Modeling and Control of Cyber-Physical Systems Distributed Control of a Multi-agents Magnetic Levitation System

Project II

Fabio Veroli
fabio.veroli@studenti.polito.it

Dario Lupo
s336550@studenti.polito.it

Fabio Brugiafreddo
fabio.brugiafreddo@studenti.polito.it

Department of Control and Computer Engineering
Politecnico di Torino

1 Introduction

1.1 Multi-agents Magnetic Levitation System

This report presents the design and analysis of a distributed control protocol for a multi-agent magnetic levitation (maglev) system, composed of one leader agent and six follower agents. The system is modeled using linearized state-space equations, with the assumption that the state variables are not directly measurable. The main objective is to achieve cooperative tracking, ensuring that the global disagreement error asymptotically converges to zero.

Two control architectures are explored:

- A distributed regulator based on neighborhood observers, where agents utilize information from their neighbors.
- A regulator based on local observers, where each agent relies solely on its own measurements.

The performance of these strategies is evaluated under different reference signals provided by the leader agent, including constant, ramp, and sinusoidal trajectories. The study also analyzes the effects of measurement noise, coupling gains, and the choice of weighting matrices on the system's behavior, offering a comparative evaluation of both control approaches.

A critical factor influencing system performance is the choice of communication network topology. This aspect affects convergence rate, robustness to noise and faults, and the scalability of the system. The report examines several network configurations, analyzes their effects, and justifies the selection of the most appropriate topology.

The project is organized as follows:

- **Section 2**: Introduces the mathematical modeling of the maglev system.
- **Section 3**: Describes the design of the distributed control protocols.
- **Section 4**: Presents the numerical results and the comparative performance analysis.

2 Mathematical Modeling of the Maglev System

The dynamics of each magnetic levitation (maglev) agent are modeled using a set of nonlinear physical equations, which are then linearized to obtain a tractable state-space representation suitable for control design.

By linearizing the dynamic equations of each maglev, the following state-space model are obtained for each agent:

$$\begin{cases} \dot{x}_i = Ax_i + Bu_i \\ y_i = Cx_i \end{cases} \tag{1}$$

The system matrices are:

$$A = \begin{bmatrix} 0 & 1 \\ 880.87 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ -9.9453 \end{bmatrix}, \quad C = [708.27 \quad 0], \quad D = [0]$$

This linear model forms the foundation for the design and analysis of the distributed control protocols developed in the following sections.

2.1 State Variables

The system's state variables are defined as follows:

- **Position** (x_1): Represents the vertical displacement of the levitated object (e.g., the ferromagnetic ball) from its equilibrium position.
- **Velocity** (x_2)**:** The time derivative of position ($\frac{dx_1}{dt}$), representing the instantaneous vertical speed of the object.

In this system, both state variables are not directly measurable, which necessitates the use of state observers to estimate them from available output measurements.

These variables form the state vector used in the state-space representation. The position is indirectly observed via the system output, as it appears in the output equation through the output matrix C. The velocity, however, does not appear in the output equation and only influences the system through the dynamics governed by the state matrix A; therefore, it must be reconstructed mathematically through observer design techniques.

2.2 Leader-Follower Framework

The leader-follower framework organizes the multi-agent maglev system into a hierarchical structure:

- **Leader** (S_0): The leader is responsible for generating the reference trajectory for the entire system. It acts as an input source and does not adapt its behavior based on feedback from the follower agents.
- **Followers** (S_i**,** $i = 1, ..., N$): The follower agents work cooperatively to track the leader's trajectory using distributed information exchange and local control strategies. Specifically:
  - **Distributed Coordination**: Each follower S_i utilizes:
    - Network topology, represented by the adjacency matrix $\mathcal{A}_d$, which encodes neighbor relationships.
    - Optional direct reference access from the leader through the pinning matrix $\mathbf{G}$, which determines which agents are directly ”pinned” to the leader.
  - **Decentralized Control**: Followers implement distributed control laws to asymptotically minimize the tracking error $\|x_i(t) - x_0(t)\|$. This is achieved using only local information and neighbor-to-neighbor communication, without relying on a centralized controller.

