i$;
3. discards the F largest strictly positive values (*i.e.*, it removes the contributions from neighbors where $x_j > x_i$); if there are fewer than F such values, it discards all of them;
4. discards the F smallest strictly negative values (*i.e.*, it removes the contributions from neighbors where $x_j < x_i$); if there are fewer than F such values, it discards all of them;

5. returns the updated control action $\sum_{j \in \mathcal{N}_i^{in} \setminus \mathcal{R}_i(t)} w_{ij}(t)[x_j(t) - x_i(t)]$, where the weights $w_{ij}(t) > 0$ for all i, j can be interpreted as proportional controller gains.

Here, $\mathcal{R}_i(t)$ denotes the set of in-neighbors whose values are discarded or ignored by node i at time t during the execution of the ARC-P algorithm.
Consequently, the continuous-time consensus dynamics for a scalar state under ARC-P is given by:

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i^{in} \setminus \mathcal{R}_i(t)} w_{ij}(t)(x_j(t) - x_i(t)). \tag{3.8}$$

In the vector case, the ARC-P is executed coordinate-wise, and the consensus dynamics becomes:

$$\dot{x}_i(t) = \begin{bmatrix} \sum_{j \in \mathcal{N}_i^{in} \setminus \mathcal{R}_{i,1}(t)} w_{ij}(t)[x_{j,1}(t) - x_{i,1}(t)] \\ \vdots \\ \sum_{j \in \mathcal{N}_i^{in} \setminus \mathcal{R}_{i,n}(t)} w_{ij}(t)[x_{j,n}(t) - x_{i,n}(t)] \end{bmatrix}, \quad x_i \in \mathbb{R}^n \tag{3.9}$$

where $\mathcal{R}_{i,k}(t)$ denotes the set of neighboring nodes whose values are discarded or ignored by node i at time t during the execution of the ARC-P for the k-th entry. To represent the ARC-P operator with respect to a vector consensus state for the i-th node, we introduce the notation $\Phi_F\left(\{x_j(t)\}_{j \in \mathcal{J}_i}\right)$, defined as:

$$\Phi_F\left(\{x_j(t)\}_{j \in \mathcal{J}_i}\right) = \begin{bmatrix} \sum_{j \in \mathcal{N}_i^{in} \setminus \mathcal{R}_{i,1}(t)} w_{ij}(t)[x_{j,1}(t) - x_{i,1}(t)] \\ \vdots \\ \sum_{j \in \mathcal{N}_i^{in} \setminus \mathcal{R}_{i,n}(t)} w_{ij}(t)[x_{j,n}(t) - x_{i,n}(t)] \end{bmatrix}. \tag{3.10}$$

Since the ARC-P algorithm discards extreme values to mitigate attacks, we establish the following necessary condition regarding the network topology:

Assumption 3.1 (Minimum Connectivity)**.** To guarantee synchronization in the presence of up to F malicious nodes (under an F-total threat model), every node in the network must have an in-degree d_i satisfying:

$$d_i \geq 2F + 1 \quad \forall i \in \mathcal{V}$$

Nevertheless, ARC-P inherently provides a *safety* property for nodes lacking a sufficiently high in-degree [3]. Specifically, if a node does not receive enough information to securely update its state via consensus, it simply evolves along its

autonomous trajectory, which is by definition considered to remain within the safe region [3].
The control framework proposed by Scardovi et al., when integrated with the ARC-P algorithm [3], becomes:

$$\begin{cases} \dot{x}_i(t) = Ax_i(t) + Bu_i(t) \\ \dot{\eta}_i(t) = (A + BK)\eta_i(t) - Qe^{Rt}\Phi_F\left(\{z_j(t)\}_{j\in\mathcal{J}_i(t)}\right) \\ u_i(t) = K\eta_i(t) \\ \xi_j(t) = x_j(t) - \eta_j(t) \\ z_j(t) = e^{-Rt}Q^{-1}\xi_j(t) \end{cases} \tag{3.11}$$

The ARC-P algorithm also has a discrete-time counterpart: the **Weighted-Mean-Subsequence-Reduced** (W-MSR) algorithm. Its operational logic is practically identical to that of ARC-P, with the fundamental difference that the weights $w_{ij}(k)$ must represent a convex combination of the unfiltered values, such that $\sum_{j\in\mathcal{J}_i\setminus\mathcal{R}_i(k)} w_{ij}(k) = 1$ for all $i \in \mathcal{N}$. Consequently, to ensure an optimal convergence rate, these weights must be dynamically updated at each time step based on the number of inputs discarded by the filter [4]. In the analyzed scenario, a consistent choice for the weights would be $w_{ij}(k) = \frac{1}{\left|\mathcal{N}_i^{in}\setminus\mathcal{R}_i(k)\right|} = \frac{1}{d_i - |\mathcal{R}_i(k)|}$.
Denoting the W-MSR operator as $\Phi_F^{\text{W-MSR}}$, the discrete-time control system by Scardovi et al., augmented with the W-MSR filter, is formulated as:

$$\begin{cases} x_i(k+1) = Ax_i(k) + Bu_i(k) \\ \eta_i(k+1) = (A + BK)\eta_i(k) - QJ^k\Phi_F^{\text{W-MSR}}\left(\{z_j(k)\}_{j\in\mathcal{J}_i(k)}\right) \\ u_i(k) = K\eta_i(k) \\ \xi_j(k) = x_j(k) - \eta_j(k) \\ z_j(k) = J^{-k}Q^{-1}\xi_j(k) \end{cases} \tag{3.12}$$

3.3 Robustness and Analysis of some Robust Topologies

Having established the local control framework, we must now define the topological conditions required to guarantee synchronization. In this context, LeBlanc et al. [3] provide the theoretical foundations for **network robustness**.
Robustness is a structural and topological property of a graph that quantifies the redundancy of information flow and connections between sets of nodes. It is fundamental to ensure that consensus can be achieved even in the presence of adversarial agents (up to F malicious nodes).
Before providing a formal definition of robustness for the entire graph, we define the robustness of a given subset of nodes $\mathcal{S}$ by adopting *Definition 4* from [3]:

Definition 3.1 ((r,s)-Edge Reachable Set [3])**.** Given a nontrivial digraph $\mathcal{D}$ and a nonempty subset of nodes $\mathcal{S}$, we say that $\mathcal{S}$ is an (r,s)-*edge reachable set* if there are at least s nodes in $\mathcal{S}$ with at least r in-neighbors outside of $\mathcal{S}$, where $r, s \in \mathbb{Z}_{\geq 0}$; i.e., given $\mathcal{X}_{\mathcal{S}}^{r} = \{i \in \mathcal{S} : |\mathcal{N}_i^{in} \setminus \mathcal{S}| \geq r\}$, then $|\mathcal{X}_{\mathcal{S}}^{r}| \geq s$ [3].

This definition implies that a set $\mathcal{S}$ is reachable (i.e., not isolated) if it contains at least s nodes, each receiving information from at least r neighbors located outside of $\mathcal{S}$.
Building upon this concept, *Definition 5* in [3] provides the formal definition of robustness for the entire graph:

Definition 3.2 ((r,s)-robustness [3])**.** A nonempty, nontrivial digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ on N nodes ($N \geq 2$) is (r,s)-*robust*, for nonnegative integers $r \in \mathbb{Z}_{\geq 0},\ 1 \leq s \leq N$, if for every pair of nonempty, disjoint subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ of $\mathcal{V}$ at least one of the following holds (recall $\mathcal{X}_{\mathcal{S}_k}^{r} = \{i \in \mathcal{S}_k : |\mathcal{N}_i^{in} \setminus \mathcal{S}_k| \geq r\}$ for $k \in \{1,2\}$):

1. $\left|\mathcal{X}_{\mathcal{S}_1}^{r}\right| = |\mathcal{S}_1|$;
2. $\left|\mathcal{X}_{\mathcal{S}_2}^{r}\right| = |\mathcal{S}_2|$;
3. $\left|\mathcal{X}_{\mathcal{S}_1}^{r}\right| + \left|\mathcal{X}_{\mathcal{S}_2}^{r}\right| \geq s$.

In other words, this notion of robustness measures the difficulty of isolating a group of nodes from the rest of the network. It is evident that as the parameters r and s increase, the network demands a significantly higher degree of connectivity. Having rigorously defined the concept of robustness, we must now determine the required level of network robustness in the presence of up to F malicious nodes. To this end, *Theorem 1* in [3] establishes the necessary and sufficient topological conditions:

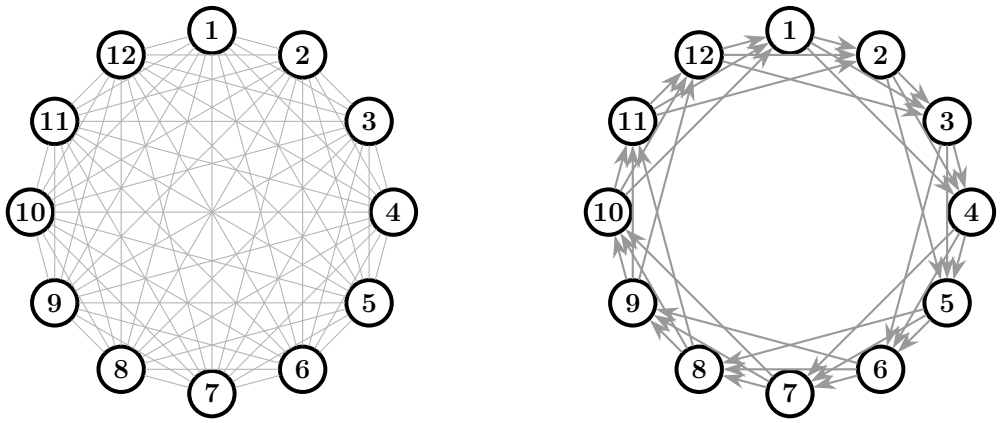
Theorem 3.1 (Theorem 1 of [3])**.** *Consider a time-invariant network modeled by a digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ where the adversaries satisfy the F-total malicious model. Suppose each normal node updates its value according to ARC-P with parameter F. Then, RAC is achieved if and only if the network topology is $(F+1, F+1)$-robust infinitely often asymptotically [3].*

Let us now analyze specific network topologies that are robust against F attacks (under the F-total malicious model). In particular, we choose to compare fully connected networks and directed circulant graphs. The objective of this analysis, along with the subsequent simulations, is to investigate how networks with different structural properties behave in the presence of attacks, and to what extent topology and connectivity influence the robustness and tracking speed within the resilient leader-follower dynamics.

Since robustness against F attacks requires $d_i \geq 2F + 1$ (3.1), in the following proofs we assume that the number of nodes is sufficiently large to guarantee this condition. Specifically, to satisfy (3.1), we require:

$$N > 2F + 1 \implies N \geq 2F + 2 \tag{3.13}$$

Furthermore, we recall that the considered networks do not contain self-loops.



(a) Fully connected topology **(b)** Circular Graph

Figure 3.1: Fully connected and circular graph topologies with 12 nodes.

