Modeling and Control of Cyberphysical Systems
Project 2

Daniele Bigagli   Federico Maresca   Salvatore Sipione

July 2021

1 Introduction

In this project, we study a Cyber-Physical System made up of 7 magnetic levitation systems (Maglev). We design a distributed control protocol where the multi-agents system perform the cooperative tracking problems; considering different network communication structures, we compare the effect of these different choices on the controlled system. Then, we select a particular network structure that allows us to highlight the convergence time instants between scenarios where the leader steady-state agent reference behaviour changes. Moreover, we analyze the effect of the introduction of a noise to the output measurements and the effect of different variables used to design the controller and the observer.

1.1 CPS Model

The CPS under study is modeled as a multi-agents system where:

- one Maglev system acts as a leader node S_0
- the other Maglev systems act as follower nodes S_i $(i = 1,..,N)$
- state variables are not directly accessible for measurements
- each single agent is described by the following state-space equations obtained by linearizing the physical equation around a suitable equilibrium point

$$\dot{x_i} = Ax_i + Bu_i, \quad y_i = Cx_i \tag{1}$$

where:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 880.87 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ -9.9453 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 708.27 & 0 \end{bmatrix} \tag{2}$$

2 Communication Network

We tried different network structures in our experiments, from the most trivial star-connected agents, in which every follower node has a link to the leader, to more complex ones. We will go deeply in discussing each case.

2.1 Star Configuration

This is the most simple configuration. In this case the convergence times are the same for all the agents, because they receive the same information at the same time so they don't need to wait for other agents. This is the network with the best performance as expected, but we decided to not choose it for the next section of experiments, due to its simplicity and the fact that every follower has the same dynamics, making the first follower indistinguishable from the other ones.

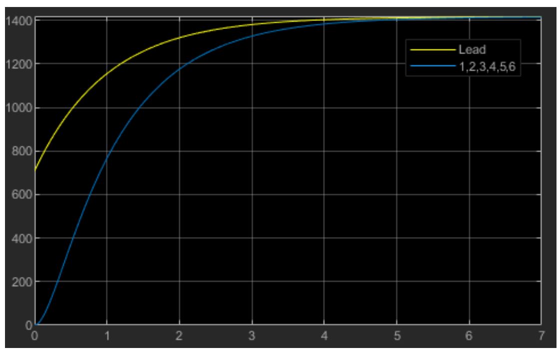

Figure 1: Plot of the outputs of all the agents in the Star Configuration

2.2 Fork Configuration

In the forked version, we created two branches of followers nodes, with the head of each branch linked to the leader node. Each branch was composed of 3 followers (noted as Even or Odd branch), where every node behaviour was identical to the one parallel to it in the other branch (1 - 2, 3 - 4, 5 - 6). We tried modifying this structure by adding edges such as 6 to 1 and subsequently changing its weights; these two changes can be seen respectively in figure (3). The former shows how adding that edge shift the Odd branch towards the last agent of the Even branch, while the latter increases that shift even more in the dynamics.

