Modeling and control of cyber-physical systems
Project II

Botta Giulia (s310478) – Di Fazio Angela (s312641) – Maggipinto Luigi (s319874)

1. Introduction

In this project, in order to create a distributed control protocol, we considered 4 topologies characterized by different links between nodes and relative weight distributions. In each topology there are 1 leader node and 6 agent nodes, that share information, among which we chose the second and the fifth ones to be affected by noise.

We designed a general scheme on Simulink such that we didn’t have to change the structure each time but only modify the matrices according to the topology.

As regards the local observers, while the leader does not receive any input, the agents receive the tracking error epsilon, and each node sends its estimate to a concatenate block, that aims to suitably organize the data for the subsequent function. After that, we used a MATLAB function in order to compute the estimation error, and a second one to convert the data in a convenient form for the demultiplexer.

The global observer uses the same structure of the local observer to compute the tracking error and the neighbourhood estimation error; in addition, each node has a local observer because the state variables are not directly measurable.

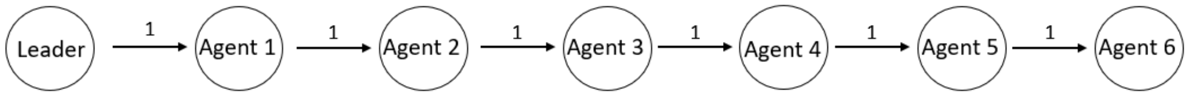
For the noise cases we used the same structure explained before with the addition of the random blocks on the output measurement of the second and fifth node.

As regards the coupling gain c and the weighting matrix Q and R, we started with these values: $c \propto 10$; Q=1*eye(2); R=1/10; and we modified them to see their impact in the behaviours of the signals.

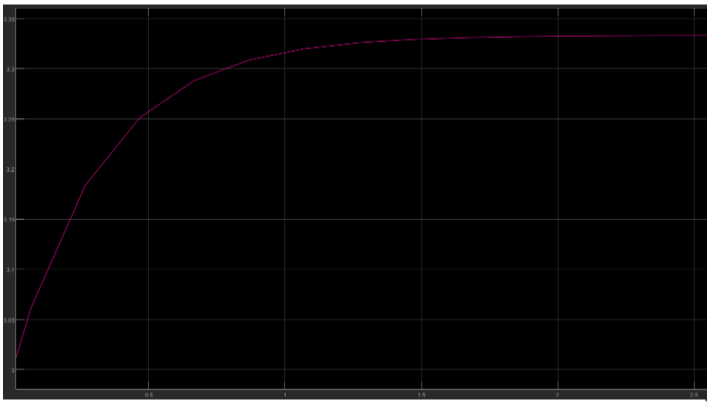
In the global case we computed the F matrix used for the tracking error solving the ARE equation, while in the local case we had to design the matrix F (F1) choosing the eigenvalues in order to obtain a stable matrix; otherwise, the system was unstable. The frequency at which happens the cut-off of the noise is determined by the eigenvalues chosen, the bigger they are, the higher is the cut-off frequency. For this reason, we chose [-1, -2] as eigenvalues so that the cut-off frequency was “small” and it could cut more noise.

2. Topologies

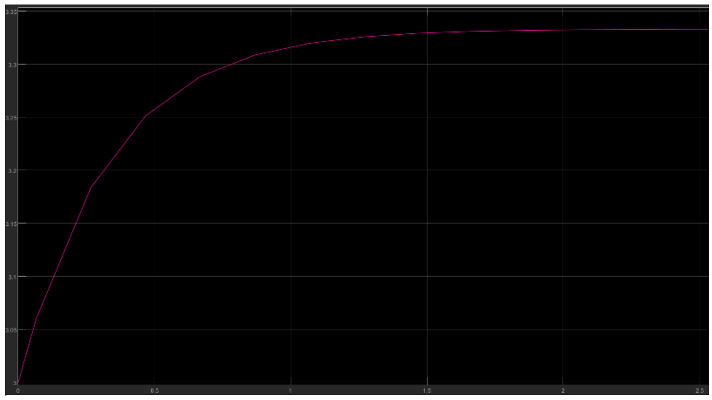
2.1. Topology 1



- Tracking to constant value, we used [0, -3] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a constant value. We tried with different values (for example [0, -5] or [0, -10]) and we saw that nothing changed except for the convergence time due to the different values of the eigenvalues.



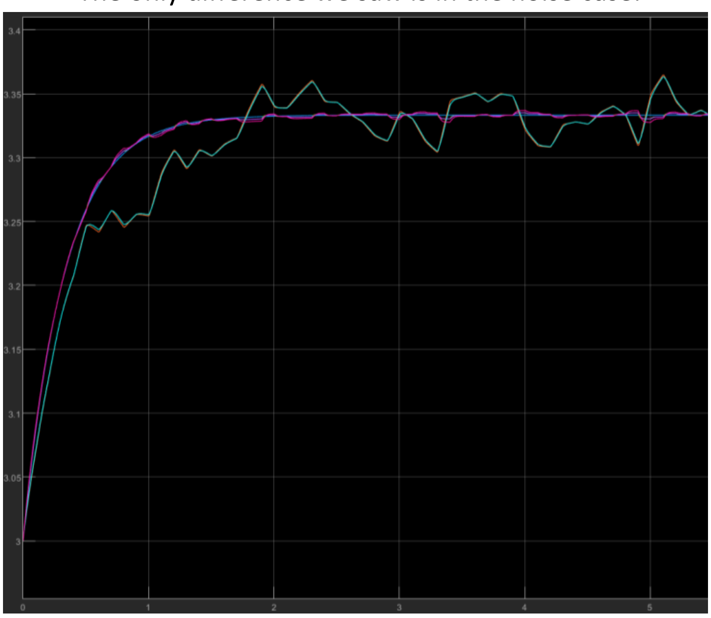
Global



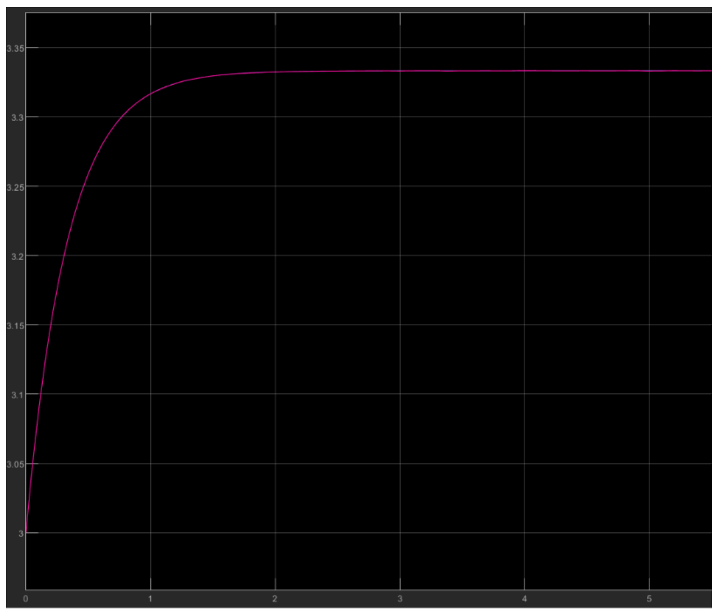
Local

It is difficult to notice a difference in the convergence time of the two graphs, this means that the performance of the global case and the local case are almost the same.

The only difference we saw is in the noise case:



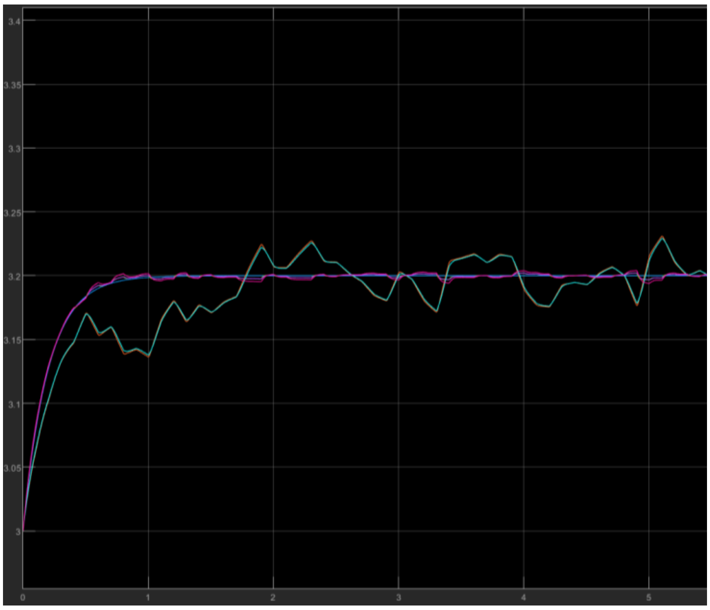
Global

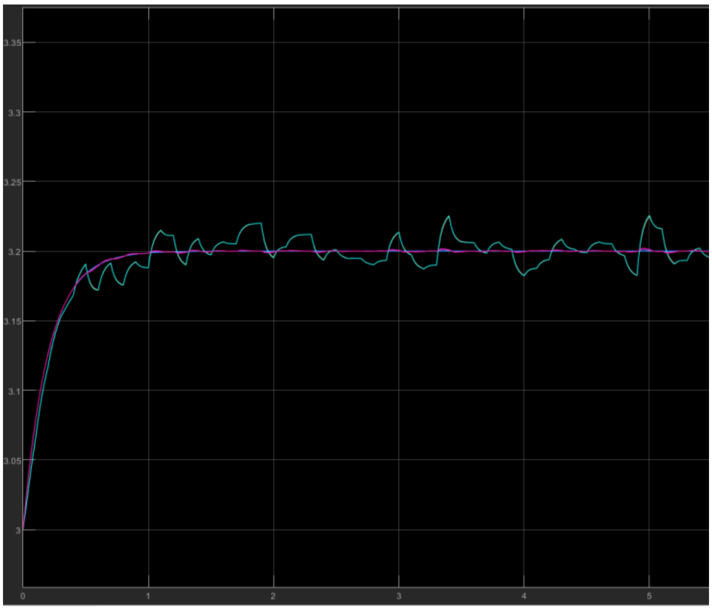


Local

The global control protocol shows the noise in the output while in the local case no noise is propagated. Taking a more precise look at the convergence time it can be seen that the global is a little bit faster than the local.

