Politecnico di Torino

# Modeling and Control of Cyber-Physical Systems

Project II

Sofia Longo 310183
Matteo Gravagnone 319634
Gabriele Martina 310789
Danilo Guglielmi 318083

# Distributed control of a multi-agent magnetic levitation system

The entry point of our implementation of the distributed control protocol for a multi-agent magnetic levitation System is the Matlab file ‘`project.m`’. Here in the ‘Observer type’ section we define some flags that allow us to select the type of simulation we want to perform:

- **no_observer**: if true there’s no observer implemented, if false we use the simulation with observers
- **coop**: decides whether the observer is cooperative
- **local**: always the contrary of coop, indicates that the observer is local.

## Topology

First, we discuss the pros and cons about different network topologies. Our attention goes towards three different options: **chain**, **star**, or **mixed topology** which are evaluated considering a sinusoidal reference signal dictated from the leader.

If we select the **chain topology**, we notice that there’s a delay proportional to the distance of the followers from the leader and the weight of the links influences the propagation speed through the network.

On the other hand, in the **star topology** the behaviour is different, as the information does not travel anymore from a follower node to the next one, meaning that there is no cascade propagation through follower nodes. Instead, since every node is only linked to the leader, the convergence speed depends only on the inverse of the pinning gain.

The most convenient compromise is a **mixed topology**: only some nodes are directly linked to the leader while others are sharing information between each other. In this way we record some improvements, such as:

- The leader has less overhead. It does not have to send the information to everyone but only to a subset of nodes, which then propagate it to their neighbours
- Robustness. If a link fails, we have some chance of remaining within reach and avoid isolation thanks to links that may come from other functioning neighbours.