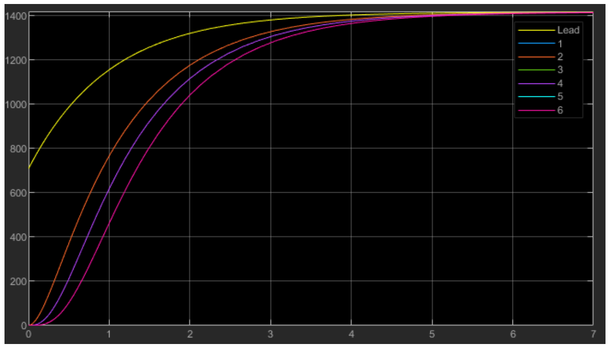

Figure 2: Plot of the outputs of all the agents in the Fork Configuration

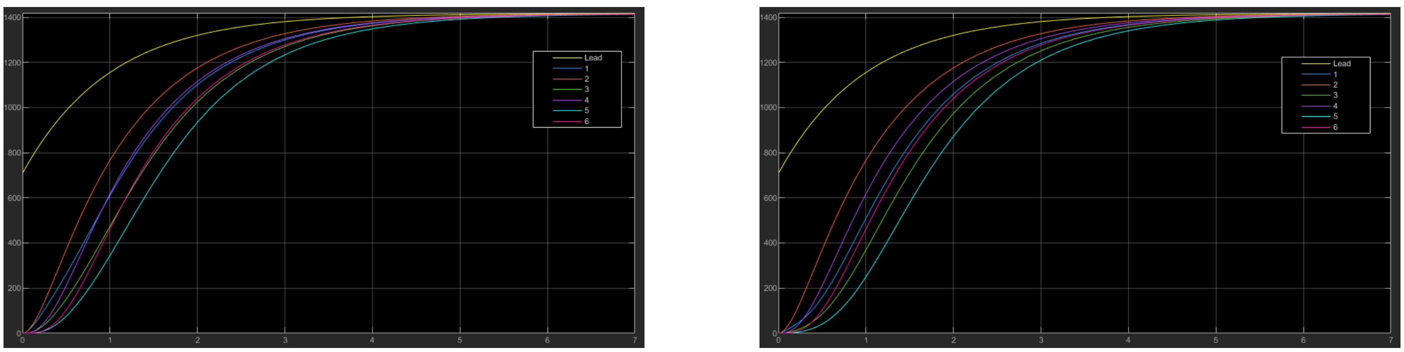

Figure 3: Plot with the 6-1 edge (left) and with augmented weights (right)

Again, the nodes' behaviour was identical, taken two by two, so we chose another structure.

2.3 Queue Configuration

In the queued version, as the name says, each follower node is connected to the next one starting from the leader node. We will refer to this as our standard queue. We tested some changes, such as an edge from 6 to 1, one from 6 to 2, with original and modified weights; the following figures show the dynamics of these different configurations. Much like in the fork structure, the 6 to 1 edge shifts all agents towards the last one, while the 6 to 2 edge leaves the first agent's behaviour untouched. In the figure standard the convergence time is quite lower with respect to both the other structures; this is due the fact that the coupling gain was much higher than in the other two configurations.