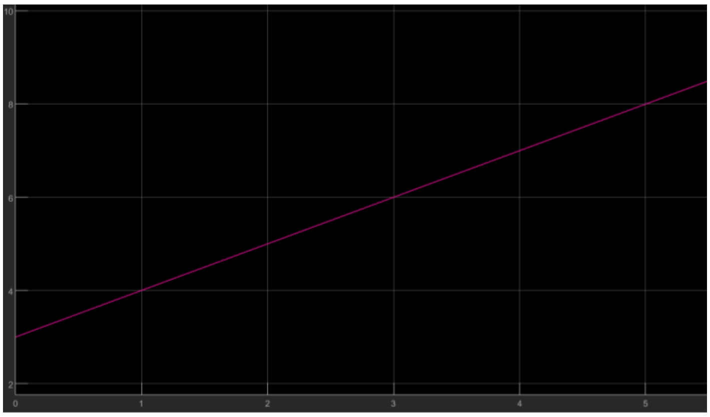
Using as eigenvalues [0, -5] and higher values, it can be seen that also the local estimator shows some noise in the plot:

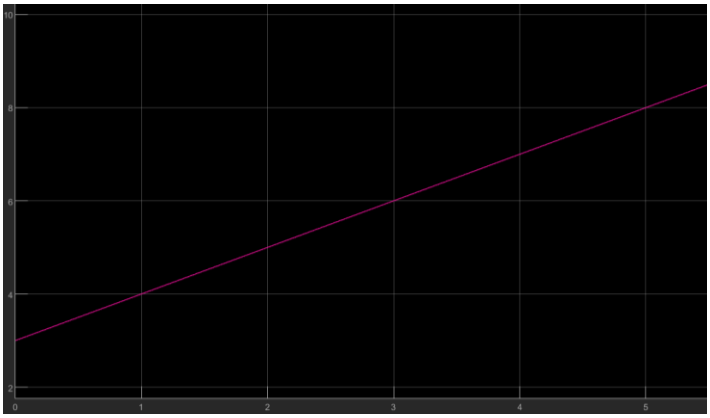

Global


Local

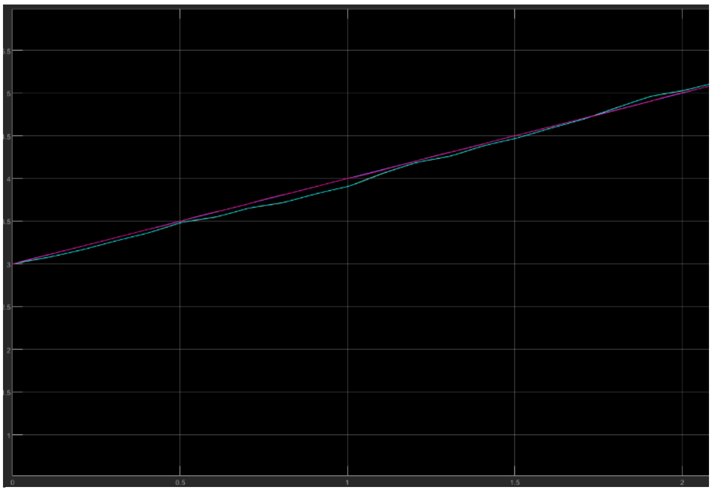
In this case we can see the noise in both graphs and we can also say that the local noise is smaller than the global noise (this is the reason why it wasn't shown in the [0, -3] case). The convergence time seems to be almost the same.

- Tracking to a ramp, we used [0, 0] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a ramp.

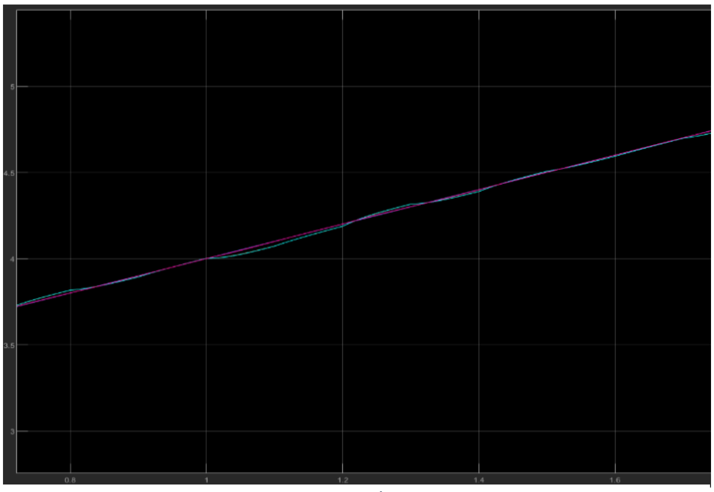

Global


Local

Without the noise, the two graphs are the same.



Global



Local

As noticed with the constant tracking, the local control protocol has less noise than the global one.

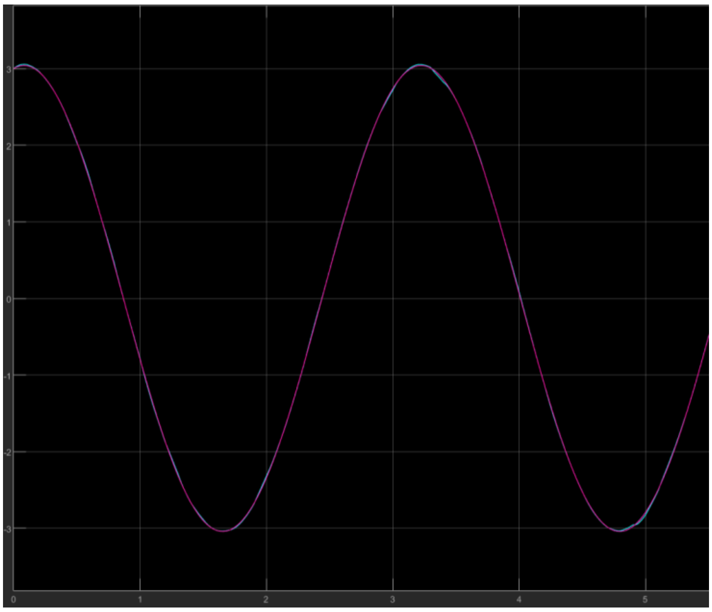
- Tracking to a sinusoidal signal, we used [-2j, 2j] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a sinusoidal signal. We tried also with other values (for example [-10j, 10j]) and we didn't notice any difference except for the frequency of the signal that is due to the magnitude of the eigenvalues.

Global

Local

As said before, without noise there's no difference in the two control protocols.

Global



Local

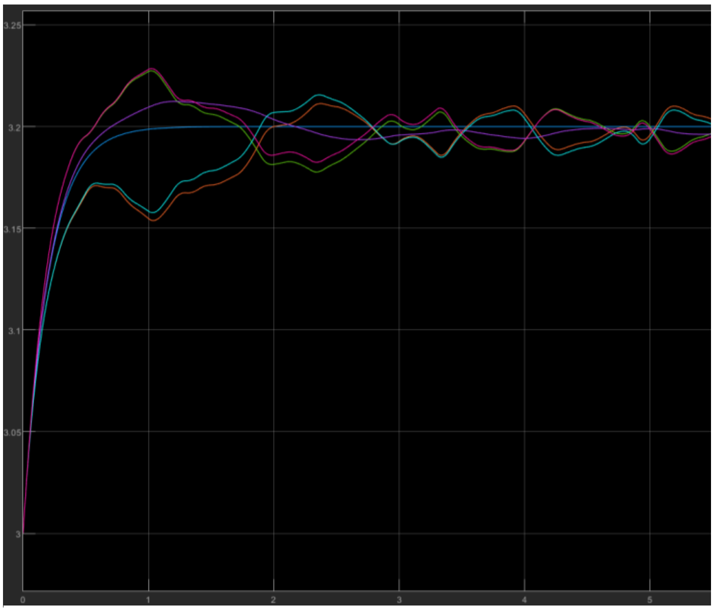
These graphs show the noise case, in the same way as in the previous cases: the local estimator is more precise than the global one and shows less noise.

In general, we understood that the local estimations are less corrupted by noise than the global ones because the noise in the local case remains local while in the global case all the nodes contribute to the global estimate so the noise ends up in each estimate.