- **Stability Conditions**: The stability and convergence of the system are influenced by:
  - The design of the control gain and observer parameters
  - The structural properties of the communication network, as described by $\mathbf{A}_d$ and $\mathbf{G}$

This architecture offers a hybrid control strategy, where centralized reference generation (via the leader S_0) is combined with decentralized execution (by the followers S_i). This ensures scalability, robustness, and flexibility, making it well-suited for large-scale cyber-physical systems. While the leader defines the global objective, each follower maintains local autonomy, contributing to the overall system performance and resilience.

2.3 Network Topologies in Multi-Agent Maglev Systems

The performance of distributed control in multi-agent systems critically depends on the underlying communication network topology, which defines how information is exchanged among agents. Our system was tested with several types of topologies, including:

- **Chain Topology**: Agents communicate in a linear sequence, with the leader acting as the first node. This simple structure is resource-efficient but suffers from cumulative latency and poor fault tolerance, as any broken link can partition the network (Fig. 1).
- **Star Topology**: In our implementation, the leader agent S_0 communicates only with one follower node, which functions as a central hub, directly communicating with all follower agents S_i. This setup introduces a single point of failure and places a high bandwidth demand on the central follower. We opted for this configuration to ensure that the assumption, that only a small subset of followers can observe the leader, is satisfied (Fig. 2).
- **Tree Topology**: A hierarchical structure in which information flows from the leader through intermediate layers of followers. This configuration offers a compromise between scalability and latency but remains vulnerable to failures in upper-layer nodes, which may disconnect downstream agents. In our setup, the leader communicates with two followers, which are then connected in a chain with the remaining agents (Fig. 3).
- **Complete topology**: The leader node S_0 connects exclusively to the first follower S_1, while all followers $\{S_1, \ldots, S_N\}$ form a fully connected subgraph ($a_{ij} > 0 \ \forall i, j \in \{1, \ldots, N\}, i \neq j$). This hybrid configuration combines centralized leader access through S_1 with decentralized all-to-all follower coordination, offering strong inter-agent synchronization while maintaining minimal leader connectivity requirements and robustness to individual agent failures (Fig. 4).

The choice of topology involves a trade-off between convergence speed (which benefits from dense connectivity) and resource constraints (which favor sparse connections). Numerical simulations in Section 4 will quantify these effects for the selected topology.

2.4 Network Topology Representation in Multi-Agent Systems

The communication structure of a multi-agent system is formally described using three fundamental matrices, which collectively define the system's topological and information-sharing architecture:

- **Adjacency Matrix** ($\mathcal{A}$): This is a square matrix of dimension $N \times N$, where N is the number of follower agents. It encodes the directed communication links between agents. An entry $a_{ij} > 0$ indicates that agents i receives information from agent j via a directed edge (v_j, v_i) and the weight of this link. The diagonal elements are zero ($a_{ii} = 0$), as self-loops are excluded in this model.
- **Pinning Matrix** (G): This is a diagonal matrix of size $N \times N$ used to indicate which follower agents are pinned to the leader, that is, which followers have direct access to the leader's state information.