$(F + 1, F + 1)$-robustness of fully connected networks:
It is straightforward to demonstrate that an N-node fully connected network is resilient to at most $F = \left\lfloor \frac{N-2}{2} \right\rfloor$ attacks. Since every node is connected to all other nodes ($|\mathcal{N}_i^{in}| = N - 1, \forall i \in \mathcal{V}$), all agents receive the complete set of network information, allowing them to successfully filter out malicious values.
Given that the network must contain at least $2F + 2$ nodes to satisfy (3.13), every node will inherently possess an in-degree $d_i \geq 2F + 1, \forall i \in \mathcal{V}$ (automatically fulfilling Assumption 3.1). In the limiting case where $N = 2F + 2$, the network is resilient to a maximum of $F = \left\lfloor \frac{N-2}{2} \right\rfloor$ attacks.
To rigorously prove that an N-node fully connected network is resilient to at most $F = \left\lfloor \frac{N-2}{2} \right\rfloor$ attacks (under the F-total malicious model), we must show that its topology is $(\left\lfloor \frac{N-2}{2} \right\rfloor + 1, \left\lfloor \frac{N-2}{2} \right\rfloor + 1)$-robust, according to Theorem 3.1. Let $\mathcal{S}_1$ and

$\mathcal{S}_2$, with $|\mathcal{S}_1| \leq |\mathcal{S}_2|$, be two disjoint, nonempty subsets ($\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$).

$$\begin{aligned} |\mathcal{S}_2| \geq |\mathcal{S}_1| \implies |\mathcal{S}_1| &\leq \left\lfloor \frac{N}{2} \right\rfloor, \quad \text{for } N \geq 2F+2 \text{ (3.1)} \\ \implies \left|\mathcal{N}_i^{in} \setminus \mathcal{S}_1\right| = |\mathcal{V} \setminus \mathcal{S}_1| &\geq N - \left\lfloor \frac{N}{2} \right\rfloor \geq 2F+2 - \left\lfloor \frac{2F+2}{2} \right\rfloor \\ &\geq F+1, \quad \forall i \in \mathcal{S}_1 \\ \implies \left|\mathcal{X}_{\mathcal{S}_1}^{F+1}\right| &= |\mathcal{S}_1|\,. \end{aligned} \tag{3.14}$$

Every node in $\mathcal{S}_1$ has at least $F+1$ neighbors strictly outside of $\mathcal{S}_1$. This holds true because, given $N \geq 2F+2$ from condition (3.13) and the assumption $|\mathcal{S}_1| \leq |\mathcal{S}_2|$, the number of nodes external to $\mathcal{S}_1$ is always greater than or equal to $F+1$. This result alone is sufficient to satisfy the first condition of the robustness definition (Theorem 3.1).
Since $N \geq 2F+2$ implies $F \leq \left\lfloor \frac{N-2}{2} \right\rfloor$, for an arbitrary N, F can be at most $\left\lfloor \frac{N-2}{2} \right\rfloor$. Consequently, the network achieves a maximum robustness level of $(\left\lfloor \frac{N-2}{2} \right\rfloor + 1, \left\lfloor \frac{N-2}{2} \right\rfloor + 1)$.

$(F+1, F+1)$-robustness of Circulant Digraphs $C_N(\{1,\ldots,2F+1\})$:
To formalize the construction of an N-node circulant digraph, denoted as $C_N(\mathcal{S}^{jump})$, we adopt an algebraic notation where nodes are identified by a zero-based index, i.e., $\mathcal{V} = \{0,\ldots,N-1\}$. By defining $\mathcal{S}^{jump}$ as the *jump set*, the circulant digraph is constructed such that each node i sends information to node $(i+k) \pmod N$, for all $k \in \mathcal{S}^{jump}$.

We will now demonstrate that a circulant digraph $C_N(\{1,\ldots,2F+1\})$, where $|\mathcal{S}^{jump}| = |\{1,\ldots,2F+1\}| = 2F+1$ and $N \geq 2F+2$ (3.13), is resilient against F attacks under the F-total malicious model.
It is worth noting that a circulant digraph $C_N(\{1,\ldots,2F+1\})$ with exactly $N = 2F+2$ nodes is, in fact, a fully connected network (since each node connects to $2F+1 = N-1$ neighbors), the robustness of which has already been proven above. For the general case where $N > 2F+2$, the robustness of this topology can be demonstrated through a worst-case scenario analysis:
A simultaneous attack by F compromised nodes is most effective when the maximum number of attackers is concentrated within the neighborhood of a single target node. In such a scenario, the information received by the victim node is saturated with anomalous values attempting to bypass the ARC-P filter parameterized by F. However, since the total number of attacks is strictly bounded by F, and every healthy node receives exactly $2F+1$ inputs, even arranging the malicious nodes consecutively will fail to compromise the network. The targeted normal node will still receive at least $F+1$ legitimate consensus states from the remaining healthy neighbors, allowing the ARC-P filter to safely discard the F malicious

inputs. Furthermore, the victim node will continue to broadcast its uncorrupted state to the rest of the network. Therefore, in a circulant digraph topology with degree $k \geq 2F+1$, it is impossible to isolate any legitimate node under the F-total malicious model. Consequently, the network is $(F+1, F+1)$-robust and, by Theorem 3.1, fully resilient to F attacks.

3.4 Convergence Speed Analysis for Leader-Follower Dynamics

We now modify the local control system (3.11) to incorporate the influence of an autonomous leader node. We operate under the standard assumption that **this leader is non-malicious**; therefore, it reliably broadcasts its initial condition in the modal space, z_0, to the network. Specifically, we can enforce the asymptotic tracking condition $\lim_{t\to\infty} z_i(t) = z_0 = Q^{-1}x_0(0)$ for all $i \in \mathcal{V}$ by modifying the differential equation governing the modal state to include the leader's contribution at the pinned nodes:

$$\dot{z}_i(t) = \begin{bmatrix} \sum_{j\in\mathcal{N}_i^{in}\setminus\mathcal{R}_{i,1}(t)} w_{ij}(t)[z_{j,1}(t) - z_{i,1}(t)] \\ \vdots \\ \sum_{j\in\mathcal{N}_i^{in}\setminus\mathcal{R}_{i,n}(t)} w_{ij}(t)[z_{j,n}(t) - z_{i,n}(t)] \end{bmatrix} + g_i\left(z_0 - z_i(t)\right) \tag{3.15}$$

where $g_i \geq 0$ is the pinning gain, as defined in (2.1.2).
Consequently, the differential equation governing the dynamic controller in (3.4) becomes:

$$\dot{\eta}_i(t) = (A+BK)\eta_i(t) - Qe^{Rt}\left[\Phi_F\left(\{z_j(t)\}_{j\in\mathcal{J}_i(t)}\right) + g_i\left(z_0 - z_i(t)\right)\right] \tag{3.16}$$

Consequently, we conducted numerical simulations to validate the theoretical derivations and to evaluate the comparative advantages and drawbacks of the network topologies described in the previous section. Specifically, the simulations were performed on networks comprising 12 nodes ($N = 12$), under the assumption of at most one malicious attacker ($F = 1$). The directed circulant graph employed in these tests features the minimal jump set $\mathcal{S}^{jump} = \{1,2,3\}$.
The local agent dynamics, adopted from Section V of [3], model a system of two coupled mass-spring oscillators. The corresponding continuous-time state-space matrices are given by:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0.5 & 0 & 0 \\ 0.25 & -0.25 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.25 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \tag{3.17}$$

alongside the feedback control gain matrix:

$$K = \begin{bmatrix} 22.75 & -12.75 & 8.75 & -14 \end{bmatrix}. \tag{3.18}$$

The initial conditions for each node were independently sampled from a uniform distribution, such that $x_{i,k}(0) \sim \mathcal{U}(-0.5, 0.5)$ for all $i \in \{0,1,\ldots,N\}$ and $k \in \{1,\ldots,n\}$. To ensure statistical reliability, we generated 20 distinct sets of initial conditions, which were kept consistent across all simulated topological configurations. Consequently, the results presented herein represent the arithmetic mean computed over these 20 independent trials.
From a practical standpoint, we define the α-convergence time t_{conv} as follows:

$$t_{conv} \in \mathbb{R}_+ : \|\delta(t)\|_2 = \|z(t) - \bar{z}_0\|_2 < \alpha, \quad \forall t > t_{conv} \tag{3.19}$$

where the global modal state $z(t)$ and the augmented leader state $\bar{z}_0$ are defined respectively as:

$$z(t) = \begin{bmatrix} z_1(t) \\ \vdots \\ z_N(t) \end{bmatrix} \in \mathbb{R}^{nN}, \quad \bar{z}_0 = \begin{bmatrix} z_0 \\ \vdots \\ z_0 \end{bmatrix} = \underline{1}_N \otimes z_0 \in \mathbb{R}^{nN}. \tag{3.20}$$

For the purpose of our numerical simulations, the convergence threshold is set to $\alpha = 0.001$.

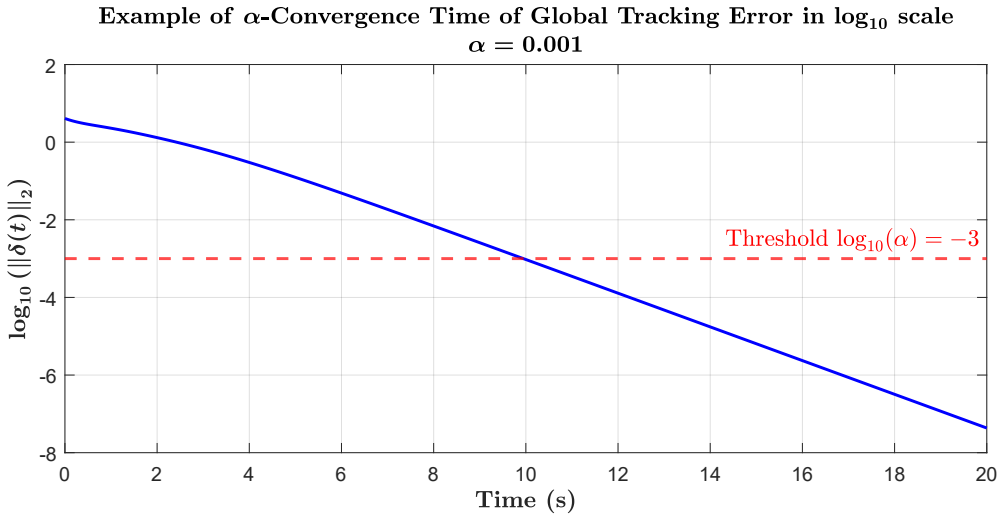


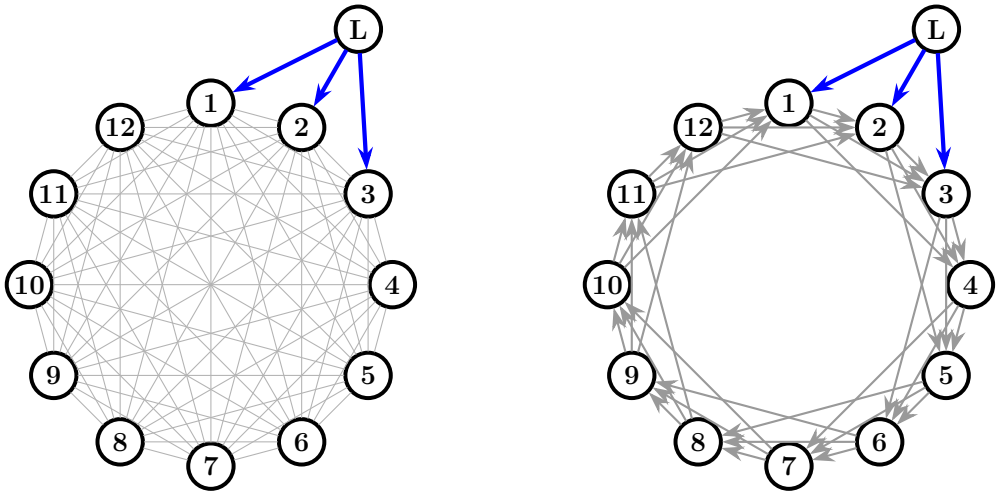
Figure 3.2: Evolution of the global tracking error $\log_{10}(\|\delta(t)\|_2)$ illustrating the definition of α-convergence.

3.4.1 Tracking speed without attack

To evaluate the leader tracking speed in the absence of attacks, we pin nodes 1, 2, and 3 by setting the pinning gains as follows:

$$g_i = \begin{cases} 1 & \text{if } i \in \{1,2,3\} \\ 0 & \text{if } i \in \{4,\dots,12\} \end{cases} \tag{3.21}$$

As we will demonstrate later, this specific configuration of pinned nodes ensures the resilience of the circulant graph when subjected to Leader-Follower dynamics. Given the superior connectivity of the fully connected network, we expect its tracking speed to surpass that of the circulant topology.



(a) Fully connected topology **(b)** Circular topology

Figure 3.3: Fully connected and circular graph topologies with 12 nodes and the first three nodes pinned to the leader.

The simulation results confirm our expectations: the circulant network achieves an average α-convergence in **48.48s**, whereas the fully connected network reaches it in **42.45s**.