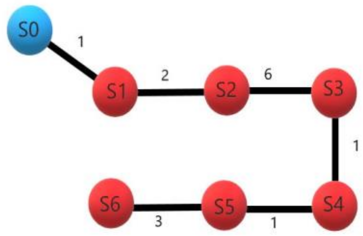



*Figure 1: Chain topology*

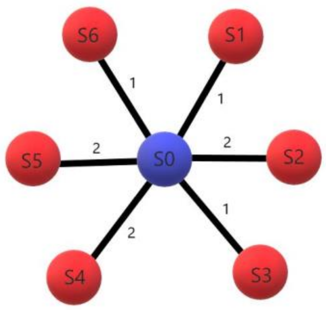



*Figure 2: Star topology*

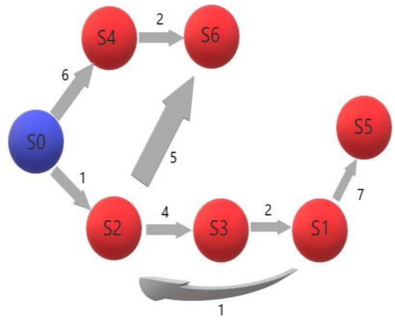



*Figure 3: Mixed topology*

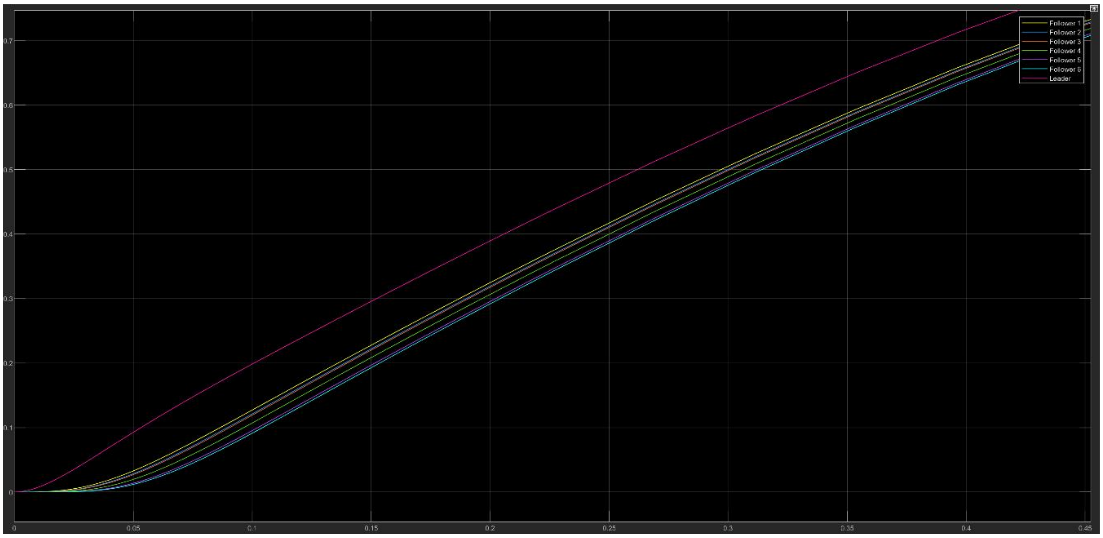



*Figure 4: Zoom on Chain Topology agents' behaviour*

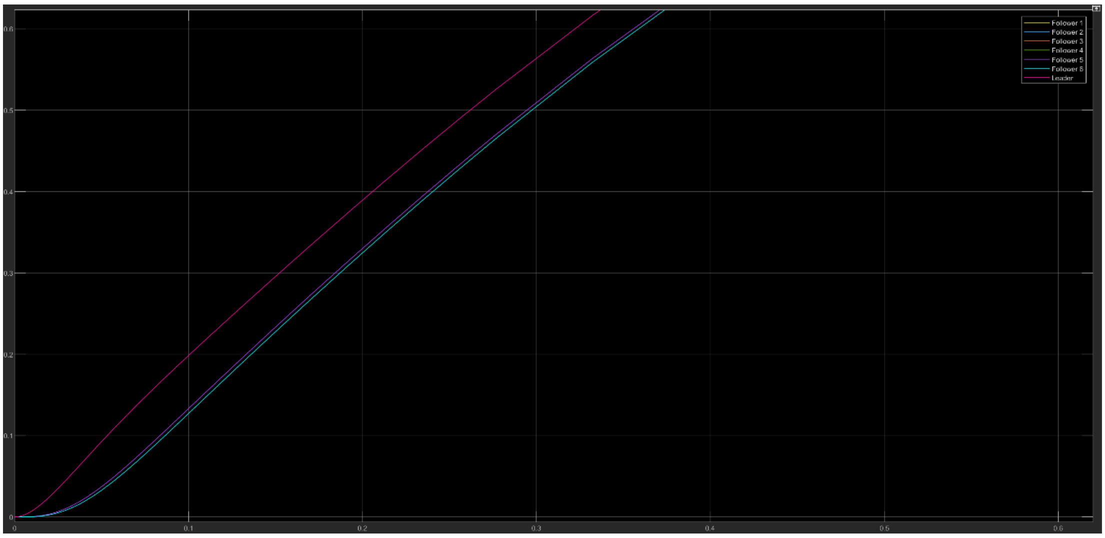



*Figure 5: Zoom on Star Topology agents' behaviour*

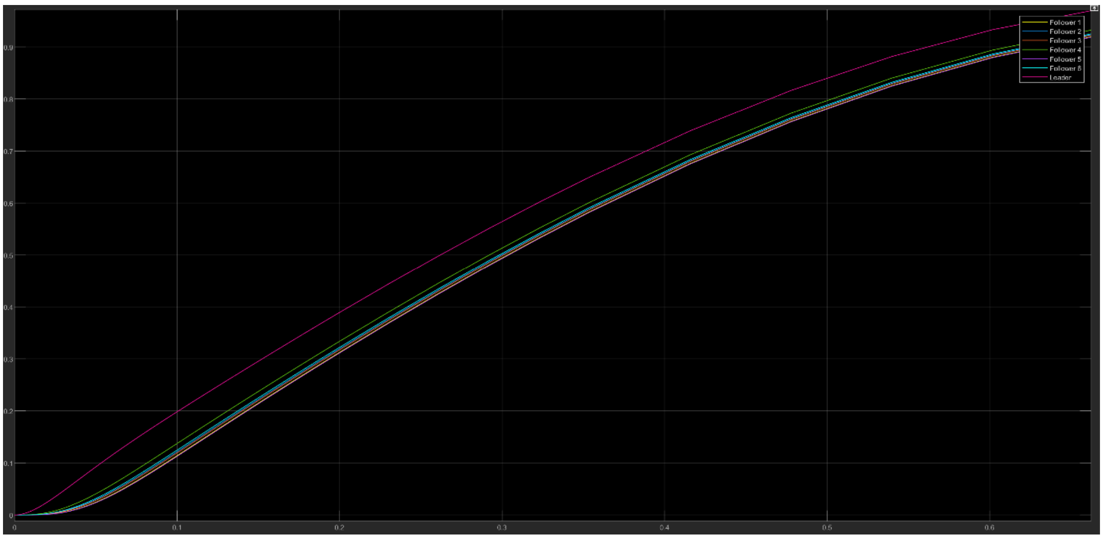



*Figure 6: Zoom on Mixed Topology agents' behaviour*

## Effect of noise affecting output measurements.

To test the effects of noise on output measurements in our problem, we simulated the system with the presence of a normally distributed noise on the output of the follower node #1, both with cooperative and local observers.