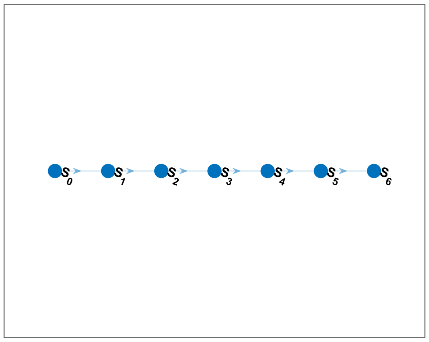

Figure 1: Chain network

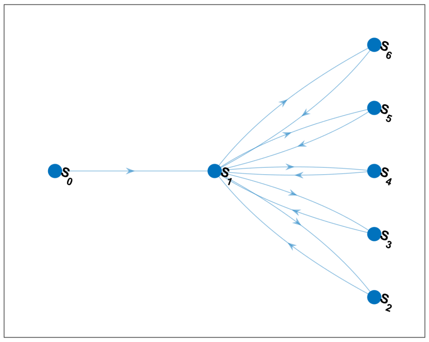

Figure 2: Star network

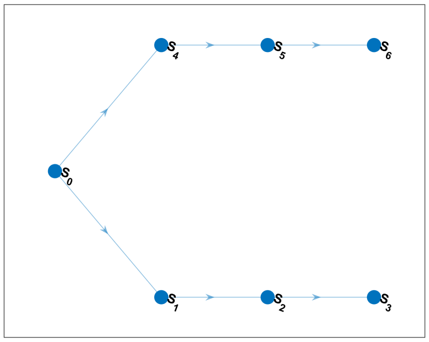

Figure 3: Tree network

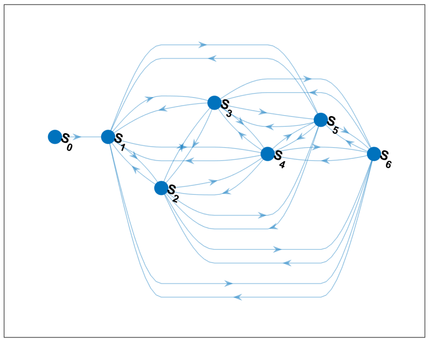

Figure 4: Complete network

- **Degree Matrix** (D): This is a diagonal matrix of size $N \times N$, where each diagonal entry d_{ii} represents the in-degree of node i, that is, the sum of the weight of all edges directed into node i. It quantifies the total incoming influence each follower receives from its neighbors.

These matrices form the foundation for computing the graph Laplacian, which is central to the stability analysis and design of distributed control protocols:

$$L = D - \mathcal{A} \tag{2}$$

This structured representation enables rigorous analysis of the network's communication patterns and their influence on the collective behavior of the system, serving as the theoretical basis for the design and evaluation of distributed control strategies in multi-agent cyber-physical systems. Summary of roles:

- $\mathcal{A}$ captures inter-agent connectivity
- G models leader-follower interactions
- D characterizes each agent's connectivity strength
- L supports stability and convergence analysis

3 Distributed Control Protocol Design

3.1 Control Problem Formulation

The cooperative tracking problem consists of designing distributed control protocols such that:

$$\lim_{t \to \infty} \|x_i(t) - x_0(t)\| = 0, \quad \forall i \in \{1, \dots, N\} \tag{3}$$

Since the system states are not directly measurable, we must design a dynamic regulator composed of: a state observer to reconstruct unmeasured states, and a state-feedback controller to drive the dynamics.

We implement and analyze two architectures:

1. A distributed regulator based on a neighborhood observer, and
2. A decentralized regulator using local observers only.

The leader node does not receive any control input u, as its behavior is uninfluenced by other agents. However, to enforce reference tracking, a local linear state-feedback controller with gain K_r is applied, resulting in the leader's closed-loop dynamics: $A_r = A - BK_r$

3.2 Distributed Observer Structure

In the neighborhood-based distributed observer, each follower S_i estimates its state via:

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\xi_i \tag{4}$$

where $c > 0$ is the coupling gain, $F \in \mathbb{R}^{n\times p}$ is the observer gain, and ξ_i is the neighborhood output estimation error computed as:

$$\xi_i = \sum_{j=1}^{N} a_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(\tilde{y}_0 - \tilde{y}_i) \tag{5}$$

where $\tilde{y}_i = y_i - \hat{y}_i$ is the output estimation error for agent i.

The observer gain matrix $F \in \mathbb{R}^{n\times p}$ is designed via:

$$F = PC^\top R^{-1} \tag{6}$$

with $R > 0$ a user-selected diagonal matrix, and P the unique positive definite solution of the Algebraic Riccati Equation (ARE):

$$AP + PA^T + Q - PC^T R^{-1} CP = 0 \tag{7}$$

where $Q > 0$ is also user-defined and diagonal.