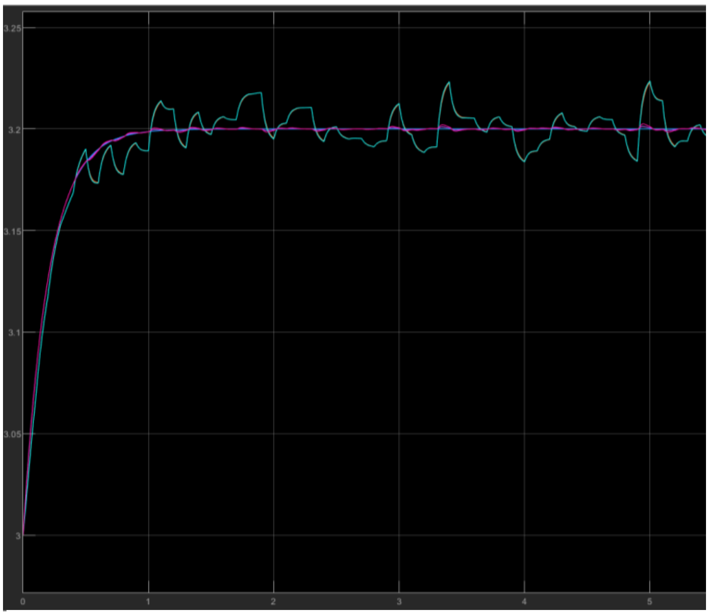
- We tried changing the values of Q, R and c
  - **R=100, Q=1*eye(2).**

  In the global case, as regards the constant tracking, the agents show a lot more noise and no one of them is close to the track. In the local the signals are corrupted by noise, less than in the global case but there is still.

  Considering the ramp tracking and the sinusoidal tracking, we saw that also in these cases the signals are more corrupted by noise but it is not so evident as in the constant tracking so we did not insert the images.



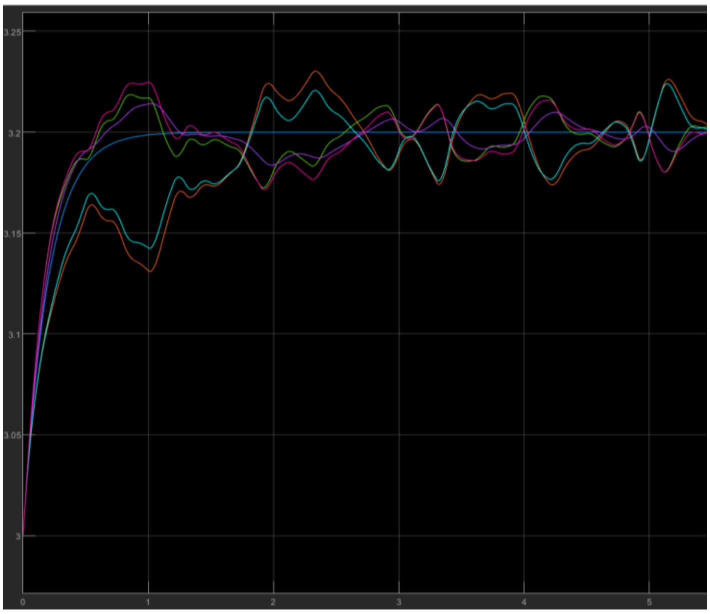
Global



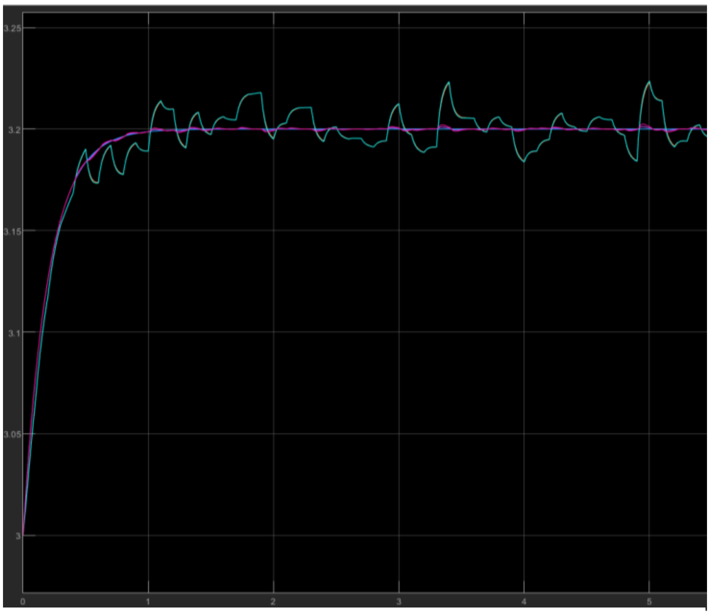
Local

- **R=1000, Q=100*eye(2).**

The local graph is the same as in the last case. The global signals otherwise are different: they are corrupted by much more noise.



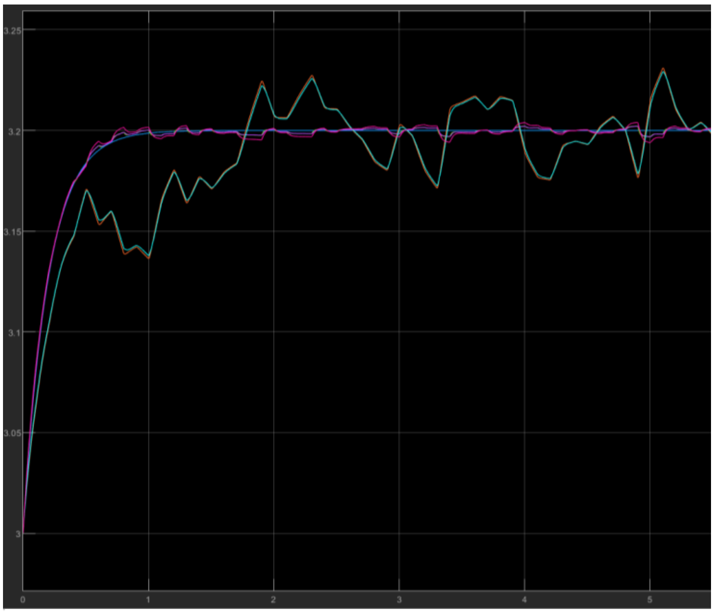
Global



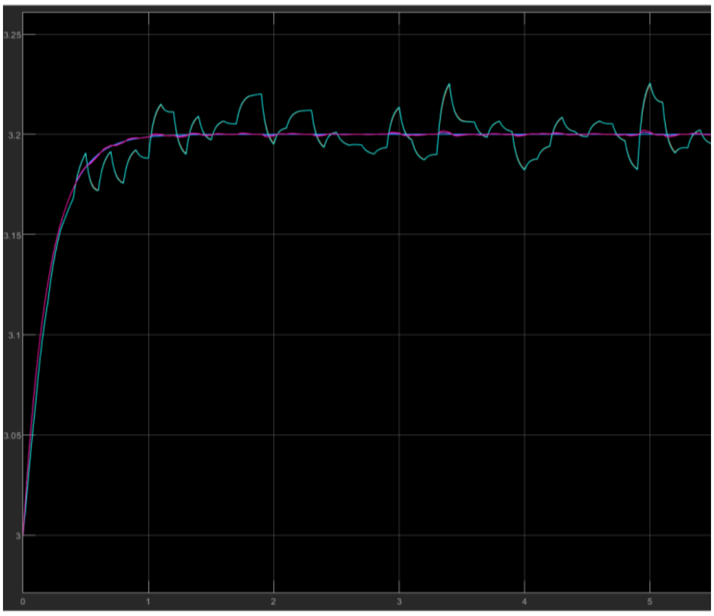
Local

- **R=10, Q=100*eye(2)**

The local estimation is always the same. The global estimation is much better than in the other cases, in fact all the signals are very close to the track (except for the noisy ones).



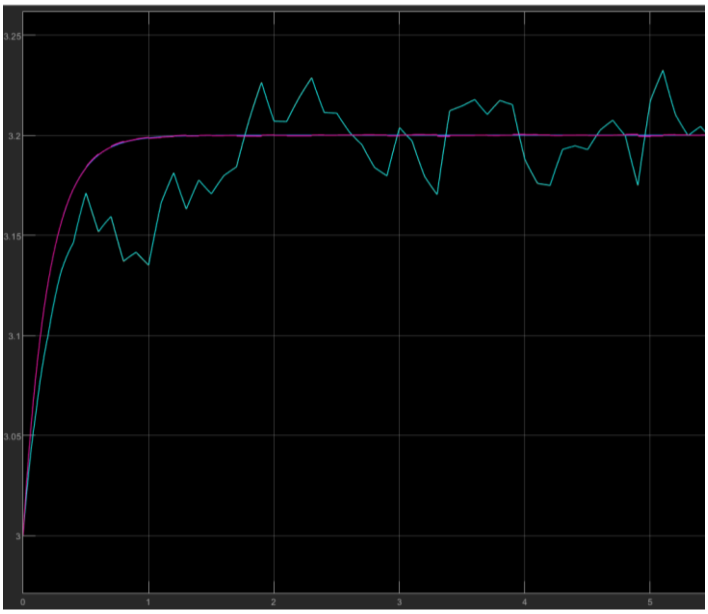
Global



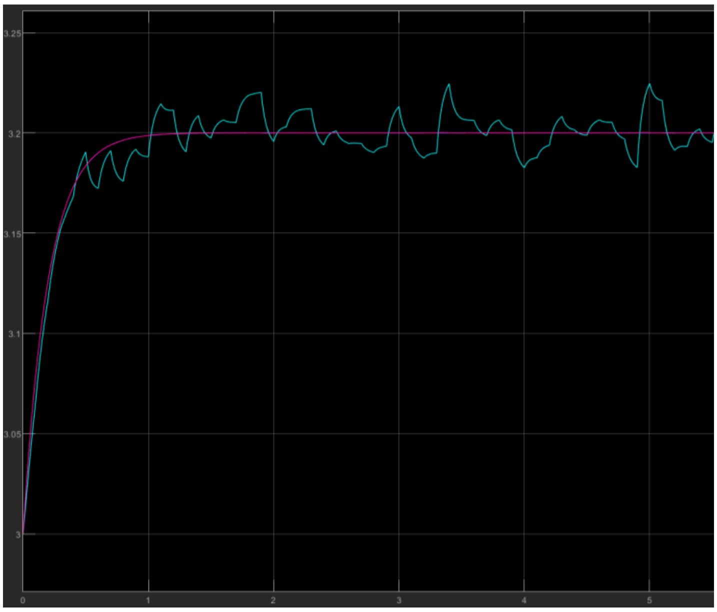
Local

- **c∝100**

The signals in both cases are corrupted by a small noise and so are closer to the track, a bigger c in this case shows the improvement with respect to the "initial case".