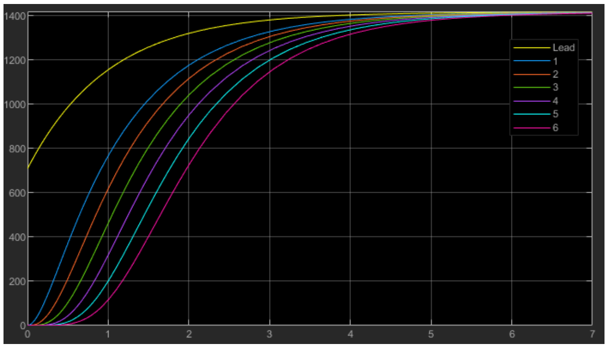

Figure 4: Plot of the outputs of all the agents in the Queue Configuration

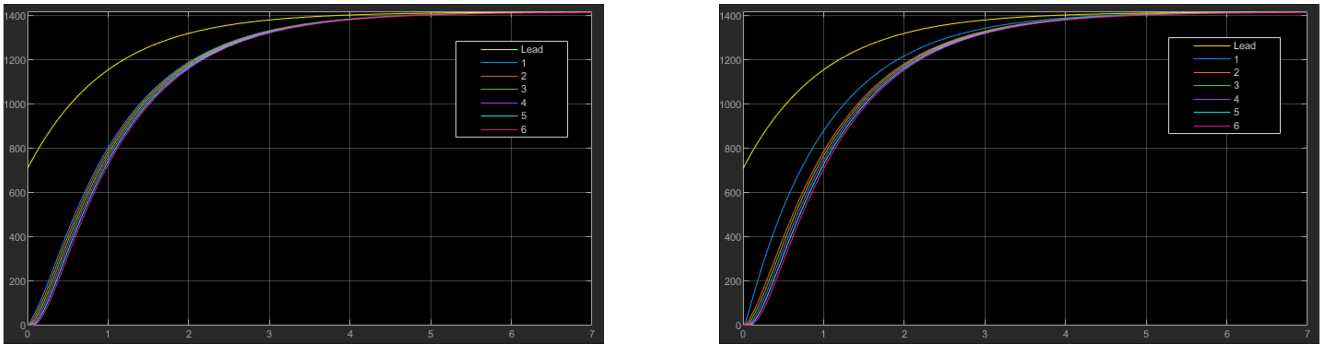

Figure 5: Plot with the 6-1 edge (left) and with 6-2 edge (right)

The more agents were included in the loop, the higher c was due to changes in the eigenvalues of the Laplacian matrix $\mathcal{L}$.

$$\mathcal{L} = Deg - Adj \tag{3}$$

Having chosen this structure then our network matrixes were defined as:

$$\mathbf{Adj} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{Deg} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{4}$$

3 Observers

3.1 Local

The local observer uses only local agent information, in particular its estimated state and the measured output from the agent.

(i)

$$\bigcup_{i=1}^{N}(A + cFC) \text{ is Hurwitz}$$

(ii)

$$\tilde{y_i} = y_i - \hat{y}_i, \forall i$$

(iii)

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

Each node's observer measures the output and computes the error from it's own output estimation. The observer gain F such that the real part of the eigenvalues of equation 1 is strictly less than zero.

3.2 Neighborhood

The neighbourhood observer technique uses the measured output from the agents neighbours to estimate the states.

$$F = PC^T R^{-1}$$

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

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

Following these equations we created a Simulink model. For the F observer gain, an Algebraic Riccati Equation had to be solved and so we had to choose values for $\mathcal{Q}$ and $\mathcal{R}$. These would influence the convergence time of our model, in fact the larger $\mathcal{R}$ is with respect to $\mathcal{Q}$ the slower the observer will be to converge. On the contrary the smaller $\mathcal{R}$ is than $\mathcal{Q}$ the faster the observer will converge. In both the local and neighbourhood approach the coupling gain plays an important role since directly influences the global state error dynamics asymptotic stability. Actually, as long as F follows the above equation and:

$$c \geq \frac{1}{2 \min_{i \in N} \mathcal{R}\mathrm{e}(\lambda_i)} \tag{5}$$

the asymptotic stability is maintained. All simulations were run for 200 seconds with a fixed coupling gain kept at the minimum value. In the figure below we plot the convergence time of our system with respect to $\mathcal{Q}$ and $\mathcal{R}$.