3.3 Local Observer Structure

In the local observer approach, each agent S_i estimates its state using only its own measurement error $\tilde{y}_i$:

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\tilde{y}_i \tag{8}$$

In this case, the observer gain F is computed numerically to ensure that the matrix $(A + cFC)$ is Hurwitz, i.e., all its eigenvalues have strictly negative real parts, thus guaranteeing:

$$\lim_{t\to\infty} \tilde{x}_i(t) = 0 \tag{9}$$

This is achieved through the following optimization procedure:

1. Define a cost function $J(F)$ penalizing:
   - eigenvalues with positive real parts (to ensure stability),

    - large Frobenius norm $||F||_F$ (for regularization)

2. Solve the unconstrained optimization problem:

$$F^* = \arg \min_{F \in \mathbb{R}^{n \times p}} J(F) \tag{10}$$

3. Apply quasi-Newton optimization method.

3.4 Control Protocol Design

For both observer architectures, the distributed control law is:

$$u_i = cK\varepsilon_i \tag{11}$$

where the consensus tracking error ε_i is:

$$\varepsilon_i = \sum_{j=1}^{N} a_{ij}(\hat{x}_j - \hat{x}_i) + g_i(\hat{x}_0 - \hat{x}_i) \tag{12}$$

The feedback gain $K \in \mathbb{R}^{m \times n}$ is designed using the Linear Quadratic Regulator (LQR) method:

$$K = R^{-1}B^{\top}P \tag{13}$$

where P is the unique positive definite solution of the Algebraic Riccati Equation (ARE):

$$A^T P + PA + Q - PBR^1 B^T P = 0 \tag{14}$$

with $Q > 0$ and $R > 0$ user-defined diagonal matrices.

To ensure overall convergence, the coupling gain c must satisfy:

$$c \geq \frac{1}{2 \min_{i \in \mathcal{N}} Re(\lambda_i)} \tag{15}$$

where λ_i are the eigenvalues of the matrix $L + G$, and $\mathcal{N}$ denotes the set of follower agents.

3.5 Global Closed-Loop Dynamics & Solution

The closed-loop dynamics for the distributed neighborhood observer regulator is described by the following equation:

$$\begin{bmatrix} \dot{\delta} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A_c & B_c \\ 0 & A_o \end{bmatrix} \begin{bmatrix} \delta \\ \tilde{x} \end{bmatrix} \tag{16}$$

where:

- $\delta_i = x_i - x_0$ is the tracking error,
- $\tilde{x}_i = x_i - \hat{x}_i$ is the estimation error,
- $A_o = (I_N \otimes A) - c((L + G) \otimes FC)$,
- $A_c = I_N \otimes A - c(L + G) \otimes BK$

The closed-loop dynamics for the local observers' regulator become:

$$\begin{bmatrix} \dot{\delta} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A_c & B_c \\ 0 & I \otimes (A + cFC) \end{bmatrix} \begin{bmatrix} \delta \\ \tilde{x} \end{bmatrix} \tag{17}$$

For convergence of the state estimation error $\tilde{x}_i \to 0$, the following matrices must be Hurwitz:

- A_o in the distributed observer case
- $A + cFC$ in the local observer case

This ensures that the full closed-loop system is asymptotically stable, leading to successful tracking of the leader's trajectory by all agents.

4 Experimental Results

In this section, we present a numerical analysis of the two implemented regulators. Each regulator was tested under the different proposed network topologies. Based on the performance observed for the three types of steady-state references of the leader node (constant, ramp, and sinusoidal signals), we selected one network topology to carry forward for further analysis.

After selecting the network, we continued our study by analyzing:

- The impact of measurement noise on the system's output.
- The influence of the coupling gain c and the weighting matrices Q and R used in the design of the controller and observer.

4.1 Topology-Dependent Performance

To select the most effective network topology, we evaluated the convergence time to zero of each agent's estimation error as the performance metric. Specifically, we define convergence time as the first instant when the state estimation errors of all agents fall below a fixed threshold (e.g. 10^{-4}).

While this is a simplified criterion, a more refined analysis could involve verifying whether the error remains below the threshold for a given time window or continues to decrease over time. However, our approach allows us to obtain reasonably low convergence times with reduced computational complexity.

For fairness in the comparison, the initial states of the follower agents were randomly generated using a standard normal distribution with a fixed seed, ensuring consistency across different topologies.

Based on the convergence times observed under both regulators and across the three reference signals (constant, ramp, sinusoidal), we selected the topology that consistently showed the best performance.

The results obtained are shown in the Table 1. For each topology and reference signal, we report the State Estimation Error (SEE) and the convergence time, both for the cooperative and local observers. If the system does not converge within the simulation time, a NaN value is reported. This indicates that the system is still converging, but not within the desired threshold.