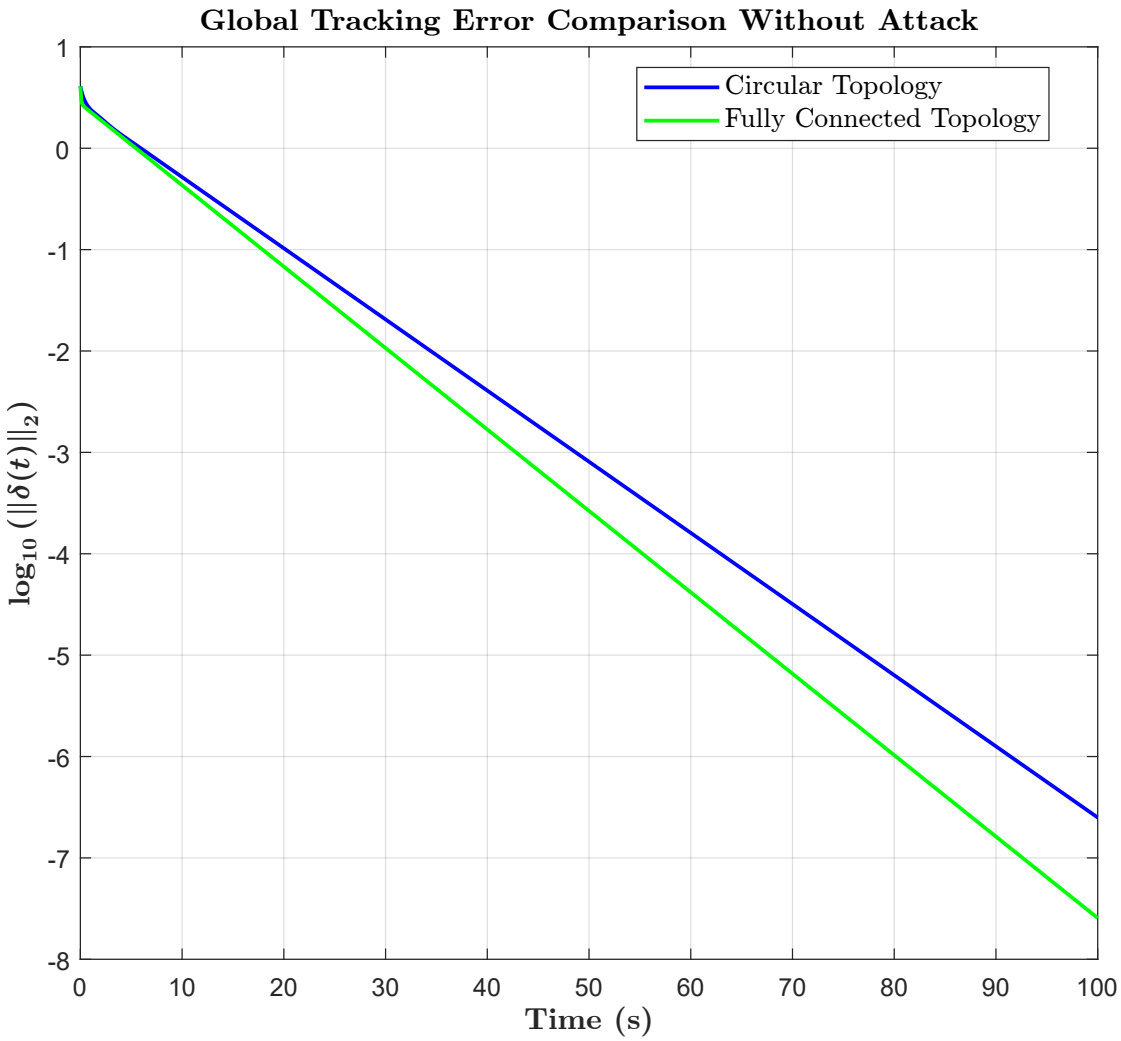


Figure 3.4: Evolution of the global tracking error $\log_{10}(\|\delta(t)\|_2)$ for the circular and fully connected networks. The plotted trajectories represent one specific realization out of the 20 independent trials used to compute the average α-convergence times.

3.4.2 Tracking speed with attack

The introduction of attacks raises new challenges: determining how many and which nodes should be pinned to the leader, as well as identifying the most effective or malicious attack strategies to rigorously stress-test the architecture's resilience. The complexity regarding pinned nodes stems from the fact that they broadcast the leader's state to the network; initially, this information will inevitably differ from the current states of the other agents. Therefore, the selected set of pinned nodes must guarantee that the leader's information reaches every node with sufficient

redundancy to prevent it from being entirely discarded by the ARC-P filter.
In other words, under the F-total malicious model, an $(F+1, F+1)$-robust network guarantees asymptotic tracking of the (non-malicious) leader if and only if it contains at least $2F+1$ spanning trees rooted at the leader. This requirement is justified by the fact that, under this topological condition, every node i not directly connected to the leader will receive at least $2F+1$ independent inputs carrying the leader's influence. Even in the worst-case scenario where the ARC-P algorithm filters out $2F$ of these values, at least one input propagating the leader's state will pass through, thereby driving the tracking dynamics.
Consequently, we can deduce that to guarantee tracking, at least $2F+1$ nodes must be pinned to the leader. Each pinned node essentially acts as a gateway, contributing to the existence of a distinct spanning tree that disseminates the leader's information throughout the network. An equivalent theoretical conclusion is established by Usevitch et al. in [5].
In the specific scenario analyzed in our simulations ($F = 1$), the network must encompass at least three spanning trees rooted at the leader, which fundamentally dictates the presence of three pinned nodes.
It now remains to determine exactly which $2F+1$ nodes should be pinned to the leader in the topologies under analysis.
For the fully connected network, since all nodes are mutually adjacent, the selection of the pinned nodes is completely arbitrary.
Conversely, for the circulant network, a specific configuration of pinned nodes must be identified to guarantee asymptotic tracking. We now demonstrate that a configuration featuring consecutive pinned nodes ensures tracking resilience:

Considering the general F-total malicious model and a directed circulant graph topology $C_N(\{1,\ldots,2F+1\})$, the optimal strategy for an attacker to isolate a specific node i is to maximize the injection of malicious information into that node. Let us assume the worst-case scenario where F nodes belonging to $\mathcal{N}_i^{in}$ are malicious, and all F malicious nodes are simultaneously pinned nodes. In this configuration, the genuine leader's information is transmitted by exactly $F+1$ healthy nodes.

Given the structural properties of the circulant network, the nodes in $\mathcal{N}_i^{in}$ are consecutive. If the remaining $F+1$ consecutive nodes are also pinned, the attack will fail because the leader's information will possess sufficient redundancy to prevent being entirely filtered out by the ARC-P algorithm. Consequently, since tracking is guaranteed by pinning all nodes in $\mathcal{N}_i^{in}$, we conclude that pinning $2F+1$ consecutive nodes in a circulant graph topology represents a sufficient condition to ensure resilience against F attacks under the F-total malicious model.

Regarding the attack vectors, the simulations were conducted using two distinct strategies: a tailored Byzantine attack and a uniform, non-Byzantine attack.

Since the threat model considered in [3] assumes an omniscient attacker with

full knowledge of the network state, a malicious node can customize its attack for each specific neighbor it transmits to. Among the various Byzantine strategies evaluated, the most effective—meaning the one that induces the greatest delay in tracking speed—is the *State Replication* (or inertia injection) attack.

In this scenario, the attacker transmits the target node's own current consensus state back to it. Because this malicious value perfectly matches the victim's local state, it systematically evades the ARC-P filter, which is designed exclusively to prune extreme outliers. Although this Byzantine attack cannot destabilize the system or prevent asymptotic synchronization, it alters the weights of the local convex combination, artificially inflating the inertia of the attacked node. This reduces the node's responsiveness to the Leader's signals, thereby degrading the overall performance in terms of tracking speed.

Conversely, the non-Byzantine attack consists of broadcasting a plausible, constant consensus value over time to all neighbors. In our simulations, the attacking nodes simply transmit a common initial state $z_a(0)$ without ever updating it. Under this scenario, the malicious nodes effectively act as antagonists to the genuine leader, attempting to substitute themselves as the reference trajectory.

We now present the simulation results for the networks under attack. Recalling that the pinned nodes are consistently the first three, we report the α-convergence times ($\alpha = 0.001$) for both network topologies under the F-total malicious model ($F = 1$). The presented results are the arithmetic means of 20 independent simulations conducted under the previously described identical conditions.

Table 3.1: Convergence time of the circulant graph network under attack

	Average Convergence Time(s), $\alpha = 0.001$	
Malicious Node	Byzantine Attack	Stubborn Malicious Attack
1	64,74	70.31
2	63.20	77.39
3	**70.50**	**90.90**
4	49,53	62.51
5	49,37	70.97
6	50.43	67.32
7	50.12	68.17
8	50.23	68.55
9	48.34	60.02
10	47.79	56.37
11	37.34	53.99
12	45.52	46.80
None	48.48	

Table 3.2: Convergence time of the fully connected network under attack

	Average Convergence Time(s), $\alpha = 0.001$	
Malicious Node	Byzantine Attack	Stubborn Malicious Attack
Unpinned Node	38.20	41.61
Pinned Node	**77.15**	**83.39**
None	42.45	

Simulation Results and Performance Analysis

The simulation results immediately reveal that the resilient system designed in [3] is particularly robust against attacks comprising non-replicated information; indeed, the Byzantine attack produces less impact than the simpler stubborn attack across both topologies. Furthermore, it is noteworthy that the fully connected network exhibits greater sensitivity to Byzantine attacks, presumably due to its high degree of connectivity.
Another intriguing result is that certain attack configurations targeting non-pinned nodes actually yield faster convergence times compared to the completely attack-free scenario. This counterintuitive phenomenon can be explained by the fact that the presence of extreme anomalous values effectively minimizes the risk of the ARC-P algorithm inadvertently filtering out the genuine leader's information.
The final significant observation concerns the drastic performance degradation observed in the circulant graph when the attacked node is node 3. We initially anticipated the worst-case performance to occur when the attacking node was node 1, given its status as a pinned node directly connected to the other pinned nodes (nodes 2 and 3).

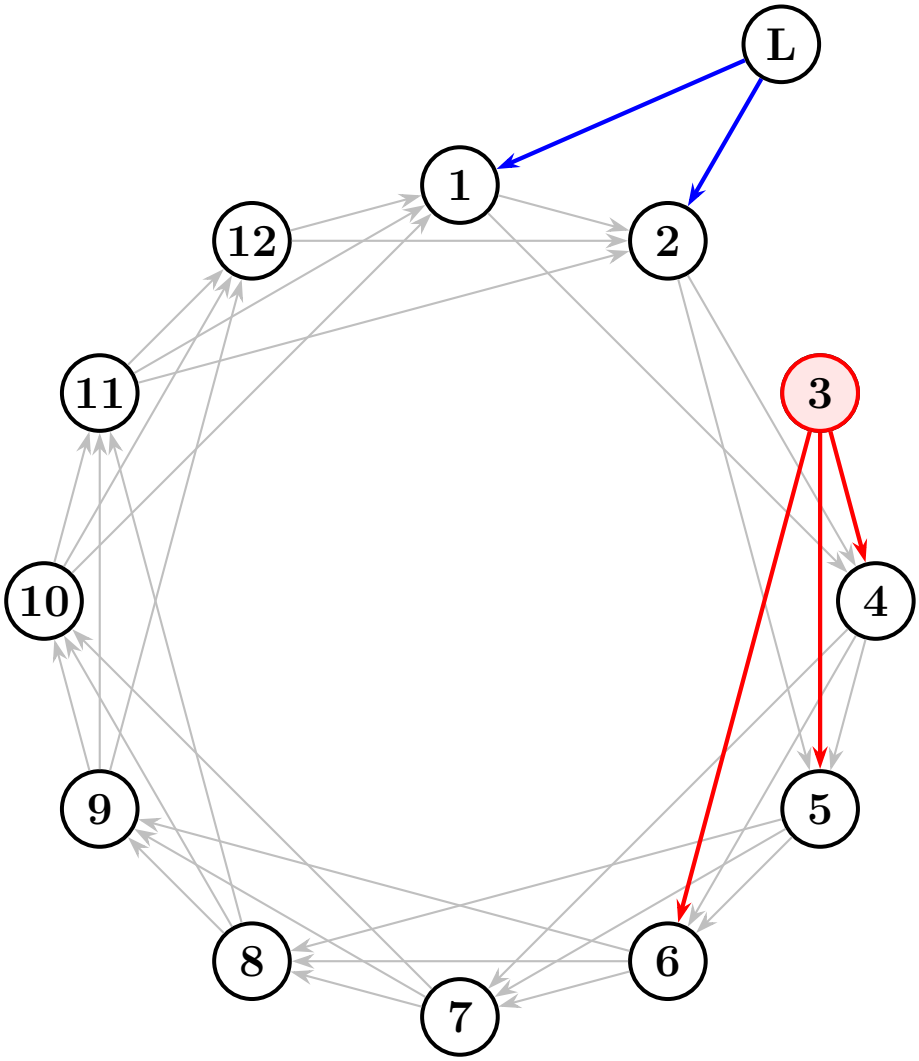


Figure 3.5: Worst Configuration

To fully comprehend this behavior, it is highly instructive to redraw the graph to explicitly illustrate how the flow of the leader's information is influenced by the circulant topology, and to highlight the shortest paths between the leader and every other node in the network:

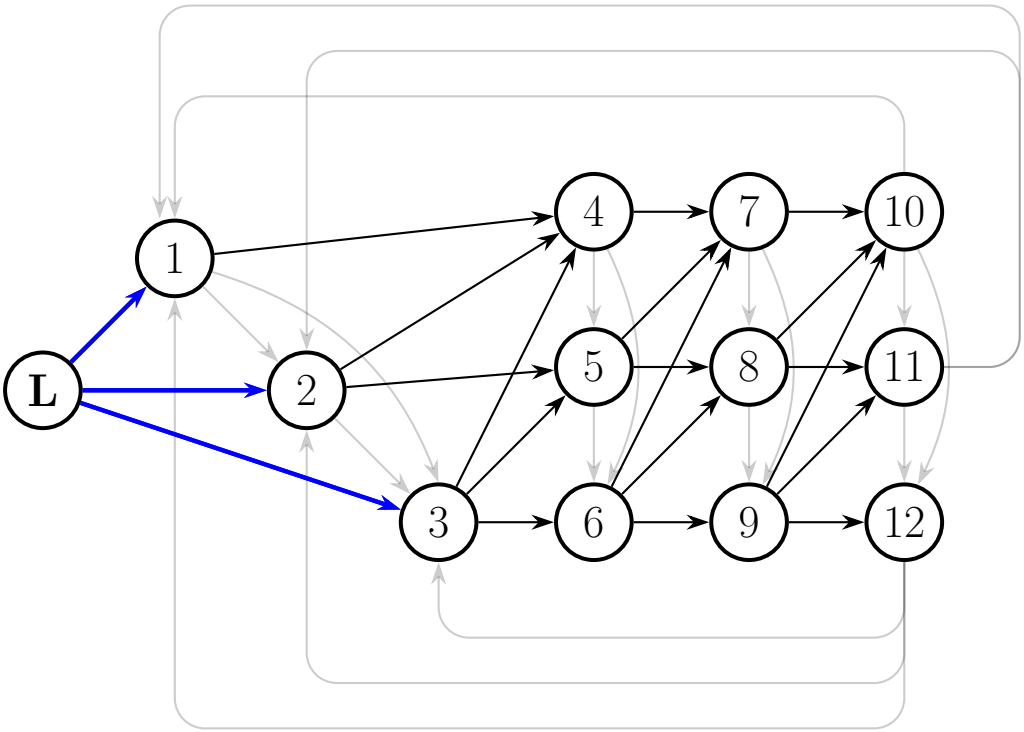


Figure 3.6: Circulant graph topology - focus on leader information flow.

It can be observed that the outgoing edges from the nodes, when partitioned into groups of three $(2F + 1)$, significantly contribute to the propagation of the leader's information flow. Specifically, node 3, with its three outgoing edges, exerts a predominant influence on the leader's information flow across the cut separating the pinned nodes from the remainder of the network. Consequently, the network topology, as constructed, renders the tracking information flow highly sensitive to the behavior of node 3.

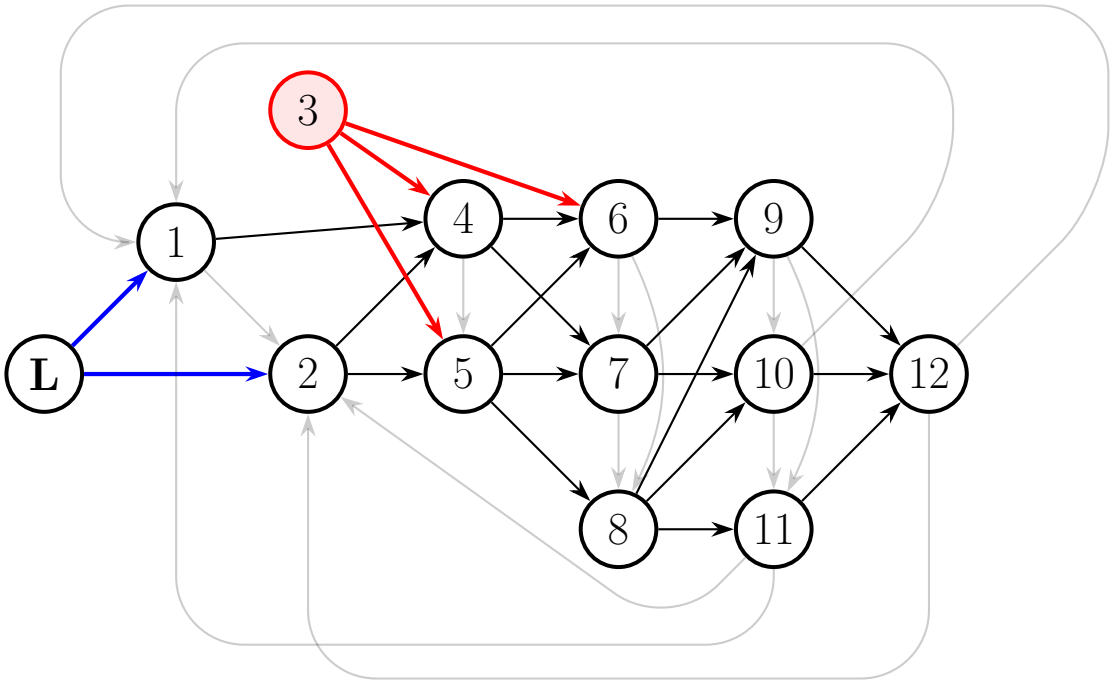


Figure 3.7: Circulant graph topology, node 3 malicious - focus on leader information flow.

Indeed, the attack perpetrated by node 3 effectively halves the throughput of the leader's information flow.
A final consideration pertains to the fact that the circulant structure contains backward-pointing edges, which propagate information in the opposite direction relative to the leader's signal flow. This feedback mechanism provides no advantage to the tracking performance; conversely, it exacerbates the inertia of the pinned nodes. These nodes are consequently decelerated by the information originating from the tail end of the network, which remains unaligned with the leader's state during the initial iterations of the algorithm.

3.5 A Robust and Scalable Method for Rapid Topology Design

Drawing upon the analyses conducted in the previous section, we can devise a scalable network model that integrates the advantages of a fully connected network (*i.e.*, attack symmetry) and a regular circulant graph (*i.e.*, bounded connectivity) to implement a leader-follower network dynamics optimized for maximum convergence speed.
We therefore commence with the circulant graph topology and systematically

remove the edges that oppose the flow of the leader's signal, thereby eliminating the inertia imposed on the pinned nodes by the outermost nodes of the network:

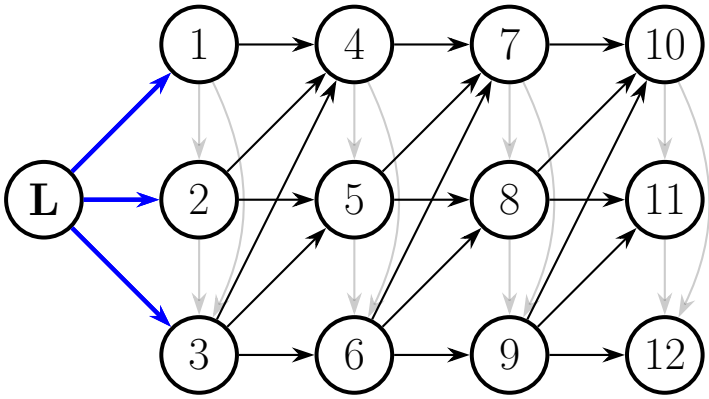


Figure 3.8: Circulant graph topology - focus on leader information flow.

At this stage, it is necessary to devise a strategy to maximize the throughput of the leader's signal. To achieve this, we can repurpose all the non-forward-pointing edges (represented as transparent edges in Figure 3.8) to actively propagate the leader's information throughout the network. This approach yields the following topology:

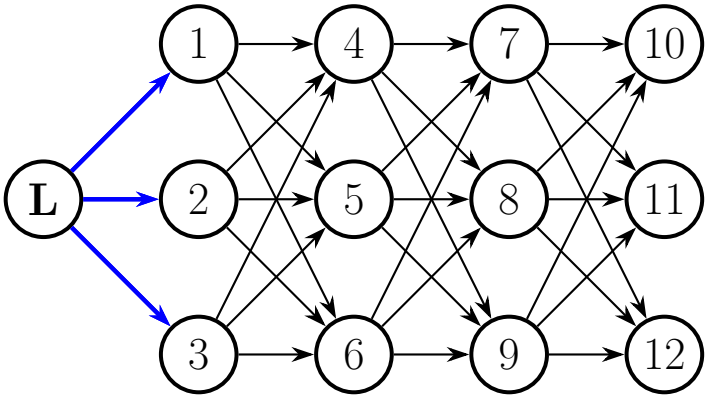


Figure 3.9: Circulant graph topology - focus on leader information flow.

This inherently acyclic design prevents the cascading propagation of attacks. If an attacker injects malicious data into a node at layer k, this corrupted information can exclusively reach the nodes at layer $k+1$. At this stage, thanks to the operation of the ARC-P algorithm, the anomalous values are immediately discarded before they can influence the dynamics of the healthy nodes. Consequently, the attack is absorbed and isolated at the subsequent layer, rendering a global compromise of

the system impossible.
Such fault compartmentalization is a crucial property, as it enables us to guarantee synchronization under a significantly more severe threat model: the F-**local** malicious model, which allows for the presence of up to F compromised nodes within the neighborhood of every single agent in the network. While this assumption typically necessitates graphs with an extremely high connection density, the absence of feedback loops in our topology allows the system to tolerate and neutralize this threat locally, thereby ensuring global resilience without having to maximize the overall connectivity. Another fundamental aspect of this architecture is its scalability, which can be adapted based on both the anticipated threat model and the total number of nodes to be connected (which we assume is always $\geq 2F + 2$, including the leader).
In the general case where a network must be resilient to F attacks, the number of nodes per layer is adjusted to be greater than or equal to $2F + 1$. This leads to a "horizontal" expansion of the network, continuing to guarantee resilience under the F-local malicious model. We will formally prove later that these modular topologies are indeed resilient to F attacks.
Conversely, when a large number of nodes must be connected, they are simply divided into multiple modules (or layers). This approach guarantees a structure that scales efficiently with the number of nodes while simultaneously maintaining resilience without demanding high connectivity. Indeed, it is imperative to note that although the connections in Figure 3.9 may appear more intricate compared to the ones in figure 3.5, the in-degree of every node is strictly bounded by $2F + 1$ in both cases.