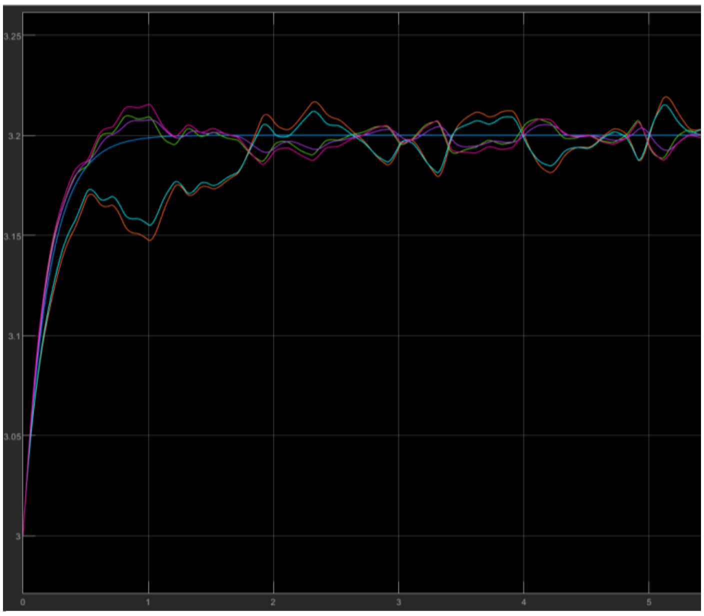
Global



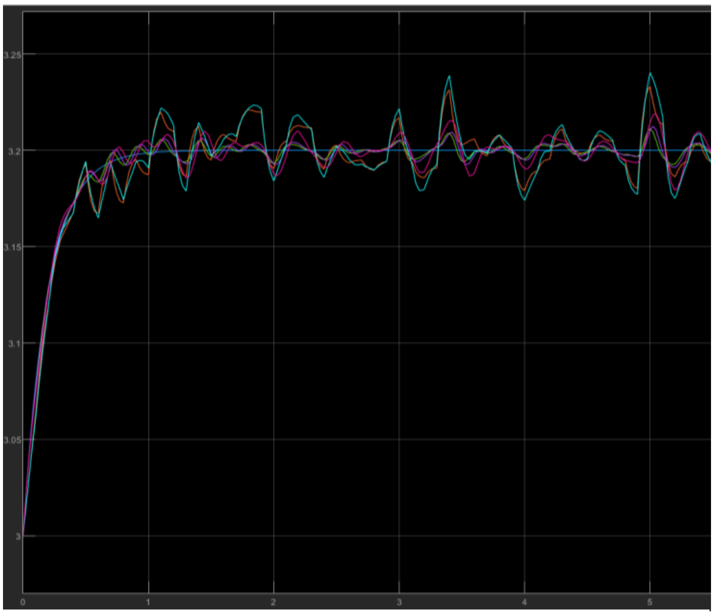
Local

- **c∝1**

Both the global and the local cases are more affected by noise.



Global



Local

In general, we can say that making c grow improves the tracking capacity of both the control protocols.

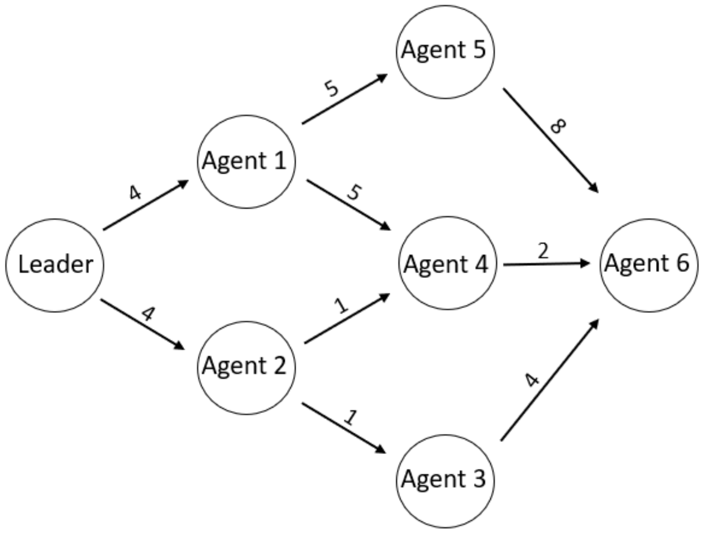
As regards R and Q, it can be seen that making Q smaller while putting R to higher values, the noise affects more the signals. On the other hand, making Q bigger than R, the signals of the agents not directly corrupted by noise are less affected by it. This is due to the fact that R and Q are the weights of the command energy and of the system energy; the following formula is used:

$$u(t) = arg \min_{u(t)} [\frac{1}{2} * \int_0^{\infty} (x'(t) * Q * x(t) + u'(t) * R * u(t))dt]$$

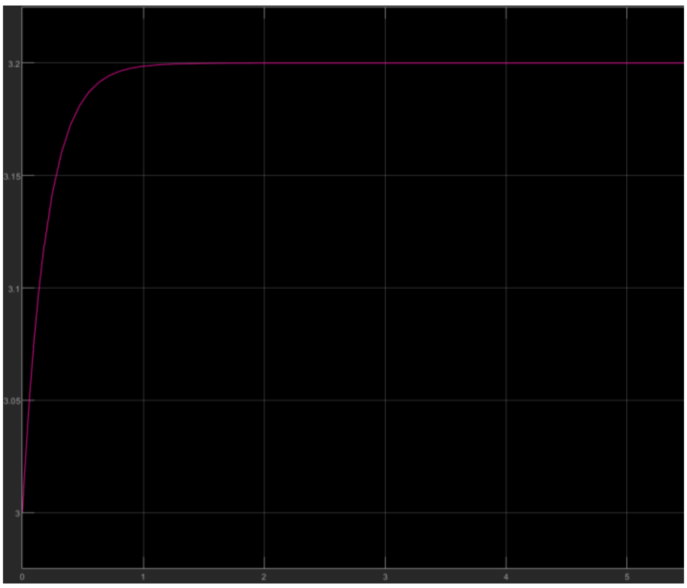
So the results will be different as the contribute of each part changes. Using a big value of R makes the command weigh more so it will be more minimized and at the end it's contribute will be less, this means that the estimate will be done not considering very much the input and the overall result will be noisier.

The changing of the Q and R parameters does not affect the local control protocol because the matrix F1 is not computed using the ARE as in the global control protocol (the ARE uses the matrices R and Q to compute F). While it is affected by the c parameter, in fact it is used in the placement of the eigenvalues of F1.

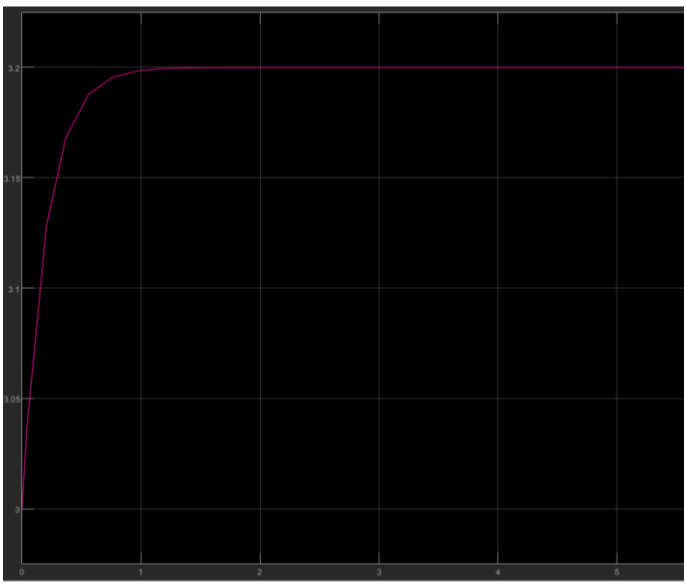
2.2. Topology 2



- Tracking to constant value, we used [0, -5] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a constant value. We tried with different values (for example [0, -3] or [0, -10]) and we saw that nothing changed except for the convergence time due to the different values of the eigenvalues.