As expected, the results are strongly influenced by the type of topology. All topologies are able to achieve approximately the same SEE, but with different convergence times. While the distributed neighborhood-based observer (cooperative) shows good performance across almost all cases, the regulator based on local observers tends to perform better in scenarios with strong communication between nodes (such as the star and complete topologies), and worse in scenarios where nodes are sparsely connected in pairs (like in the chain and tree topologies).

This behavior is likely influenced by the absence of the ξ_i term in the local observer formulation. The ξ_i term, present in the cooperative observer, helps balance the new state estimate even when the communication graph is less connected.

As expected, the topologies yielding the best results are the star and complete configurations, which offer stronger communication among agents. However, each of these topologies presents specific advantages and disadvantages:

Topology	Reference	SEE Coop	SEE Local	Time Coop	Time Local
Chain	Const	1.99e-04	1.99e-04	70.75	71.00
	Ramp	1.99e-04	9.92e-05	10.75	151.40
	Sin	1.92e-04	1.87e-04	15.35	9.10
Star	Const	1.93e-04	1.99e-04	11.25	11.25
	Ramp	1.98e-04	1.98e-04	11.55	29.55
	Sin	1.98e-04	1.88e-04	12.45	8.10
Tree	Const	1.98e-04	1.98e-04	47.60	47.80
	Ramp	1.92e-04	4.55e-04	11.55	NaN
	Sin	1.99e-04	1.89e-04	12.25	8.10
Complete	Const	1.95e-04	1.91e-04	11.75	11.70
	Ramp	1.94e-04	1.99e-04	11.75	89.70
	Sin	1.98e-04	1.00e-04	12.05	11.05

Table 1: Cooperative and Local SEE and convergence times for each topology and reference type

- The star topology is more sensitive to failures since all follower nodes are connected only to a central node, which receive information from the leader. A failure in the central node can isolate all followers.
- The complete topology offers better robustness to node failures, as each node is connected to all others. However, it introduces a significantly higher communication cost due to the need for each agent to communicate with all others.
- Both topologies share a design limitation in that only one node communicates directly with the leader, regardless of the topology.

All analyses in this preliminary stage were conducted under the assumption of no measurement noise. In practice, noise can significantly affect performance, and its impact depends both on the topology and on the specific node(s) where the noise is introduced.

Based on preliminary analysis of noise effects and considering communication efficiency, we chose the star topology as the reference configuration for the rest of the study.

4.2 Noise Effect

After selecting the star topology as the network architecture, we investigated how measurement noise affects the performance of the two distributed regulators: one based on neighborhood (cooperative) observers, and the other based on local observers. To simulate realistic conditions, we added Gaussian measurement noise to all agents in the system, including the leader node. The noise was modeled with zero mean and different variance levels to simulate increasing noise severity:

- Low noise (variance $\approx$ 200)
- Moderate noise (variance $\approx$ 2500)
- High noise (variance $\approx$ 10000)

We tested all three noise levels across different types of reference signals (constant, ramp, and sinusoidal). The results showed that under low noise conditions, both regulators maintain good performance, with the state estimation error (SEE) converging effectively toward zero. This indicates that both approaches are robust when noise is minimal.

However, as the noise level increased, we observed a clear performance gap between the two regulators. Specifically, the local observer-based regulator exhibited significant degradation in estimation quality under moderate and high noise. In contrast, the neighborhood-based regulator remained more stable and accurate.

This behavior can be attributed to the difference in how each observer processes measurement information. The neighborhood observer uses measurements from neighboring agents, which allows it to average out or attenuate the effect of noise. On the other hand, the local observer relies solely on its own noisy output, making it much more vulnerable to measurement disturbances.

This effect can be seen when comparing the SEE plots (Figures 5 and 6) under moderate noise and a constant reference signal: the cooperative observer maintains a low and stable error, while the local observer shows larger fluctuations.

In conclusion, the neighborhood-based regulator demonstrates better noise robustness due to its ability to exploit distributed sensing, while the local observer struggles in high-noise conditions.