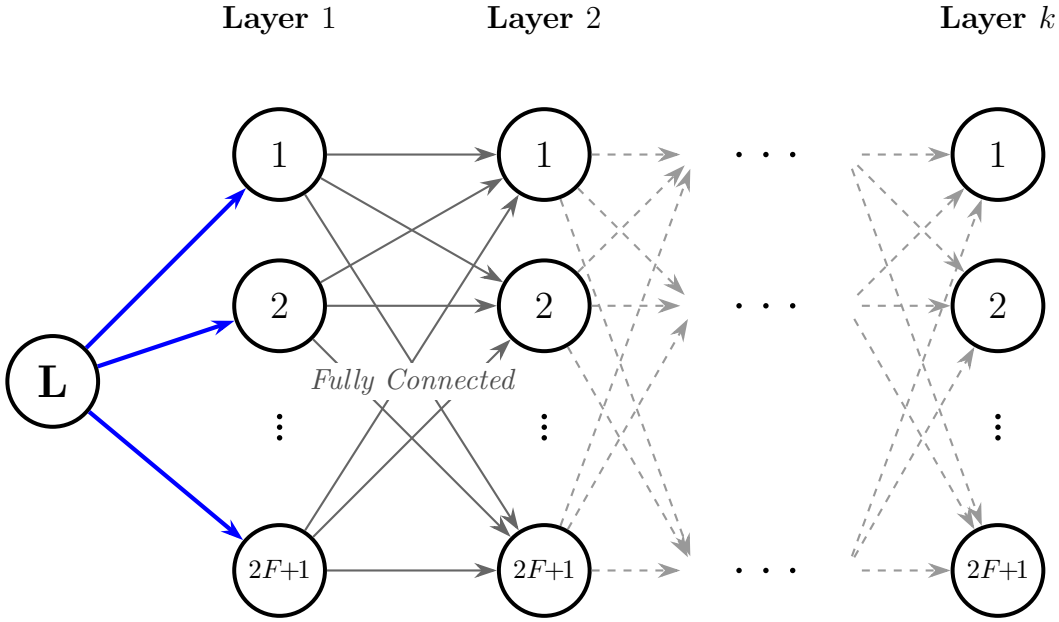


Figure 3.10: Generalized architecture of the proposed scalable modular DAG topology, structured in k sequential layers.

We will now formally demonstrate that this scalable topology is indeed resilient to F attacks under the F-local malicious model, by applying Theorem 5 from [4] regarding the construction of robust graphs:

Theorem 3.2 (Theorem 5 of [4])**.** *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be an (r, s)-robust digraph (with $s \geq 1$). Then the digraph $\mathcal{D}' = (\mathcal{V} \cup \{v_{new}\}, \mathcal{E} \cup \mathcal{E}_{new})$, where v_{new} is a new vertex added to $\mathcal{D}$ and $\mathcal{E}_{new}$ is the directed edge set related to v_{new}, is (r, s)-robust if the in-degree $d_{v_{new}} \geq r + s - 1$.*

Let node $\mathcal{L}$ denote the leader. We consider the initial state as a trivial digraph $\mathcal{D}$ containing only the source (leader), which acts as the robust root of the topology. We then iteratively add the pinned nodes. Let us add the first pinned node, denoted as node $\{1\}$, as illustrated in Figure (3.11). To satisfy the condition that the leader is never filtered by ARC-P we model the pinning as $2F{+}1$ parallel edges from the leader to node 1. The resulting digraph is $\mathcal{D}' = \left(\mathcal{V} \cup \{1\}, \mathcal{E} \cup \{\underbrace{(\mathcal{L} \to 1), \ldots, (\mathcal{L} \to 1)}_{2F+1 \text{ times}}\}\right)$.

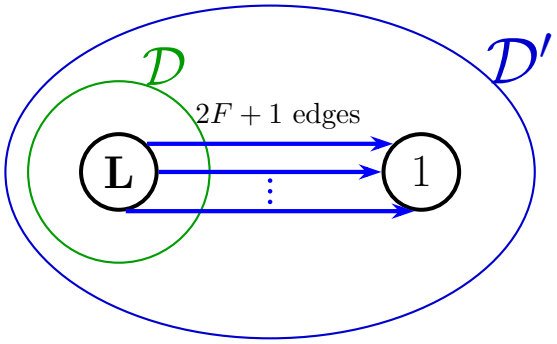


Figure 3.11: Topology construction via Theorem 3.2 - Appending the first pinned node to the robust root.

We check the condition for $(F+1, F+1)$-robustness:

$$r + s - 1 = F + 1 + F + 1 - 1 = 2F + 1$$

Since the in-degree of the new node is $d_1 = 2F + 1$, the condition $d_1 \geq r + s - 1$ is satisfied, and $\mathcal{D}'$ remains $(F+1, F+1)$-robust.
The same reasoning applies recursively to the other pinned nodes (2 to $2F + 1$) connected to the leader. Furthermore, for any subsequent node added to the network according to our proposed design strategy—where every new module or node connects to the $2F + 1$ node of the previous module—the in-degree is always $d_{v_{\text{new}}} = 2F + 1$. Consequently, the robustness condition $d_{v_{\text{new}}} \geq r + s - 1 = 2F + 1$ is consistently satisfied at every step of the construction.

We can synthetically redraw the network shown in Figure 3.9 by employing a block-level representation, where each rectangle denotes a module (or layer) of interconnected agents, and the arrows represent the fully connected inter-module pathways:

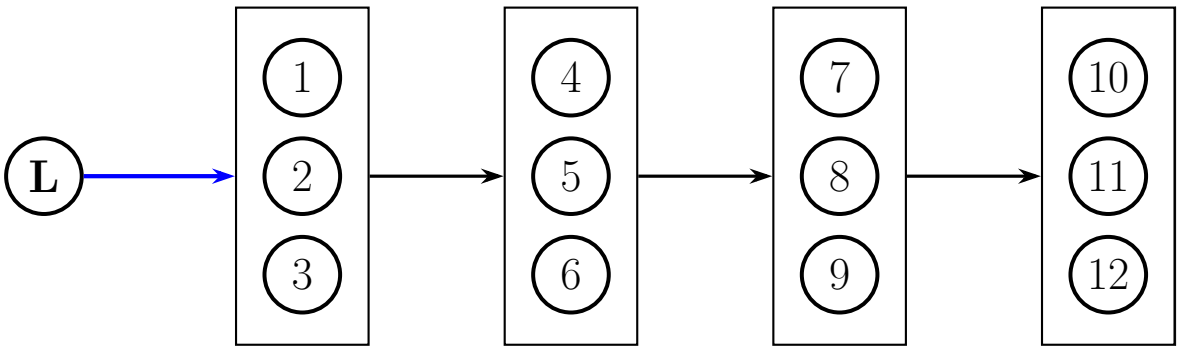


Figure 3.12: Modular Synthesis View of Figure 3.9.

Since the construction Theorem 3.2 imposes the in-degree of the appended nodes

as its sole structural constraint, it also enables the design of structural variations from the initially proposed cascaded network topology.
Considering the node in-degree as the only hardware or technical limitation–conceptually analogous to having receiver communication channels that can be selectively tuned to the signal of one specific node over another–we can apply the same design philosophy to generate a more generalized network architecture. Specifically, this architecture would consist of a directed spanning tree of interconnected modules, provided that each module strictly contains at least $2F + 1$ nodes.

An example of a spanning tree topology is presented in the figure below (Figure 3.13):

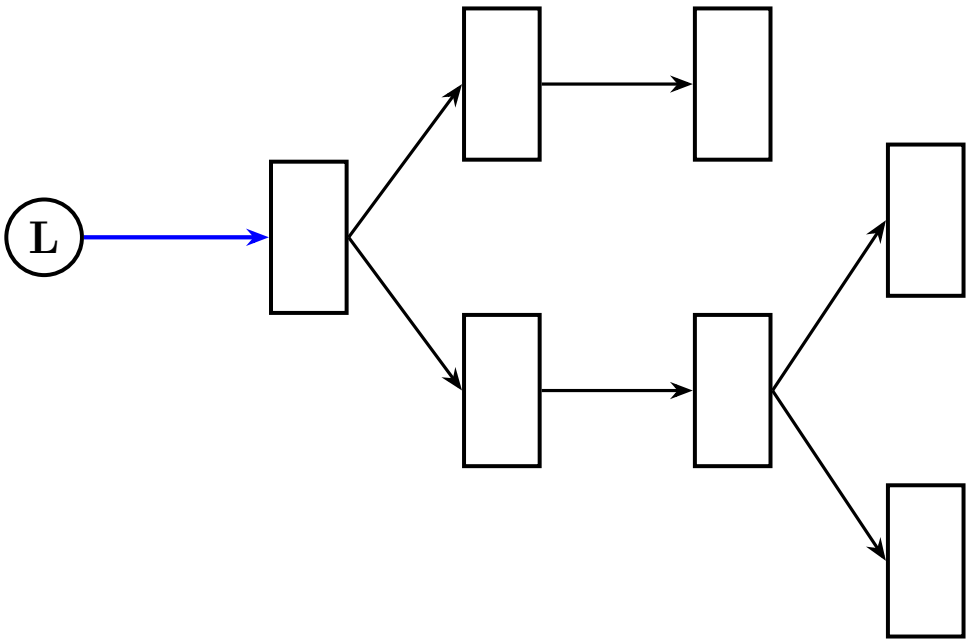


Figure 3.13: Example of a modular spanning tree topology represented synthetically.

In this architecture, the branches are specifically formed by a multitude of overlapping connections. A detailed representation of a single module branching into two distinct output pathways is provided below.

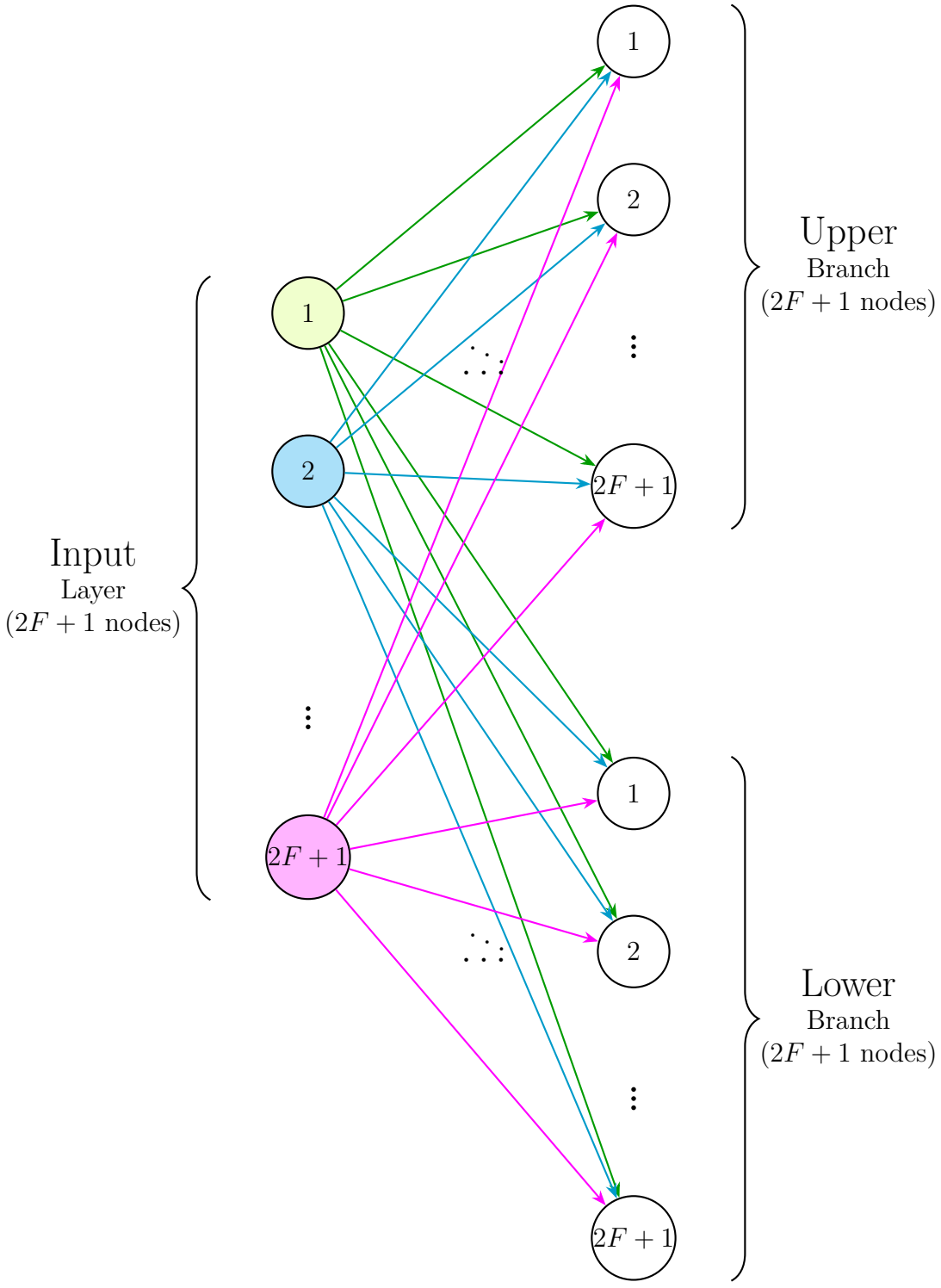


Figure 3.14: Detailed representation of an inter-module connection splitting into two output branches.