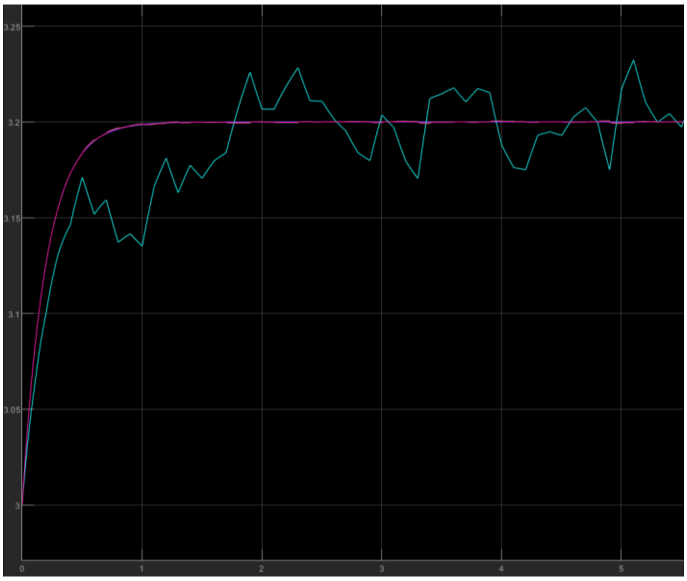


Global

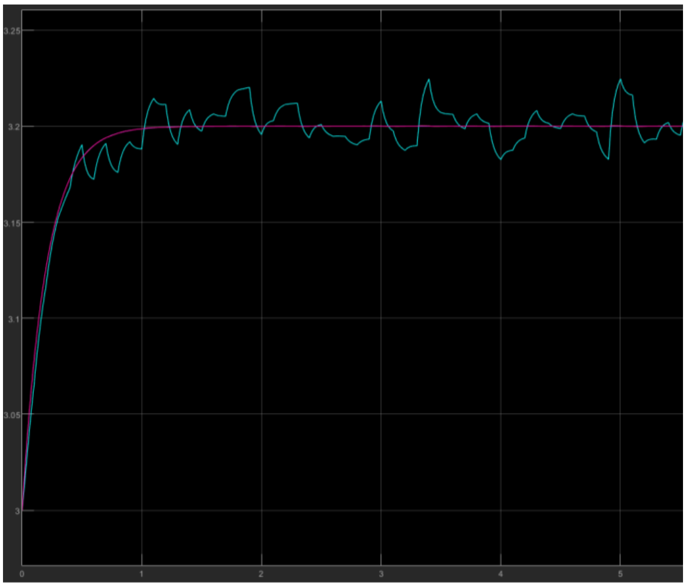


Local

We can say that there is a little difference in the two curves in terms of “design”, the global curve is less sharp than the local one.



Global

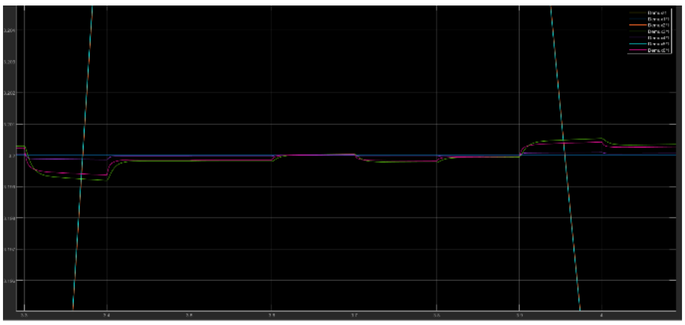


Local

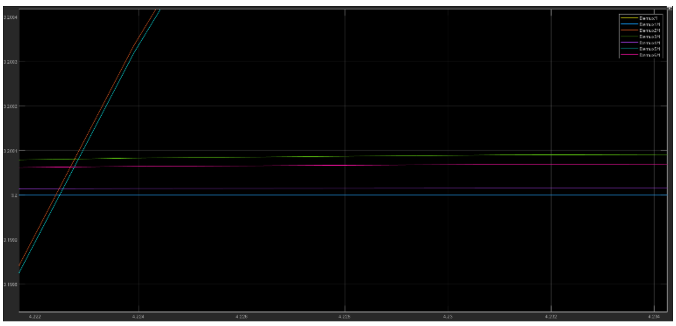
In the noise case we can say that the local estimator behaves better than the global one as the noise shown in the graph is less. As we can see in the zooms below, the noise affecting the states in the local case is much less than the global case.

From the legend in the corner, it can be seen that the agents 2 and 5 are the agents affected by the highest noise (in fact we chose the noise to directly affect them) while the others are less affected because of the compensation of the other agents.

In both cases the agent 3 is the worst, in fact it receives information only from the second node; the sixth one is slightly less than the 3rd as it receives information from more nodes and can adjust better the estimation.

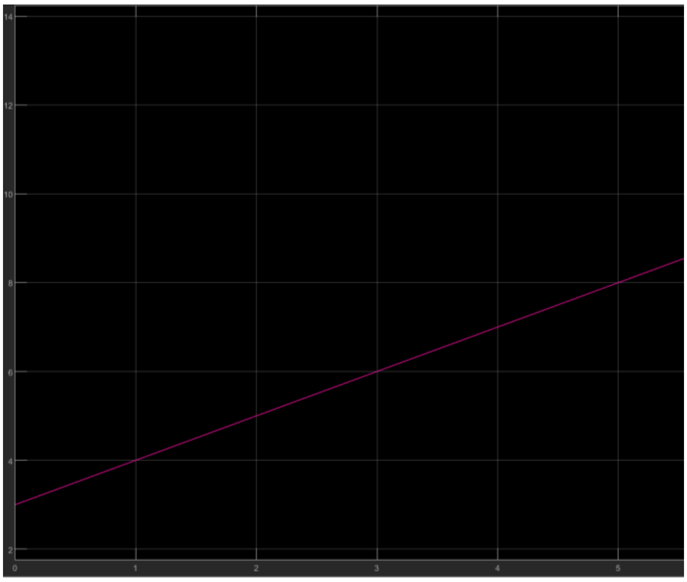


Global

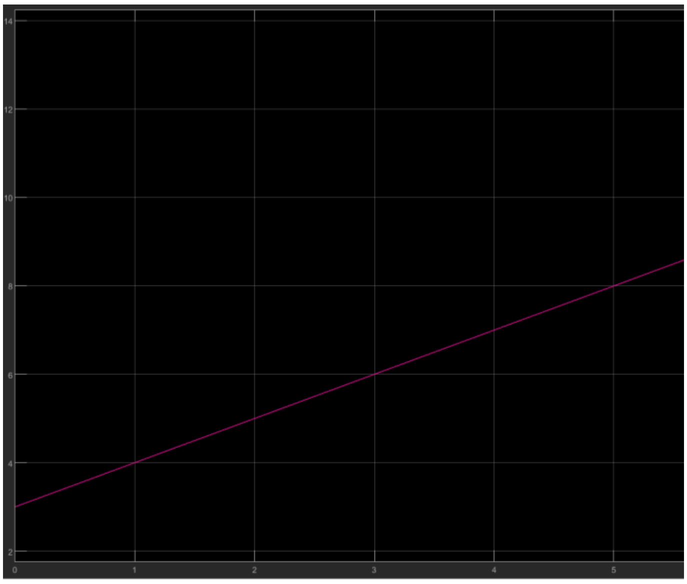


Local

- Tracking to a ramp, we used [0, 0] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a ramp.