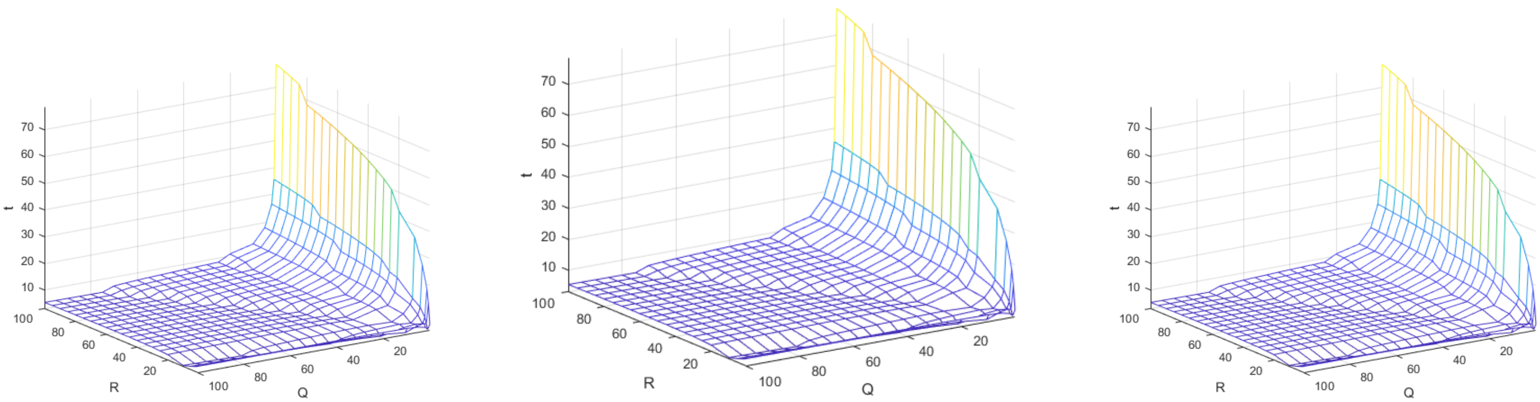

Figure 6: Convergence time with variations of $\mathcal{Q}$ and $\mathcal{R}$ with constant leader (left), ramp leader (center) and sinusoidal leader (right)

Below we show the convergence time with respect to the coupling gain, for this plot we kept Q and R to fixed values chosen from our previous test.

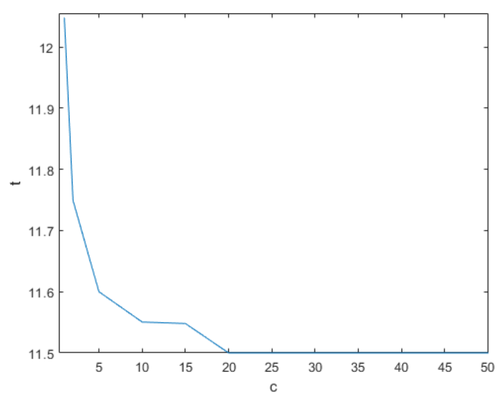

Figure 7: Convergence time variation with respect to the coupling gain

4 Steady state reference

In the end we chose $\mathcal{Q}$ as a diagonal matrix with a value of 50 in the diagonal and $\mathcal{R} = 10$. We tested both the observers with 3 different steady state reference behaviour; a constant of amplitude 1, a ramp of slope 1, and a sinusoidal signal of amplitude 1. Each observer reached asymptotic stability with every reference. We've run different experiments for the local observer and the neighborhood observer, with randomized starting states for the six follower agents. Both the local and neighborhood one performed similarly with negligible difference between them. In fact in absence of noise the local observer and the neighborhood one behaved identically with no discernible difference in any of the 3 test cases. In presence of noise they still performed similarly with the neighborhood being less influenced by the noise. We modeled the noise as a white gaussian noise with mean 0 and variance 0.001. In plot Figure 7 we show the convergence time with different values of c, our coupling gain.

4.1 Experiments

In the various experiments we set the initial state of the leader according to the values that let it produces the constant, the ramp and the sinusoidal signal. The follower agents states x_0 and x_1 are initialized with random values bounded in the interval $[-1, 3]$.