The following plots of the error between output and desired output show how the designed controller is still able to bring it to zero, however the choice of the observer type influences the transient.

In particular, the nature of the noise always affects the way the "poisoned" follower agent chases down the leader's behaviour, as the output is not a smooth curve but a saw-toothed one.

The noteworthy observation is, when choosing the local observer, this happens for only that single agent, whereas the noise propagates to other agents when computing their inputs accordingly to the SVFB control algorithm, except for a single agent in our topology: the only node (follower #4) which is linked only to the leader.

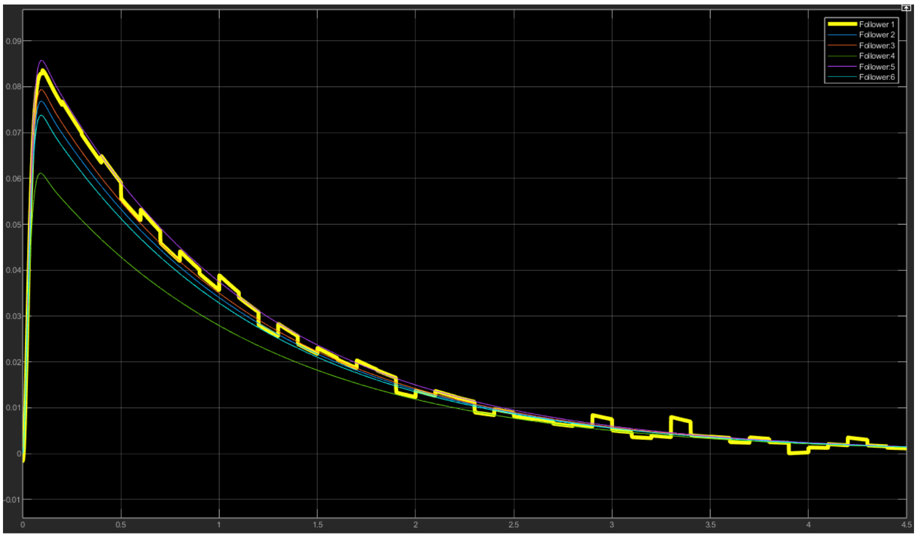



*Figure 7: Noise on Follower #1 with local observer*

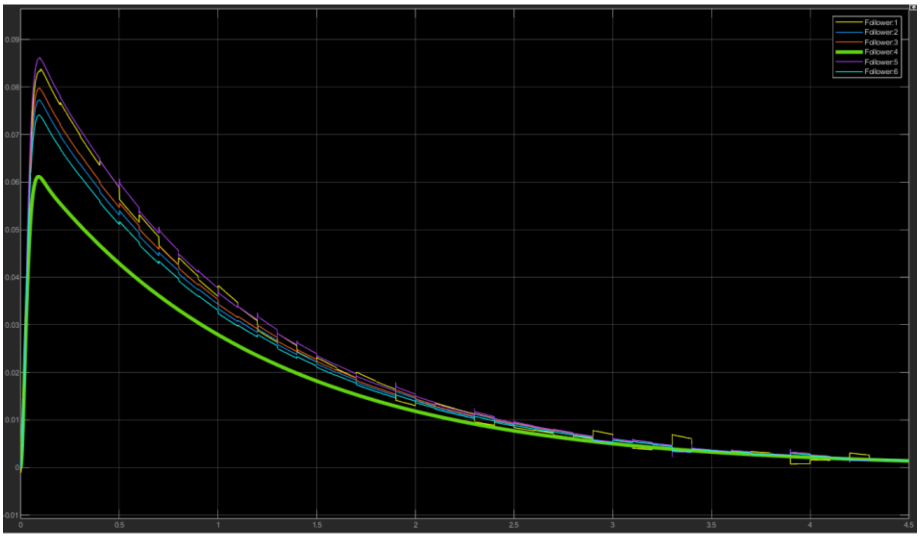



*Figure 8: Noise on Follower #1 with cooperative observer*

## Choice of c and Q,R.

- **Varying c keeping Q, R constant**

The experiments performed show that the choice of the coupling gain *c* affects the behaviour of the followers tracking error. When setting *c* as the minimum required by Theorem 1 the tracking error reaches noticeably different peaks among the followers, typically up to a maximum[1].