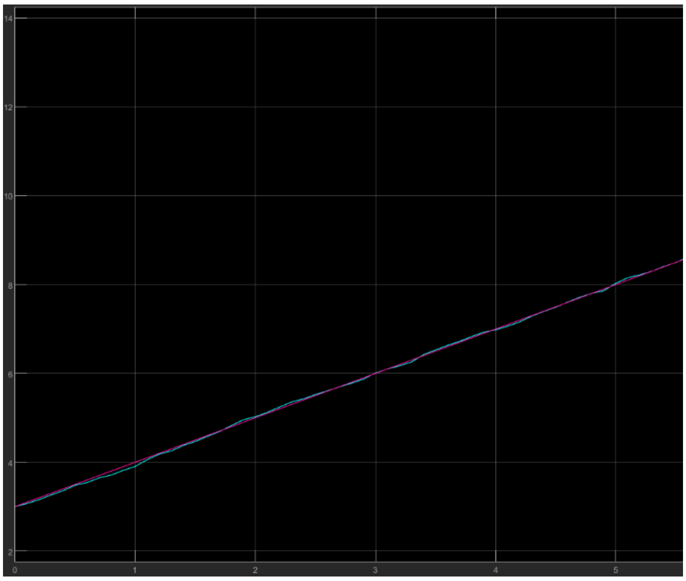


Global

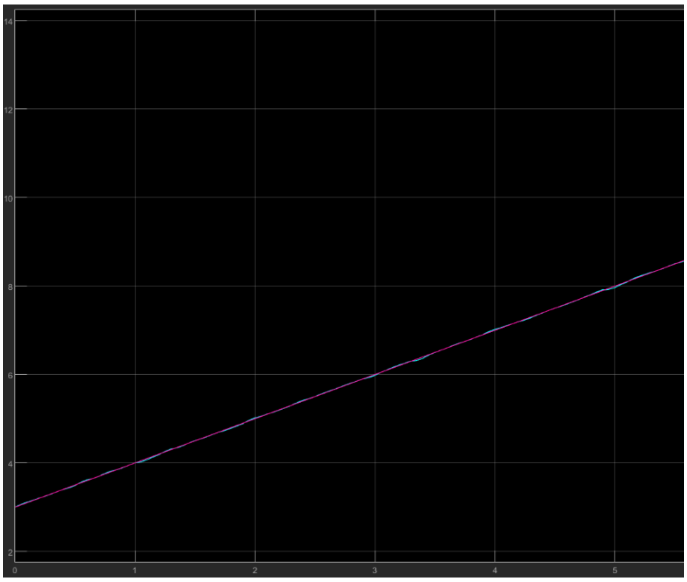


Local

Without noise no difference can be noticed between the two control protocols



Global

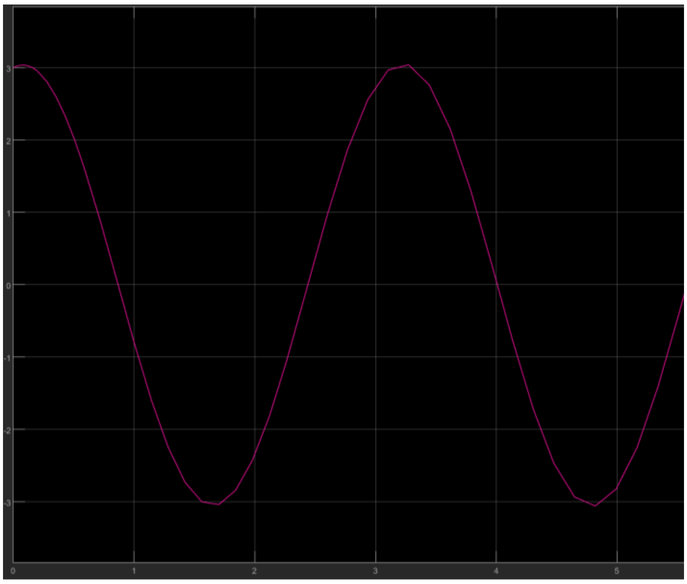


Local

As noticed in the first topology, the local graph is better than the global one.

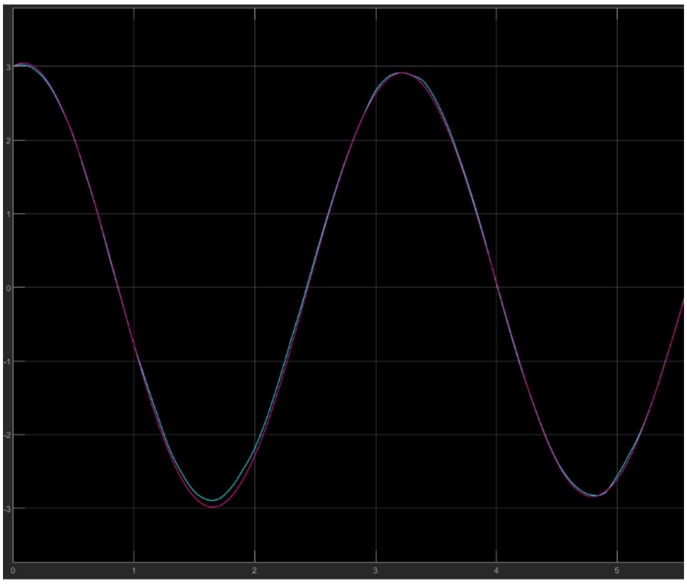
- Tracking to a sinusoidal signal, we used [-2j, 2j] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a sinusoidal signal. We tried also with other values (for example [-10j, 10j]) and we didn't notice any difference except for the frequency of the signal that is due to the magnitude of the eigenvalues.

Global

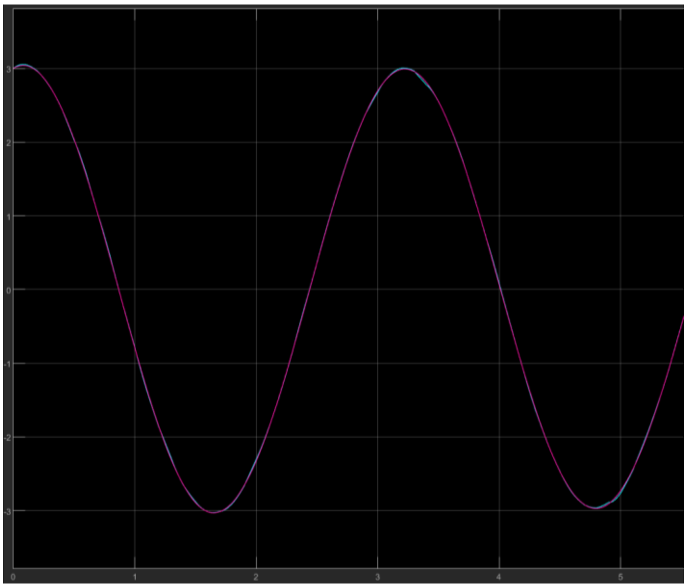


Local

No difference from the other cases, the graphs are the same.



Global

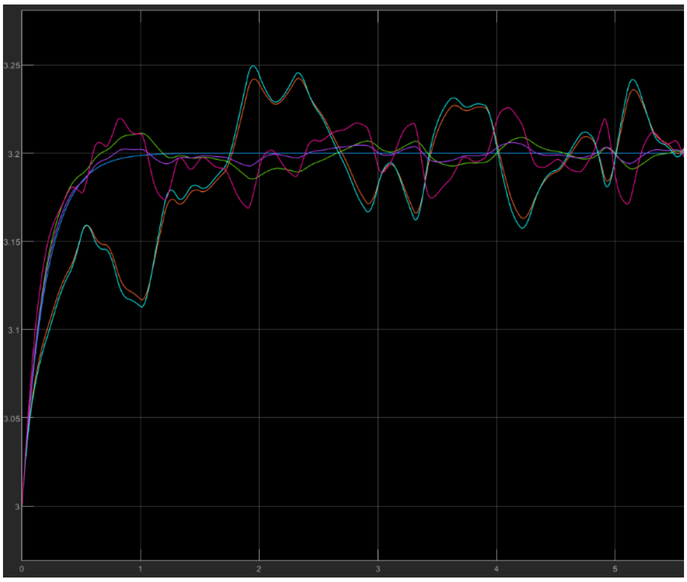


Local

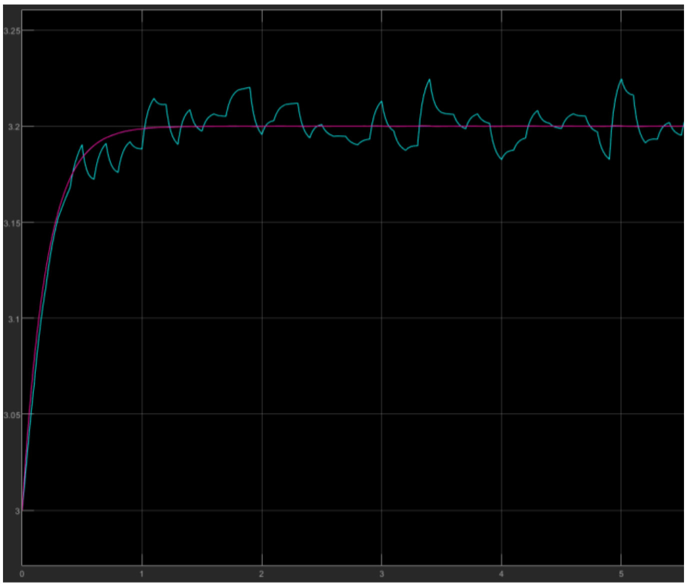
The local control protocol is still better than the global one, the signals "out of the track" are the second and the fifth ones.

- We tried changing the values of Q, R and c
  - **R=100, Q=1*eye(2).**

  In the global case, as regards the constant tracking, the agents show a lot more noise and no one of them is close to the track while in the local case nothing changes.
  Considering the ramp tracking and the sinusoidal tracking, we saw that also in these cases the signals are more corrupted by noise but it is not so evident as in the constant tracking, so we did not insert the images.



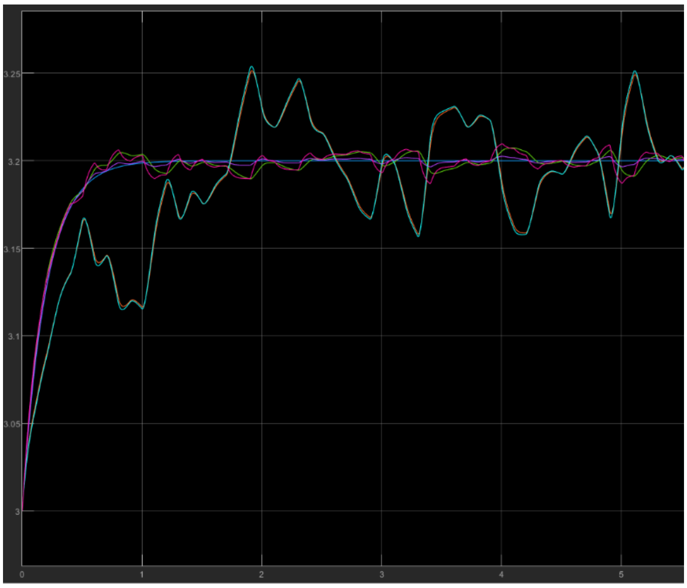
Global



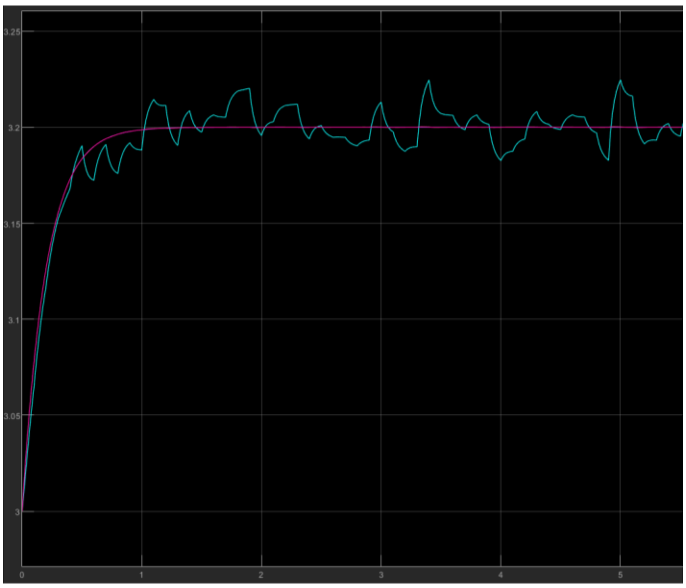
Local

  - **R=1000, Q=100*eye(2).**

  As in the previous case the local graph doesn't change. The global signals otherwise are different: the 2nd and the 5th ones are heavily corrupted by noise, the other signals are closer to the track.