Simulations on the Optimized Topology

Numerical simulations were conducted to evaluate the α-convergence time of the optimal network topology derived in the preceding paragraphs (Figure 3.9), aiming to benchmark its performance against the robust topologies analyzed in the previous section (*i.e.*, the fully connected network and the directed circulant graph).
These simulations were executed employing the identical agent dynamics and initial conditions as those detailed in Section 3.4. Consistent with our established methodology, the presented results represent the arithmetic mean computed over 20 independent trials.
Unlike the topologies investigated in Section 3.4, the proposed optimal network satisfies a more stringent resilience classification, specifically the 1-local malicious model. Accordingly, the attacks were simulated by compromising the first node within each individual agent layer. This arbitrary selection of the attacking nodes is entirely justified mathematically, as the proposed topology inherently possesses attack symmetry, rendering the system's resilience uniform regardless of which specific node is compromised within a given layer.

Table 3.3: Convergence time of topology in figure 3.9

Attack Type	Average Convergence Time(s), $\alpha = 0.001$
None	9.74
Byzantine	**13.78**
Stubborn	13.38

As demonstrated by the results, the optimized topology consistently yields superior performance across all evaluated scenarios. By comparing the worst-case convergence time recorded in the new simulations (subjected to a Byzantine attack under the 1-local model) against the best-case convergence time from the simulations in the previous section (subjected to a Byzantine attack targeting node 11 in the circulant graph topology, under the 1-total model), we achieve a minimum performance improvement of $\frac{t_{old,best}}{t_{new,worst}} \times 100\% = \frac{37.34}{13.78} \times 100\% \simeq 271\%$.

3.6 Conclusions

This chapter demonstrated the efficacy of the Asymptotic Resilient Consensus Protocol (ARC-P) in achieving resilient synchronization for multi-agent systems under adversarial attacks. By sorting local neighborhood data and pruning extreme anomalous values, ARC-P effectively neutralizes malicious injections and ensures safe convergence to the leader's trajectory.
However, our analysis revealed that algorithmic filtering alone is insufficient without

an optimized network topology. Standard architectures, such as fully connected or circulant graphs, suffer from information bottlenecks and structural inertia. To address this, we proposed a novel class of scalable, directed acyclic topologies. By providing the necessary structural redundancy ($2F + 1$ independent pathways) and maximizing the forward throughput of the leader's signal, our design drastically accelerates tracking convergence. Crucially, these performance improvements are achieved while maintaining a strictly bounded local connectivity footprint.

The following chapter will focus on extending these resilience concepts to the cooperative leader-follower dynamics proposed by Lewis et al. [1]. Since Lewis's framework accommodates any arbitrary system matrix A, backward state projections become mathematically unfeasible. Consequently, the primary objective of the next chapter will be to investigate whether integrating ARC-P to directly evaluate instantaneous trajectory data can effectively eliminate the restrictive requirement of marginally stable dynamics, thereby facilitating the development of robust, higher-order resilient consensus protocols for a broader class of autonomous multi-agent systems.

Chapter 4

Resilient High Order Consensus in Leader-Follwer Network Model

The cooperative control architecture proposed by Lewis et al. [1] provides a highly effective, static distributed feedback mechanism for leader-follower synchronization. However, in the presence of adversarial nodes or compromised communication channels, this standard linear consensus protocol is inherently vulnerable to malicious data injection. The primary objective of this chapter is to investigate the integration of ARC-P into the Lewis framework, aiming to achieve a resilient higher-order consensus without sacrificing the structural simplicity of the original static controller.

Unlike previous discussions that may have explored discrete-time formulations to accommodate algorithmic estimators, this chapter explicitly adopts the continuous-time dynamics of the Lewis model. The rationale behind this modeling choice is twofold: first, the ARC-P filter is natively formulated in continuous time; second, and most importantly, utilizing a continuous-time framework allows for a rigorous, one-to-one performance comparison with the continuous-time resilient architecture of LeBlanc et al. [3] analyzed in Chapter 3. Maintaining this mathematical coherence is strictly necessary to evaluate the true impact of the control logic on the network's synchronization capabilities.

However, as the mathematical and simulative analyses presented in the following sections will reveal, achieving resilient synchronization by applying ARC-P directly within the state space—rather than the decoupled modal space used in [3]—introduces critical performance trade-offs. The absence of a dynamic local controller capable of enforcing monotonic error decrease inevitably leads to slower convergence rates compared to the Scardovi-Sepulchre framework when subjected

to identical threat models.

Nevertheless, exposing and understanding this performance discrepancy mathematically motivates our primary innovative contribution. Unlike the architecture in [3], which strictly requires the agents' autonomous dynamics to be at most marginally stable, the ARC-P-augmented Lewis framework proposed here can successfully synchronize nodes characterized by strictly unstable, divergent trajectories (such as ramp dynamics driven by a double integrator). By coupling this robust static control formulation with the optimized Directed Acyclic Graph (DAG) topology designed in Chapter 3, we demonstrate that the system successfully mitigates uninformed, rudimentary attacks under an F-local malicious threat model, offering a clear and novel trade-off between control complexity, dynamic restrictions, and convergence speed.

The chapter is organized as follows: Section 4.1 details the integration of the ARC-P filter into the continuous-time Lewis framework, formally deriving the global error dynamics and the robust constant coupling gain. Section 2 provides a comparative analysis of the convergence rates against the Scardovi-Sepulchre framework, highlighting the structural performance discrepancies. Section 3 presents the simulative evaluation of the proposed framework, validating its resilience using unstable intrinsic dynamics under various attack scenarios. Finally, Section 4 summarizes the chapter's findings and outlines the core theoretical trade-offs.

4.1 Lewis Framework with ARC-P Filter

Let us now return to the continuous-time leader-follower dynamics proposed by Lewis et al. [1] and investigate the integration of the ARC-P filter. Since ARC-P operates in continuous time, we adopt the continuous-time analogue of the model analyzed in Chapter 2, which is extensively detailed in the third chapter of [1].

In the continuous-time leader-follower dynamics, the leader node evolves according to an autonomous linear trajectory:

$$\dot{x}_0(t) = Ax_0(t) \tag{4.1}$$

where $x_0(t) \in \mathbb{R}^n$ is the state of the leader and $A \in \mathbb{R}^{n\times n}$ is the system dynamics matrix. Similarly to the discrete-time case, the continuous-time dynamics of the i-th follower node is governed by:

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t) \tag{4.2}$$

where $x_i(t) \in \mathbb{R}^n$ is the state vector of the i-th node (assumed locally measurable), $u_i(t) \in \mathbb{R}^m$ is the control input, and the pair (A, B), with $B \in \mathbb{R}^{n\times m}$, is assumed to be stabilizable.

The local distributed control input is synthesized as:

$$u_i(t) = cK\varepsilon_i(t) \tag{4.3}$$

where $c \in \mathbb{R}_+$ is the global coupling gain, $K \in \mathbb{R}^{m\times n}$ is the feedback gain matrix, and $\varepsilon_i(t) \in \mathbb{R}^n$ is the *local neighborhood tracking error*. By integrating the ARC-P filter, the standard linear consensus summation is replaced by the nonlinear filtering operator Φ_F:

$$\varepsilon_i(t) = \Phi_F\left(\{x_j(t)\}_{j\in\mathcal{J}_i(t)}\right) + g_i(x_0(t) - x_i(t)) \tag{4.4}$$

where $g_i \in \mathbb{R}$ is the local pinning gain.

By using the control system with the error computed as in (4.4), we implement a higher-order resilient consensus. Indeed, to implement a first-order resilient consensus, it would be necessary to employ a control system such as the one described in [2], which leverages the matrices Q and R to project the node states into the past and compare their respective initial values within the mode space of the dynamics. However, this approach would contradict two key properties of the system proposed in [1]: the fact that the dynamics matrix A is not required to be stable, and the static nature of the controller.

Since A is not assumed to be marginally stable, computing the continuous projection of a node's state back to the initial instant $t = 0$ becomes practically unfeasible due to numerical ill-conditioning. If A contains unstable modes (i.e., eigenvalues with positive real parts), the corresponding state trajectories will naturally diverge to infinity as $t \to \infty$. Consequently, the matrix exponentials used for forward and backward projection (e^{Rt} and e^{-Rt}) will contain terms that exponentially diverge to infinity and others that decay to zero. In a practical digital implementation, attempting to reconstruct the constant initial state by multiplying an unbounded state vector $x(t)$ by a vanishing projection matrix e^{-At} inevitably leads to severe numerical instability, overflow, and catastrophic loss of precision.

Implementing a dynamic controller via $\eta(t)$ (see 3.1) would significantly complicate the static control analyzed in Chapter 2. One of our main objectives is, in fact, to leverage the simplicity of the control system presented in the third chapter of [1].

Consequently, this requires us to impose limitations on the attacks addressed by the filter in (4.4), which can no longer be executed within the dynamics mode space, unlike the simulations presented in Section 3.4 of this document.
Nevertheless, since the established objective is to create a system that is robust against sparse and constant attacks on the communication channels (Section 2.2), the system analyzed in this section is sufficient to achieve this goal.

Before deriving the global network dynamics, it is important to formally define the topological matrices $\mathcal{A}$, D, L, and G, and to highlight how this notation transitions from the discrete-time case. In the discrete-time models analyzed

previously, the topology was conventionally modeled using an adjacency matrix often denoted as E, from which the discrete Laplacian was constructed. To maintain notational consistency with the continuous-time algebraic graph theory utilized in [1], we denote the continuous-time adjacency matrix as $\mathcal{A} \in \mathbb{R}^{N\times N}$, where each entry $a_{ij} \geq 0$ represents the weight of the directed edge (ν_j, ν_i). The diagonal in-degree matrix is defined as $D = \text{diag}(d_1, \ldots, d_N) \in \mathbb{R}^{N\times N}$, where each element represents the sum of the incoming edge weights, $d_i = \sum_{j=1}^{N} a_{ij}$. The continuous-time graph Laplacian is then naturally given by $L = D - \mathcal{A}$. Finally, $G = \text{diag}(g_1, \ldots, g_N) \in \mathbb{R}^{N\times N}$ is the pinning gain matrix, containing the terms $g_i \geq 0$ which map the direct tracking connections to the leader.

Crucially, because the ARC-P filter dynamically discards extreme input values based on instantaneous state discrepancies, the effective communication network becomes time-varying. Consequently, the adjacency matrix $\mathcal{A}(t)$, the degree matrix $D(t)$, the Laplacian $L(t)$, and the augmented Laplacian $L(t)+G$ change dynamically at each time instant.

Let us define the global follower state vector $x = \begin{bmatrix} x_1^T & \cdots & x_N^T \end{bmatrix}^T \in \mathbb{R}^{nN}$, the augmented leader state $\bar{x}_0 = \underline{1}_N \otimes x_0 \in \mathbb{R}^{nN}$, and the global disagreement error $\delta(t) = x(t) - \bar{x}_0(t) \in \mathbb{R}^{nN}$. The closed-loop global error dynamics are given by:

$$\dot{\delta}(t) = [I_N \otimes A - c(L(t) + G) \otimes BK]\, \delta(t) \tag{4.5}$$

To achieve asymptotic synchronization ($\lim_{t\to\infty} \delta(t) = 0$), the matrix governing this dynamic must be Hurwitz.

To address this issue, Lewis et al. [1] propose the *local Riccati Design of Synchronizing Protocols* (formalized in Theorem 3.1), which systematically yields a suitable pair (c, K). Similarly to the methodology presented in the second chapter, this approach leverages Algebraic Riccati Equations (AREs) to synthesize a feedback gain matrix K that not only stabilizes the pair (A, B) but also maximizes the synchronization region—that is, the permissible range for selecting the coupling gain c.