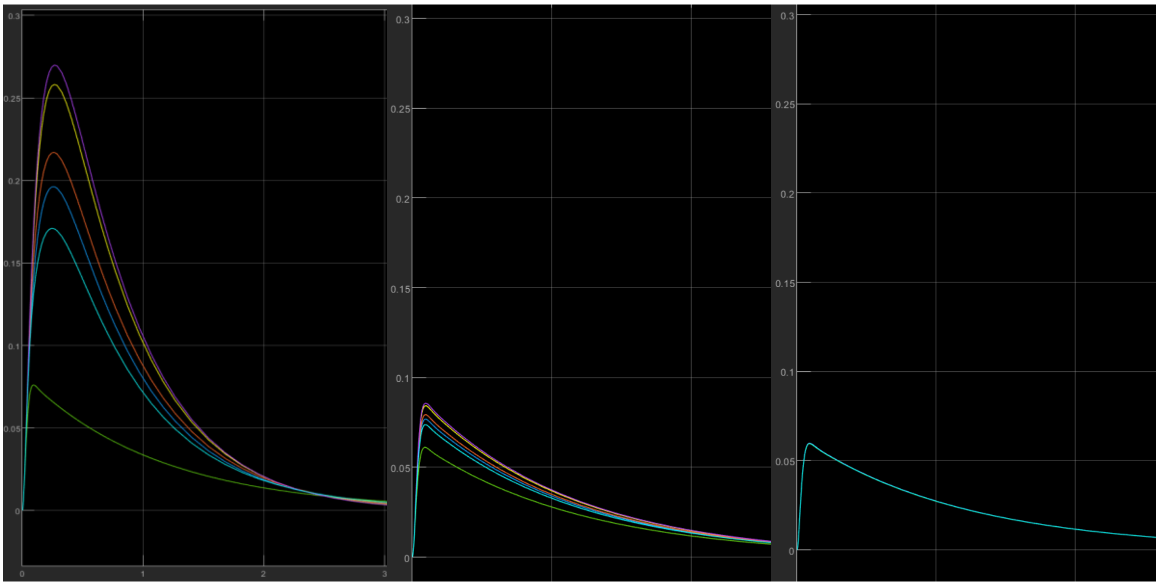
When increasing c first by a 10x factor and second by a 100x factor the peaks gradually decrease in magnitude up to a certain value[1], leaving the convergence pace of the algorithm equal.

When increasing c by 1000x there is no significant variation in the magnitude of peaks. However, we notice that the error function behaviours of the various agents overlap and there is much less differentiation between them.

[1] *different for each type of input signal*

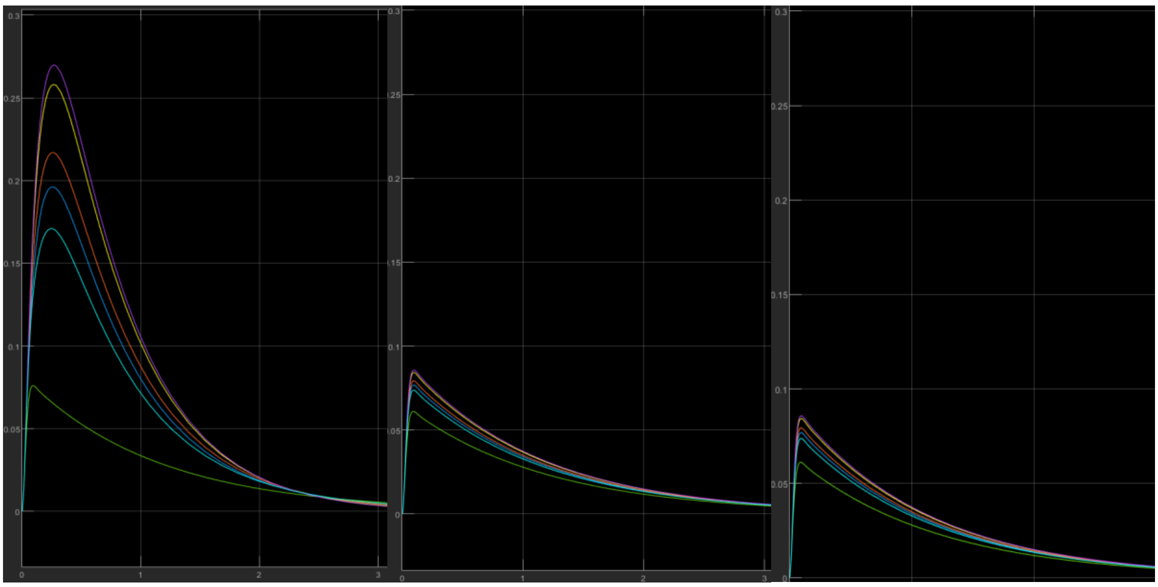
This pattern is witnessed across both types of observers and even when altering the shape of the input (sinusoidal, constant, ramp).