4.1.1 Constant

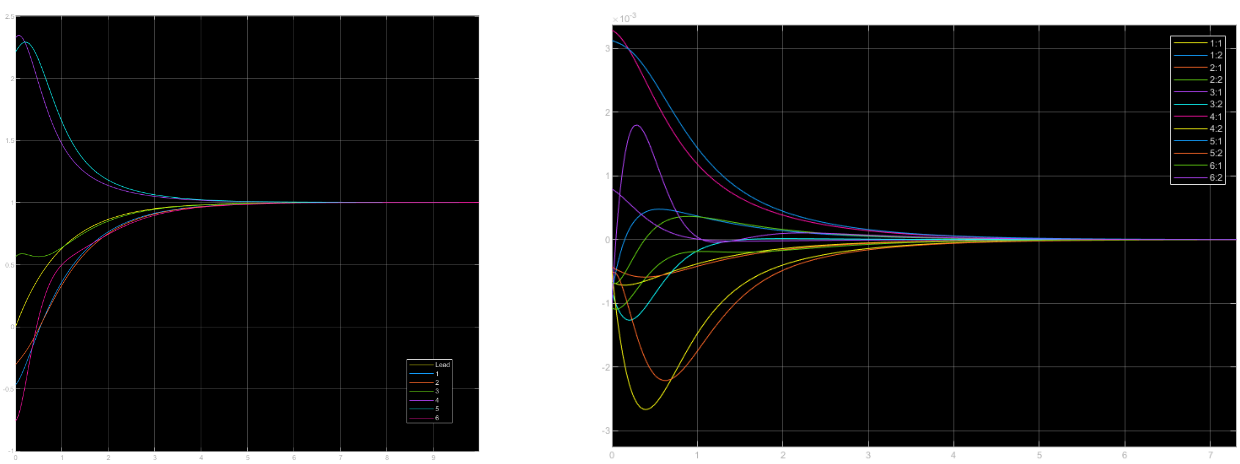

Figure 8: Plot of the outputs (left) and the states (right) with constant leader behaviour

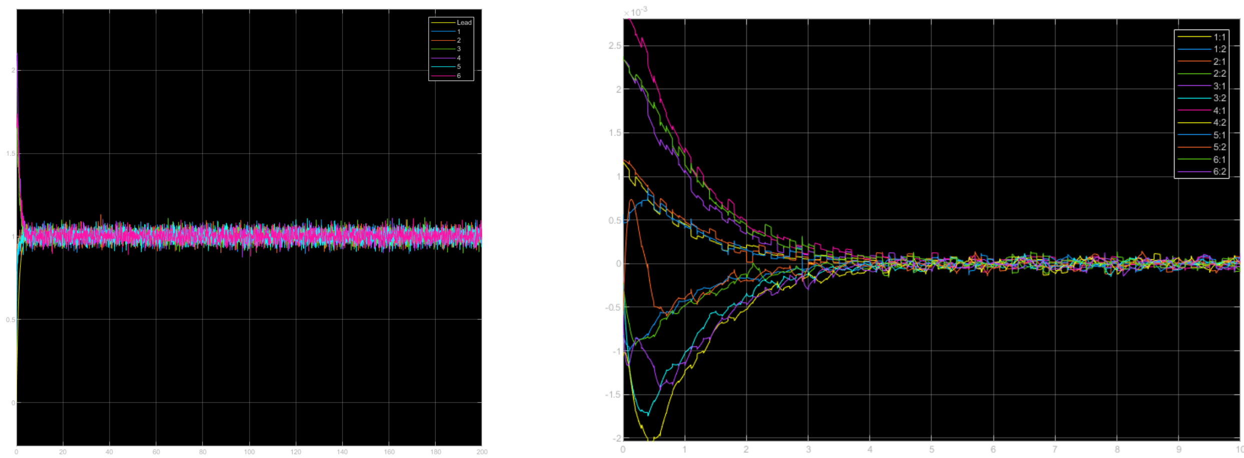

Figure 9: Plot of the outputs (left) and the states (right) with constant leader behaviour and measurement noise