Theorem 4.1 ((Theorem 3.1 of [1])). *Consider local distributed control protocols (4.3). Suppose (A, B) is stabilizable and let design matrices $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ be positive definite.*
Design the state variable feedback control gain K as

$$K = R^{-1}B^T P, \tag{4.6}$$

where P is the unique positive definite solution of the control algebraic Riccati equation

$$0 = A^T P + PA + Q - PBR^{-1}B^T P. \tag{4.7}$$

Then, under the assumption that there is a directed path (not necessarily unique) from the leader node to every follower node, the global disagreement error dynamics

(4.5) is asymptotically stable if the coupling gain satisfies

$$c \geq \frac{1}{2\min_{i\in\mathcal{V}} Re(\lambda_i)} \tag{4.8}$$

with $\lambda_i\,(i \in \mathcal{V})$ the eigenvalues of $L+G$. Then, all agents synchronize to the leader node stationary trajectory [1].

Although Theorem 4.1 provides a straightforward design methodology, the ARC-P filter induces continuous topological switching. Implementing a time-varying coupling gain based on the instantaneous eigenvalues, such as

$$c(t) \geq \frac{1}{2\min_{i\in\mathcal{V}} \mathrm{Re}(\lambda_i(t))},$$

is practically unfeasible, as individual nodes cannot observe the global network topology changes in real-time. Therefore, we must define a constant coupling gain c that is structurally robust to these topological variations.

To obtain this constant robust gain, we leverage the novel topology designed in Chapter 3 (Section 3.5). The networks generated through this design possess the crucial property of being Directed Acyclic Graphs (DAGs). Let $\mathcal{S}_p \subseteq \mathcal{V}$ denote the set of pinned nodes. In a DAG topology, the i-th node only accepts information from nodes with an index $j < i$. Furthermore, the $2F+1$ pinned nodes act as pure sources (in-degree $a_{ij} = 0,\ \forall i \in \mathcal{S}_p$), and all unpinned followers have a zero pinning gain ($g_i = 0,\ \forall i \in \mathcal{V} \setminus \mathcal{S}_p$).

Thus, the adjacency matrix $\mathcal{A}$ takes the form of a strictly lower triangular matrix with a null diagonal, composed as follows:

$$\mathcal{A} = \begin{bmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ a_{2F+2,1} & \cdots & a_{2F+2,2F+1} & 0 & \cdots & 0 & 0 \\ a_{2F+3,1} & \cdots & a_{2F+3,2F+1} & a_{2F+3,2F+2} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & 0 & 0 \\ a_{N,1} & \cdots & a_{N,2F+1} & a_{N,2F+2} & \cdots & a_{N,N-1} & 0 \end{bmatrix} \tag{4.9}$$

The pinning matrix G, containing only the positive gains for the pinned nodes,

assumes this diagonal form:

$$G = \begin{bmatrix} g_1 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & g_{2F+1} & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{4.10}$$

Consequently, it is straightforward to construct the Laplacian matrix $L = D - \mathcal{A}$, yielding:

$$L = \begin{bmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ -a_{2F+2,1} & \cdots & -a_{2F+2,2F+1} & d_{2F+2} & \cdots & 0 & 0 \\ -a_{2F+3,1} & \cdots & -a_{2F+3,2F+1} & -a_{2F+3,2F+2} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & d_{N-1} & 0 \\ -a_{N,1} & \cdots & -a_{N,2F+1} & -a_{N,2F+2} & \cdots & -a_{N,N-1} & d_N \end{bmatrix} \tag{4.11}$$

Finally, the augmented Laplacian $L + G$ governing the global dynamics is a lower triangular matrix:

$$L + G = \begin{bmatrix} g_1 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & g_{2F+1} & 0 & \cdots & 0 & 0 \\ -a_{2F+2,1} & \cdots & -a_{2F+2,2F+1} & d_{2F+2}(t) & \cdots & 0 & 0 \\ -a_{2F+3,1} & \cdots & -a_{2F+3,2F+1} & -a_{2F+3,2F+2} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & d_{N-1}(t) & 0 \\ -a_{N,1} & \cdots & -a_{N,2F+1} & -a_{N,2F+2} & \cdots & -a_{N,N-1} & d_N(t) \end{bmatrix} \tag{4.12}$$

From (4.12), we can derive an explicit and general formula for the eigenvalues $\lambda_i(t)$ of $(L(t) + G)$, which are strictly real and correspond to the diagonal entries:

$$\lambda_i(t) = d_i(t) + g_i \tag{4.13}$$

Specifically, the eigenvalues associated with the pinned nodes are simply equal to g_i, whereas those associated with the unpinned nodes are equal to their time-varying in-degree $d_i(t)$ (due to the ARC-P filter operation):

$$\lambda_i(t) = \begin{cases} g_i & \text{if } i \in \mathcal{S}_p \\ d_i(t) & \text{if } i \in \mathcal{V} \setminus \mathcal{S}_p \end{cases} \tag{4.14}$$

To systematically find a constant solution for $\min_{i\in\mathcal{V}} \mathrm{Re}(\lambda_i(t))$, we can simplify the framework by assuming binary values. Let $g_i \in \{0,1\}$, meaning:

$$g_i = \begin{cases} 1 & \text{if } i \in \mathcal{S}_p \\ 0 & \text{if } i \in \mathcal{V} \setminus \mathcal{S}_p \end{cases} \tag{4.15}$$

Similarly, we assume unweighted adjacency edges $a_{ij} \in \{0,1\}$:

$$a_{ij} = \begin{cases} 1 & \text{if } (\nu_j \to \nu_i) \in \mathcal{E} \\ 0 & \text{if } (\nu_j \to \nu_i) \notin \mathcal{E} \end{cases} \tag{4.16}$$

so that the in-degree $d_i(t)$ corresponds directly to the number of active incoming edges at time t.

To ensure resilience under the targeted threat model, the topology guarantees that the initial connectivity satisfies $d_i \geq 2F + 1$ for all unpinned nodes $i \in \mathcal{V} \setminus \mathcal{S}_p$. Because the ARC-P filter discards at most $2F$ inputs, the minimum remaining in-degree in the absolute worst-case scenario is bounded by $d_i(t) \geq (2F+1) - 2F = 1$.

Therefore, for any topology designed using the framework presented in Chapter 3 (Section 3.5), we have $\min_{i\in\mathcal{V}} \mathrm{Re}(\lambda_i(t)) = 1$. Consequently, we can universally satisfy the stability condition of Theorem 4.1 by choosing a constant coupling gain c which satisfy:

$$c \geq \frac{1}{2\min_{i\in\mathcal{V}} \mathrm{Re}(\lambda_i(t))} \implies c \geq \frac{1}{2} \tag{4.17}$$

4.2 Comparison of Convergence Rates against the Scardovi–Sepulchre Framework

To validate the effectiveness of the proposed distributed higher-order resilient consensus framework, numerical simulations were conducted. Consistent with the methodology established in Chapter 3.1 (Section 3.4), these simulations employ the continuous-time state-space matrices A and B adopted from Section V of [3]:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0.5 & 0 & 0 \\ 0.25 & -0.25 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.25 \end{bmatrix}. \tag{4.18}$$

The feedback gain matrix K was synthesized following the design procedure outlined in Theorem 4.1, selecting the weighting matrices as $Q = I_n$ and $R = 1$. This configuration yields:

$$K = \begin{bmatrix} -0.7233 & 1.0864 & -0.1735 & 3.1131 \end{bmatrix}. \tag{4.19}$$

The local control protocol detailed in Section 4.1 is applied to every node in the network. The global coupling gain is set to $c = 1$.

The network topology implemented in these simulations corresponds to the optimized architecture depicted in Figure 3.9. Specifically, it consists of 12 follower nodes partitioned into consecutive layers of 3 nodes each, where inter-layer communication is fully connected.

To rigorously test the system's resilience, a constant (stubborn) attack is injected into the communication channels of nodes 1, 4, and 7. This specific configuration perfectly aligns with the 1-local malicious threat model, as exactly one node is compromised within each topological layer. Specifically $a_1(t) = a_1 = 50 \cdot \underline{1}_n$, $a_4(t) = a_4 = \underline{1}_n$, $a_7(t) = a_7 = 10 \cdot \underline{1}_n$.

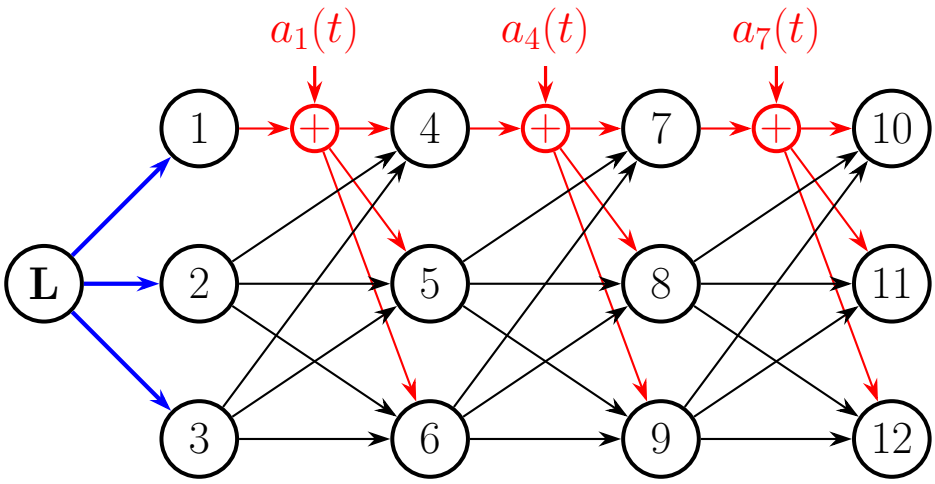


Figure 4.1: Representation of the network model used in simulations

Recalling the definition of the α-convergence time t_{conv} as:

$$t_{conv} \in \mathbb{R}_+ : \|\delta(t)\|_2 = \|x(t) - \bar{x}_0(t)\|_2 < \alpha, \quad \forall t > t_{conv} \tag{4.20}$$

the performance results are evaluated based on the average α-convergence time (setting $\alpha = 0.001$) computed over 20 independent simulation trials. For each trial, the initial conditions of the individual nodes are randomly sampled from a uniform distribution:

$$x_{i,k}(0) \sim \mathcal{U}(-0.5, 0.5), \quad \forall i \in \{0,1,\dots,N\}, \forall k \in \{1,\dots,n\} \tag{4.21}$$

The simulation results demonstrate that, in the attack-free scenario, the system achieves α-convergence in **116.47s**. When subjected to the aforementioned attacks, the average α-convergence time increases to **133.25s**.

These findings indicate that the framework described in Chapter 3 of [1], when

augmented with the ARC-P filter to achieve higher-order resilient consensus, falls short of the performance achieved by the resilient architecture proposed in [3]. This performance discrepancy can be attributed to two primary structural differences: first, the underlying consensus dynamics are of a higher order rather than first-order; second, the consensus protocol operates directly within the state space, as opposed to the decoupled modal space of the agents' dynamics.

Finally, a further element contributing to the suboptimal performance of this framework is the absence of a dynamic local controller capable of ensuring the monotonic convergence of the tracking error.

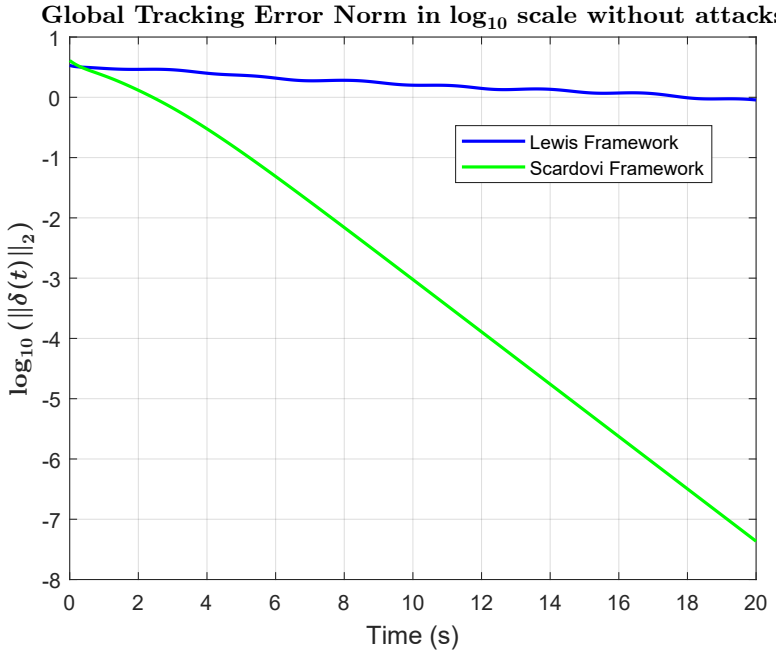


Figure 4.2: Comparison of the global tracking error norm evolution between the Lewis and Scardovi control frameworks

4.3 Performance Evaluation under Ramp Dynamics

Given that the networked leader-follower framework proposed by Lewis et al. [1] enables the synchronization of agents with potentially unstable intrinsic dynamics (i.e., a double integrator), we investigate the behavior of the resilient system designed in Section 4.1 using nodes characterized by ramp dynamics. Specifically, we evaluate the system's performance under various attacks injected into the communication channels of nodes 1, 4 and 7 as described in 2.2, operating strictly

under a 1-local malicious threat model.