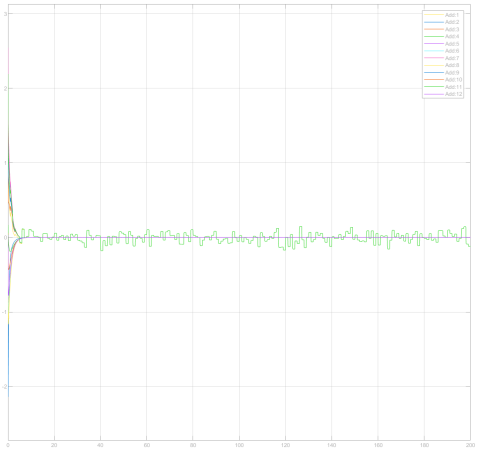

Figure 5: Cooperative Observers with constant reference

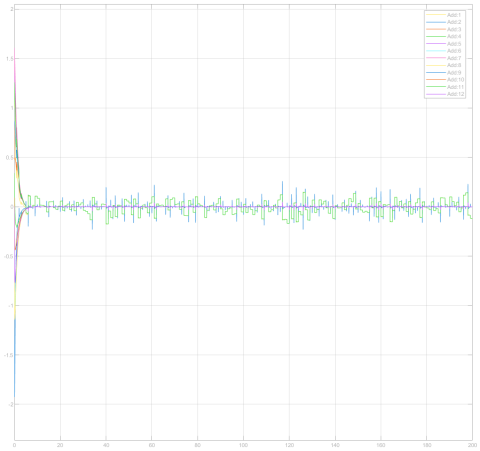

Figure 6: Local Observers with constant reference

4.3 Influence of Weighting Matrices Q and R

The observer gain matrix F is derived from the solution of the Algebraic Riccati Equation (ARE), and its performance heavily depends on the choice of weighting matrices Q and R. Moreover, these matrices are also involved in the optimal solution of the Linear Quadratic Regulator (LQR) problem:

$$u(t) = \arg\min_{u(t)} \left[\frac{1}{2} \int_0^\infty \left(x^T(t)Qx(t) + u^T(t)Ru(t) \right) dt \right]$$

For these reasons, it is primarily the ratio between Q and R that influences the system's behavior, rather than their absolute magnitudes.

State Weighting Matrix (Q)

- **Role**: Penalizes state deviation in the control objective and estimation error in the observer dynamics.
- **Design Considerations**:
  - Larger entries in Q lead to faster convergence of the estimation error and more aggressive control responses.
  - Excessively large values may amplify measurement noise and reduce robustness.

Control Weighting Matrix (R)

- **Role**: Penalizes control effort and implicitly models measurement noise in the observer design.

- **Design Considerations**:
  - Larger values of R produce smoother estimates and improved noise rejection.
  - However, too conservative choices may lead to slow system response and reduced tracking performance.

The fundamental trade-off between estimation accuracy (weighted by Q) and control smoothness or noise sensitivity (weighted by R) is a key consideration in both observer and controller synthesis. A balanced choice is crucial to achieve desirable performance.

To empirically analyze the effect of different choices of Q and R, three configurations are tested on a star topology network with zero measurement noise:

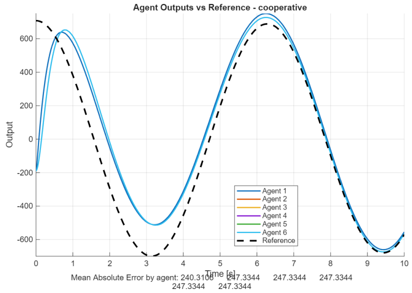

Figure 7: Emphasizing R

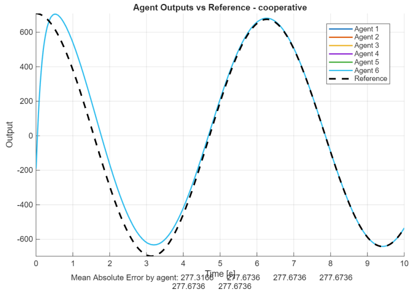

Figure 8: Emphasizing Q

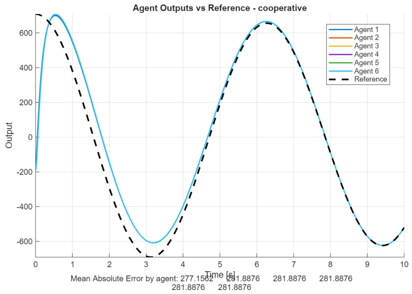

Figure 9: Equally weighted

As observed in Figures 7 and 8, emphasizing Q over R yields faster and more aggressive tracking of the reference. In contrast, emphasizing R results in a smoother but slower response. The best trade-off between convergence speed and estimation robustness is achieved when Q and R are balanced, as illustrated in Figure 9.