4.1.2 Ramp

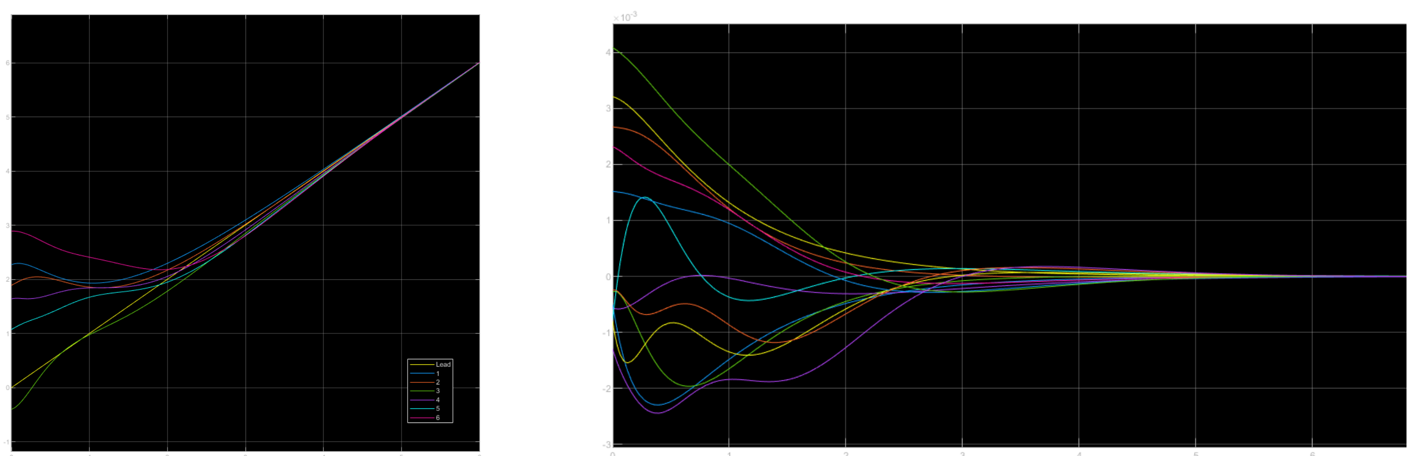

Figure 10: Plot of the outputs (left) and the states (right) with ramp leader behaviour

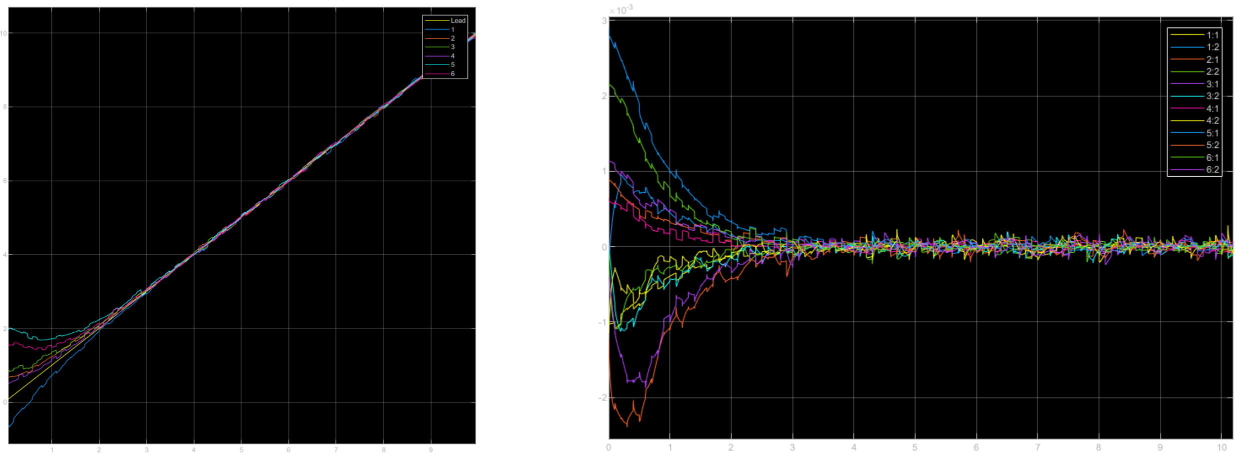

Figure 11: Plot of the outputs (left) and the states (right) with ramp leader behaviour and measurement noise