For the numerical simulations, the agents' unstable intrinsic dynamics were modeled using the state-space matrices:

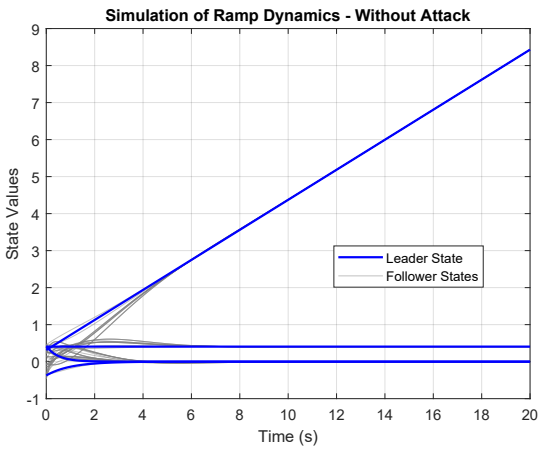
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{4.22}$$

The feedback gain matrix K was synthesized following the design procedure outlined in Theorem 4.1, selecting the weighting matrices as $Q = I_n$ and $R = 1$. This configuration yields:

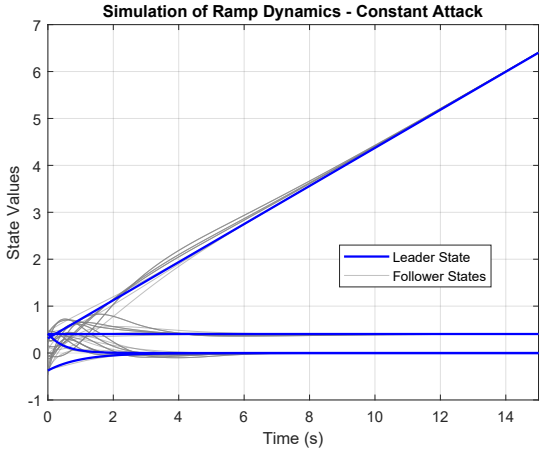
$$K = \begin{bmatrix} 1 & 1.8346 & 0.1271 & 0.1117 \end{bmatrix}. \tag{4.23}$$

Finally, the global coupling gain was set to $c = 1$.
The network topology selected for the simulations is identical to the one employed in the previous section (depicted in Figure 4.1).
To evaluate the robustness of the proposed framework, three distinct types of attack vectors are injected into the network:

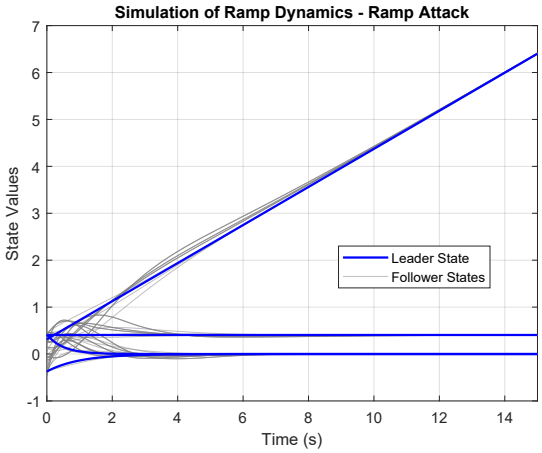
- **Constant attacks:** $a_1(t) = a_1 = 50 \cdot \underline{1}_n$, $a_4(t) = a_4 = \underline{1}_n$, $a_7(t) = a_7 = 10 \cdot \underline{1}_n$
- **Ramp attacks:** $a_1(t) = 50t \cdot \underline{1}_n$, $a_4(t) = t \cdot \underline{1}_n$, $a_7(t) = 10t \cdot \underline{1}_n$
- **Sinusoidal attacks:** $a_1(t) = 50\sin(10t) \cdot \underline{1}_n$, $a_4(t) = \sin(10t) \cdot \underline{1}_n$, $a_7(t) = 10\sin(10t) \cdot \underline{1}_n$



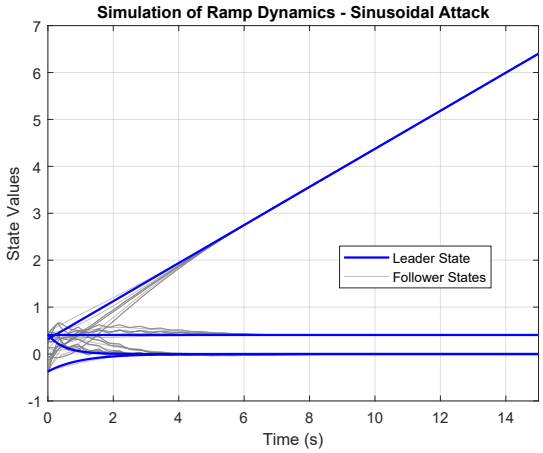
(a) Ramp dynamics convergence without any attack



(b) Ramp dynamics convergence with constant attack on communication channels



(c) Ramp dynamics convergence with ramp attack on communication channels



(d) Ramp dynamics convergence with sinusoidal attack on communication channels

Figure 4.3: Evaluation of distributed control system designed in 4.1 under different rudimentary attacks

A visual inspection of the above figures clearly reveals that the ramp attack has the most severe impact on degrading the tracking dynamics. This phenomenon can be attributed to the nature of the attack itself: although uninformed, the ramp signal closely simulates the genuine state dynamics of the agents, thereby making the malicious data injection perceived as a highly plausible state trajectory.

4.4 Conclusions

In conclusion, regarding leader-follower tracking dynamics, the framework designed in this chapter successfully enables resilient tracking against sparse and rudimentary attacks on the inter-node communication channels.

While the framework analyzed in Chapter 3.1 achieves the same objective with significantly superior performance, the resilient architecture proposed by LeBlanc et al. [3] introduces critical trade-offs. Specifically, it requires a dynamic local control system—making it structurally more complex than the static approach proposed by Lewis et al. [1]—and imposes a strict constraint on the agents' autonomous dynamics, which cannot be strictly unstable (i.e., divergent).

Therefore, the control framework analyzed in this chapter, paired with the topology designed in Section 3.5, represents a highly effective solution for networked synchronization problems where the leader (sharing the same autonomous dynamics as the followers) is characterized by an unstable system matrix A, and the network is subjected to uninformed, rudimentary attacks under an F-local malicious threat model. This validity stems from its reliance on a simple, straightforward local control protocol and a topology that structurally guarantees a minimum connectivity bound, ensuring robust synchronization without compromising the convergence rate.

Chapter 5

Final Remarks and Future Developments

This thesis addressed the critical challenges of security and resilience in Cyber-Physical Systems, focusing on cooperative control architectures based on Leader-Follower dynamics. The strict reliance on network communication exposes these architectures to severe vulnerabilities; therefore, the primary objective of this work was to analyze, model, and overcome the limitations of traditional consensus protocols subjected to malicious attacks on their communication channels.

The obtained results demonstrate that ensuring robust and scalable resilience requires simultaneous interventions on two fronts: the optimization of the topological infrastructure and the innovation of control laws.

5.1 Summary of Contributions

A first fundamental result pertained to the topological design. The analysis of network architectures traditionally considered robust (such as regular circulant graphs) highlighted significant vulnerabilities related to the structural inertia caused by backward-pointing edges. To resolve this critical issue, a novel class of modular and scalable networks based on Directed Acyclic Graphs was designed. This topology proved capable of compartmentalizing the cascading propagation of malicious data, guaranteeing system synchronization under the severe F-local malicious threat model. Numerical simulations confirmed that such an architecture maximizes tracking speed, yielding a 271% performance improvement compared to traditional resilient circulant topologies, while maintaining a local connectivity degree strictly bounded to $2F + 1$.

In the realm of control theory, the implementation of the nonlinear ARC-P filter proved essential to overcome the ineffectiveness of linear algorithmic estimators

(such as Online Gradient Descent) in the face of recursive error accumulation. However, the reference literature applied this filter by resorting to modal space projections backward in time, an approach that imposes a stringent theoretical constraint: the agents' intrinsic dynamics must be at most marginally stable.

To overcome this obstacle and generalize the control scheme, an innovative *higher-order resilient consensus* framework was proposed. The direct integration of the ARC-P filter into the state space, within the static distributed control protocol of Lewis et al., successfully removed the marginal stability constraint. The results demonstrated the capability of this new framework to resiliently synchronize agents characterized by intrinsically unstable and divergent dynamics (such as ramp dynamics driven by a double integrator) in the presence of rudimentary and uninformed attacks, while simultaneously preserving the architectural simplicity of the static controller.

5.2 Future Developments

Despite the promising theoretical and simulative results, the analysis highlighted important trade-offs that pave the way for several directions for future work.

Primarily, integrating the nonlinear filter directly into the instantaneous state space guarantees resilience against uninformed attacks, but exposes the system to vulnerabilities against advanced, informed attacks. An adversary with complete knowledge of the network state could inject malicious data specifically engineered to avoid appearing as extreme values. Since the ARC-P's *sorting and pruning* logic cannot detect such attacks, a natural future development will entail extending the higher-order resilient consensus to tolerate informed attacks. This milestone could be achieved by hybridizing the ARC-P filter with active, residual-based detection mechanisms (e.g., leveraging the global neighborhood disagreement error), or by introducing adaptive dynamic weights that penalize nodes exhibiting suspicious behavior over time.

Finally, an additional avenue of investigation should address the optimization of the convergence speed for the higher-order framework. The absence of a dynamic local controller (such as the one present in the Scardovi framework) inevitably causes a slowdown in achieving global consensus. Future research could focus on the synthesis of hybrid controllers designed to guarantee a monotonic decrease of the tracking error without reintroducing restrictions on the intrinsic dynamics of the system's state matrix.

Bibliography

[1] Frank L. Lewis, Hongwei Zhang, Kristian Hengster-Movric, and Abhijit Das. *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. Springer Science & Business Media, 2014 (cit. on pp. 1, 5, 7, 9, 16, 17, 19, 35, 62–67, 70, 71, 74).

[2] Luca Scardovi and Rodolphe Sepulchre. «Synchronization in networks of identical linear systems». In: *Automatica* 45.11 (May 2008), pp. 2557–2562 (cit. on pp. 3, 35, 36, 65).

[3] Heath J. LeBlanc and Xenofon Koutsoukos. «Resilient First-Order Consensus and Weakly Stable, Higher Order Synchronization of Continuous-Time Networked Multiagent Systems». In: *IEEE Transactions on Control of Network Systems* 5.3 (Sept. 2018), pp. 1219–1231 (cit. on pp. 3, 35, 36, 38–42, 45, 49, 51, 63, 64, 69, 71, 74).

[4] Heath J. LeBlanc, Haotian Zhang, Xenofon Koutsoukos, and Shreyas Sundaram. «Resilient asymptotic consensus in robust networks». In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE. 2012, pp. 1629–1634 (cit. on pp. 41, 57).

[5] James Usevitch and Dimitra Panagou. «Resilient Leader-Follower Consensus to Arbitrary Reference Values in Time-Varying Graphs». In: *IEEE Transactions on Automatic Control* 65.4 (2020), pp. 1755–1762. DOI: 10.1109/TAC.2019.2934954 (cit. on p. 49).