4.4 Effects of the Coupling Gain on System Behavior

The coupling gain c plays a pivotal role in shaping both the stability and performance of the multi-agent system. The influence of the coupling gain manifests in two main aspects:

1. **Stability and Convergence**: The gain c directly scales the eigenvalues of the matrix $L+G$, where L is the Laplacian matrix encoding the inter-agent communication topology, and G represents the leader-follower coupling. This scaling affects how quickly the agents synchronize and track the leader.
2. **Trade-off Between Robustness and Performance**: Increasing c generally improves the rate of convergence and robustness to variations in the communication topology. However, excessively large values of c may result in high-gain control, which can: amplify measurement noise sensitivity and lead to larger control inputs, imposing stress on actuators

or exceeding practical limits, as shown in Fig.11. Conversely, a very small c may result in: weak coupling, preventing effective synchronization and poor disturbance rejection and slower convergence.

Hence, the choice of c involves a fundamental trade-off. It must be carefully tuned to achieve an optimal balance between speed of convergence, noise robustness, and practical implementation constraints.

The following figures illustrate the behavior of the system under different values of the coupling gain c:

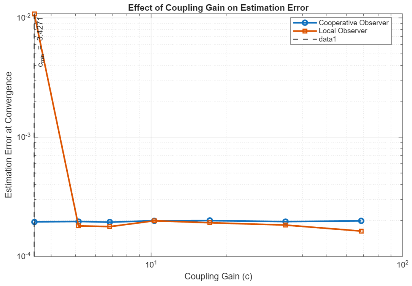

Figure 10: Coupling gain vs estimation error

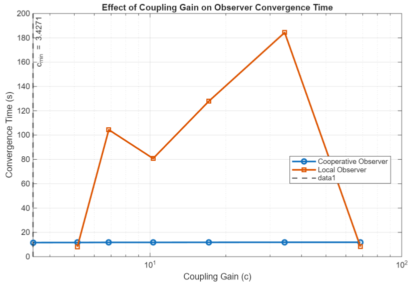

Figure 11: Coupling gain vs convergence time

4.5 Cooperative Observer's Insensitivity to Coupling Gain

The cooperative observer demonstrates remarkable robustness to changes in the coupling gain, as evidenced by the plateau in both convergence time and estimation error metrics. This behavior can be attributed to:

- **Network-Averaging Effect**: The cooperative observer's architecture leverages information exchange across neighboring agents. This distributed consensus mechanism inherently mitigates the impact of individual coupling gains through network-wide error averaging.
- **Eigenvalue Dominance**: Furthermore, the observer's stability is governed by the eigenvalues of $L + G$. The network's algebraic connectivity (minimum $\mathrm{Re}(\lambda_i)$) establishes a stability threshold for c, beyond which additional gain provides diminishing returns.

5 Conclusion

This study examined distributed control strategies for a multi-agent magnetic levitation system, comparing cooperative and local observer-based approaches. The experimental results demonstrate that the cooperative observer architecture, which utilizes information exchange between neighboring agents, provides superior performance in terms of:

- Robustness to measurement noise and network disturbances
- Consistent convergence across different network topologies
- Better handling of sparse communication configurations

In contrast, the local observer approach, while computationally simpler, showed significant performance degradation in scenarios with:

- High noise environments
- Limited inter-agent communication
- Demanding reference trajectories

The investigation of network topologies revealed that star and complete configurations offer the most favorable trade-offs between convergence speed and system robustness. The analysis of control parameters highlighted the critical role of:

- The coupling gain in balancing responsiveness and stability
- The weighting matrices in managing control effort and estimation accuracy

These findings suggest that the cooperative observer represents a more reliable solution for practical implementations where noise resilience and network flexibility are essential. The insights gained from this study contribute to the broader understanding of distributed control in cyber-physical systems, particularly for applications requiring precise coordination under real-world constraints.