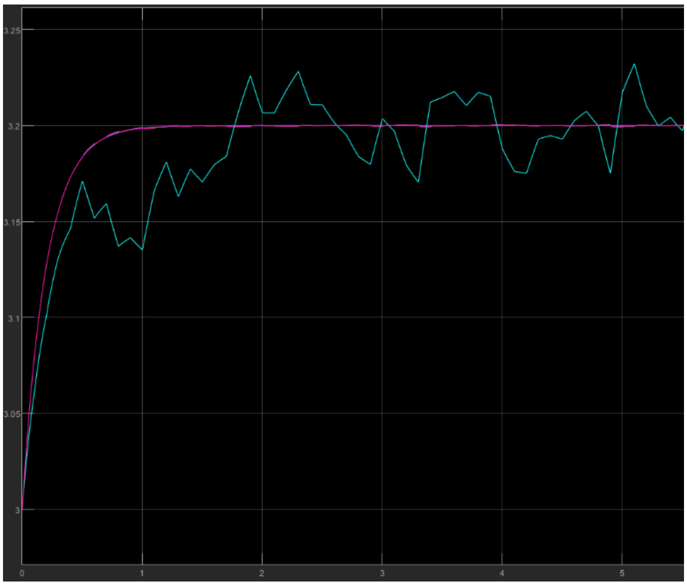
Global



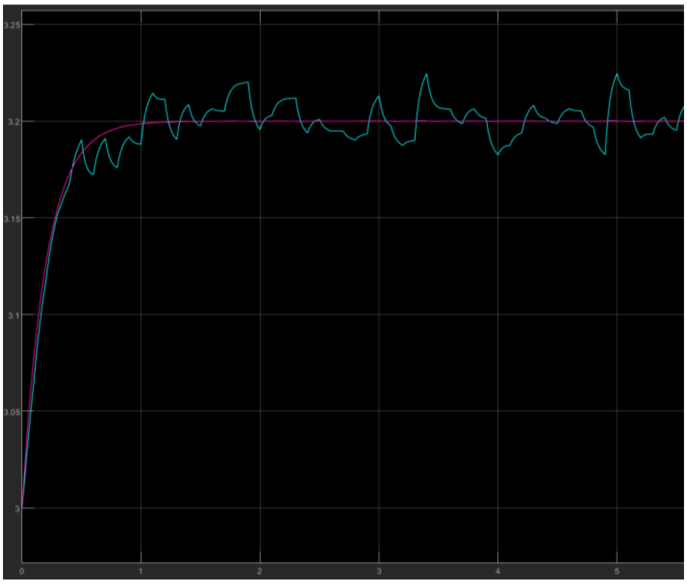
Local

- **R=10, Q=100*eye(2)**

The local estimation is always the same. The global estimation is much better than in the other cases, in fact all the signals are very close to the track (except for the noisy ones).



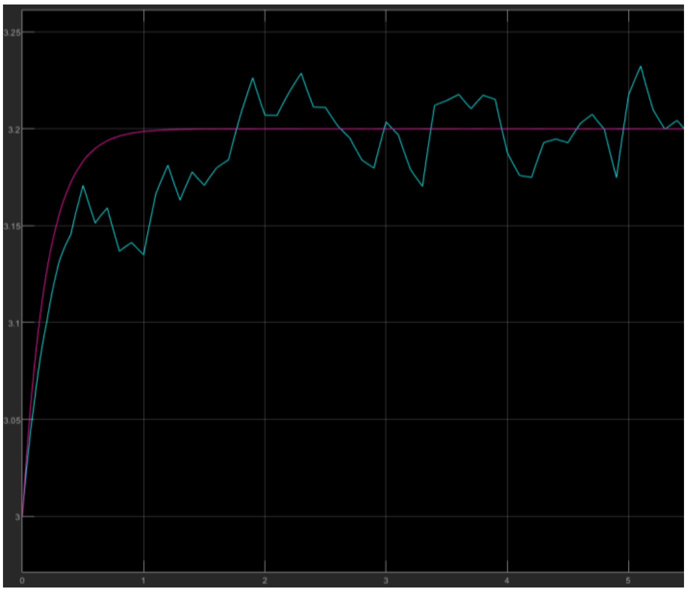
Global



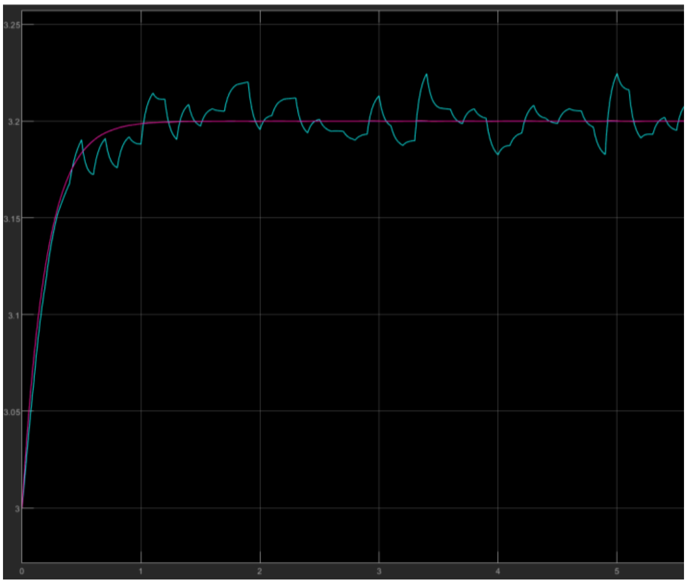
Local

- **c∝100**

The signals in the global case are a little bit closer to the track, nothing changes for the noisy signals or for the local case.



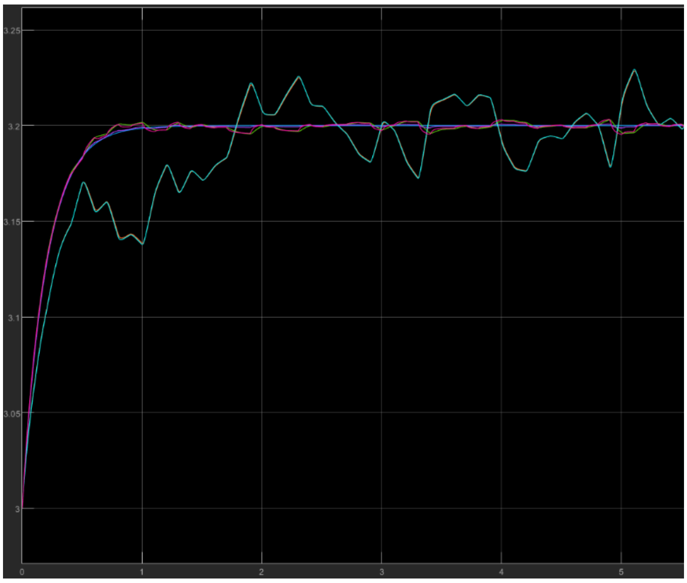
Global



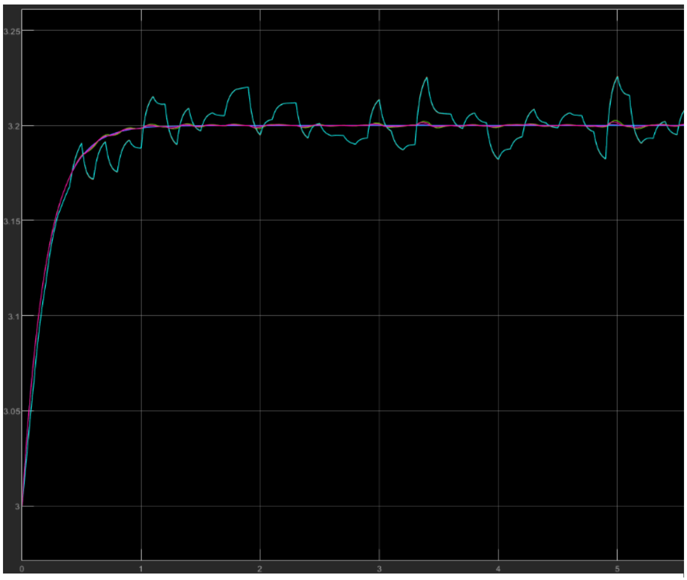
Local

- $c \propto 1$

Both the global and the local cases are more affected by noise.