- *Local observer*



c = 0.5997 x10 x1000

- *Cooperative observer*

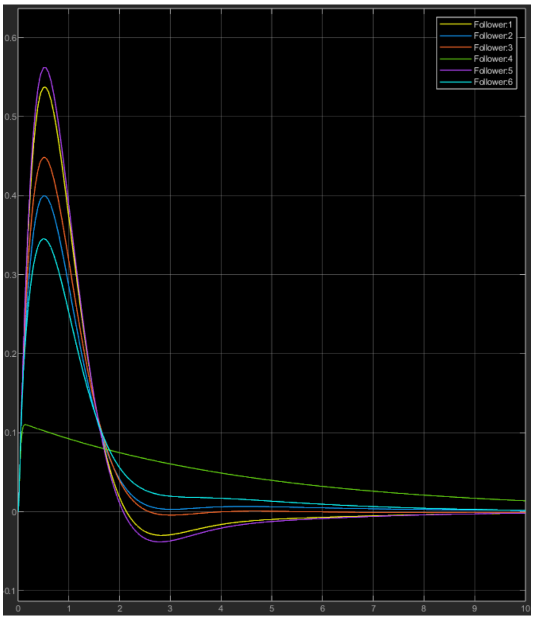
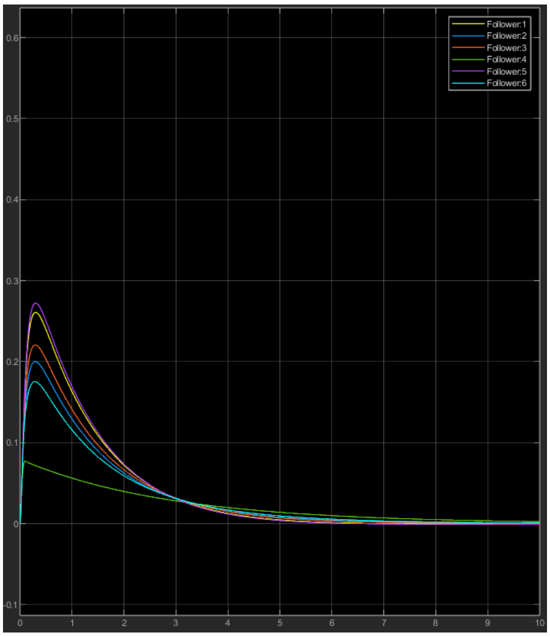


c = 0.5997 x10 x1000

- **Varying Q, R keeping c constant**

A realistic design of the control algorithm must account for the balance between high performance and energy consumption of the command activity.

We can manage this by manual tuning of the weight matrices Q and R as these properties depend on the relationship between the two. Values of Q greater than R lead to better performance (faster convergence) at the expense of a greater energy consumption. Lower consumption can be ensured by Q and R having comparable magnitudes or when R surpasses Q.

<table>
<tr>
<td>R = q*100<br>We are demanding a “very” low energy consumption, unable to bring error to zero in the first 10s with the imposed restriction on energy<br>
</td>
<td>R = q*10<br>Here we request low energy consumption at the cost of longer convergence time<br>
</td>
</tr>
</table>

Values comparable to those in matrix Q or smaller lead to similar results, like the ones following, without meaningful improvements.

This shows a saturation in the effect of these matrices on the convergence time.