4.1.3 Sinusoidal

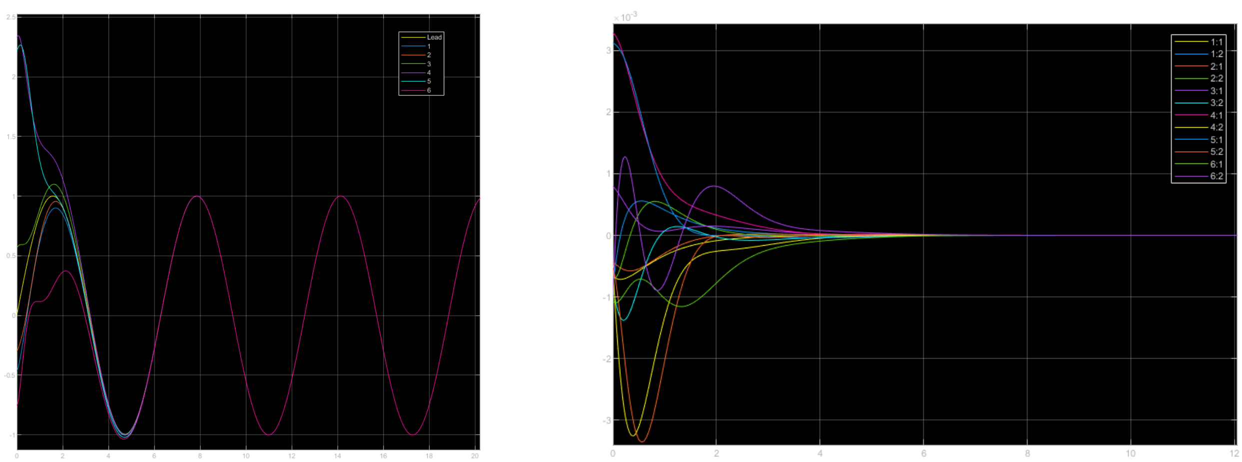

Figure 12: Plot of the outputs (left) and the states (right) with ramp leader behaviour

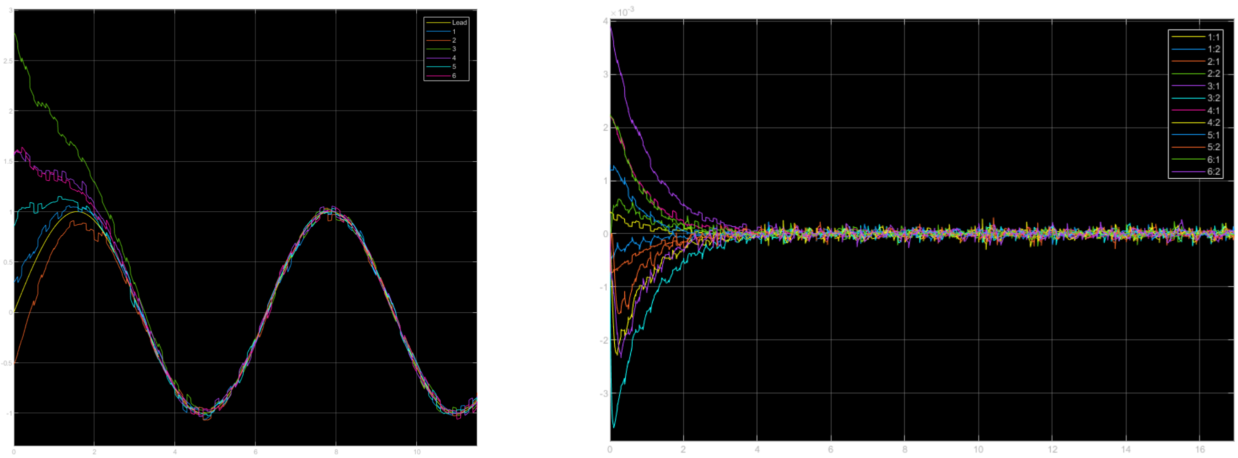

Figure 13: Plot of the outputs (left) and the states (right) with ramp leader behaviour with measurement noise