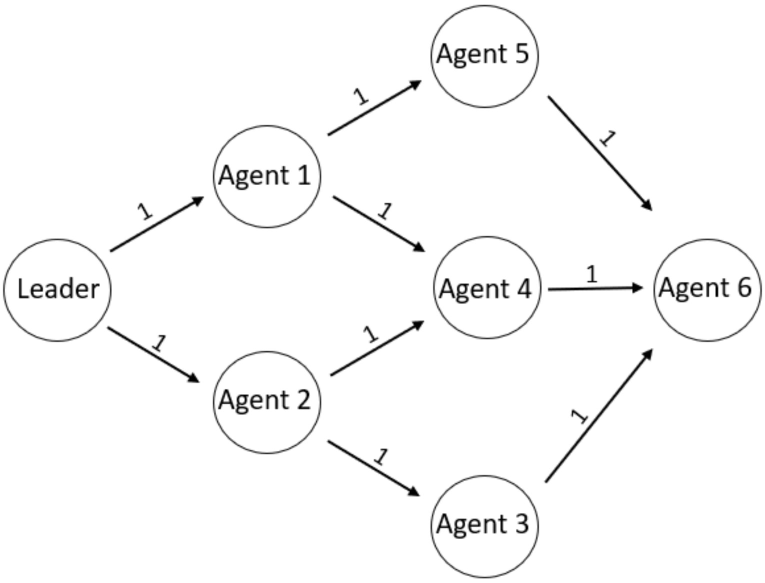
Global



Local

We noticed the same behaviour (as in the first topology) changing the parameters, so we can confirm what we explained in theory in the last topology.

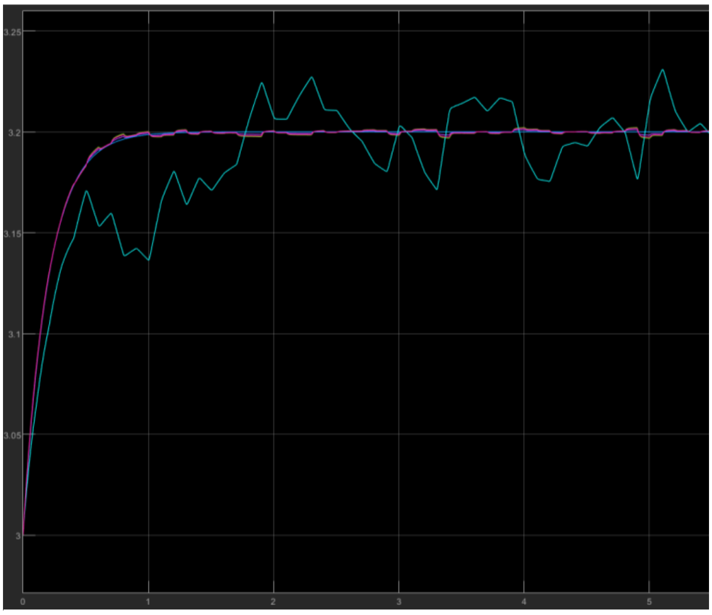
2.3. Topology 3



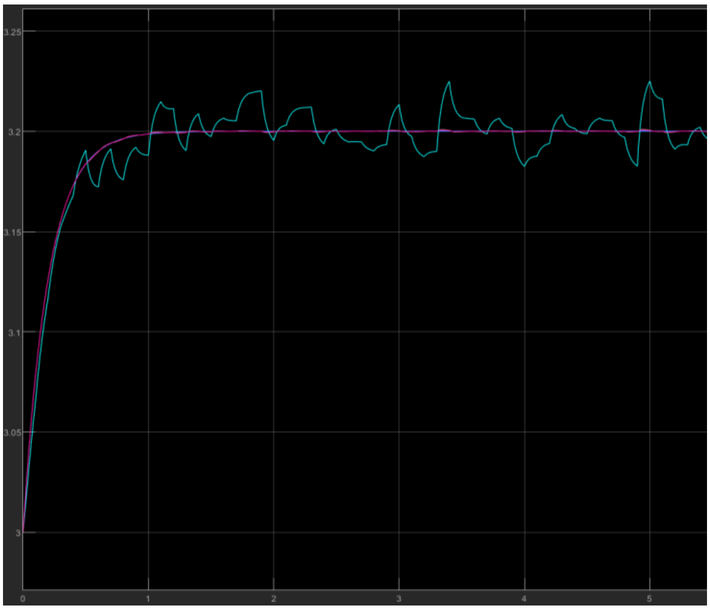
As third topology we decided to use the second one but with all the same weights.

We analysed the behaviour of the topology without noise, and we noticed that there were no differences in what we obtained (exactly as in the previous topologies) so we decided not to insert the images and to avoid commenting on them.

- Tracking to constant value, we used [0, -5] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a constant value. We tried with different values (for example [0, -3] or [0, -10]) and we saw that nothing changed except for the convergence time due to the different values of the eigenvalues.

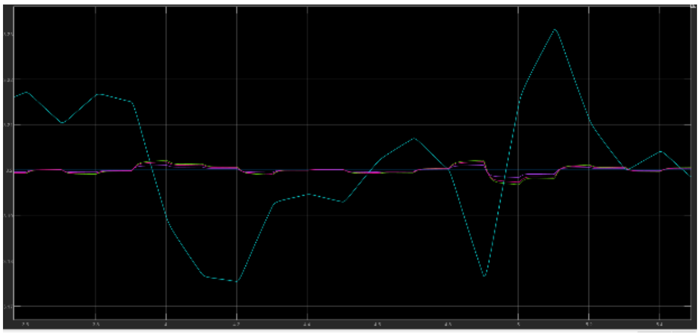


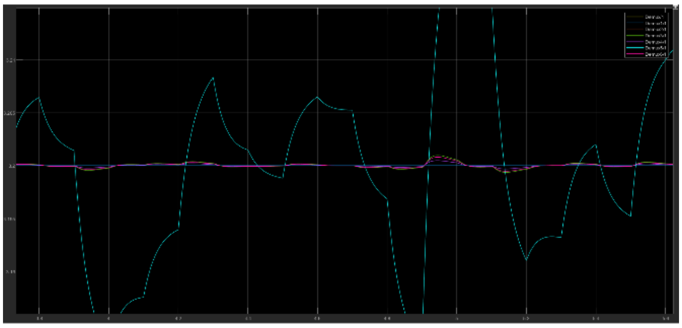
Global



Local

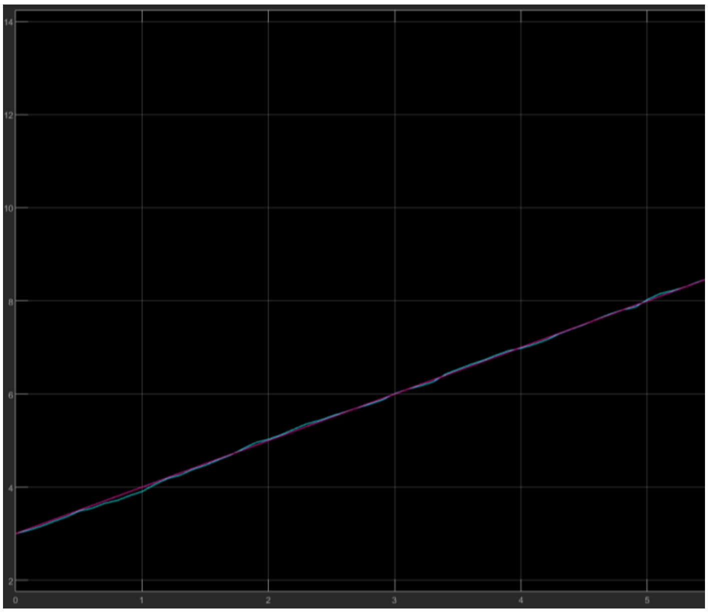
This time we can notice noise also in the local case, below there are the zooms of the graphs:



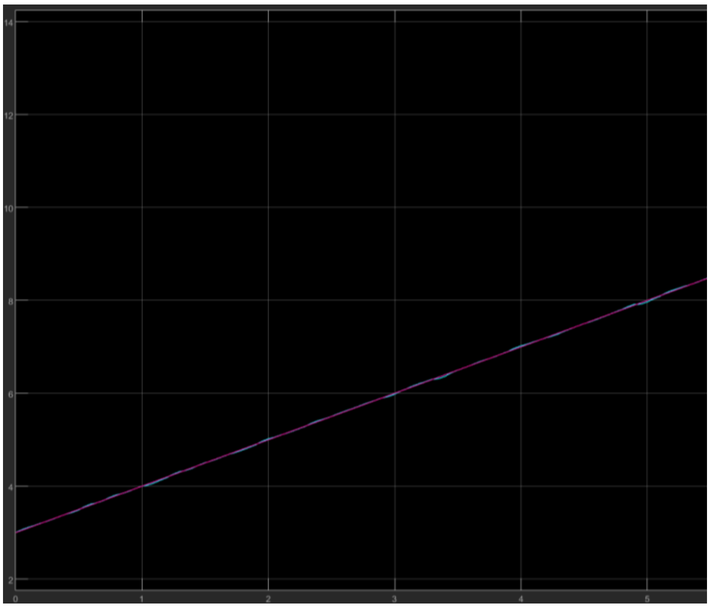


In the global case the noise is still bigger than in the local one but with this topology, the local control protocol is not able to cancel it.

- Tracking to a ramp, we used [0, 0] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a ramp.



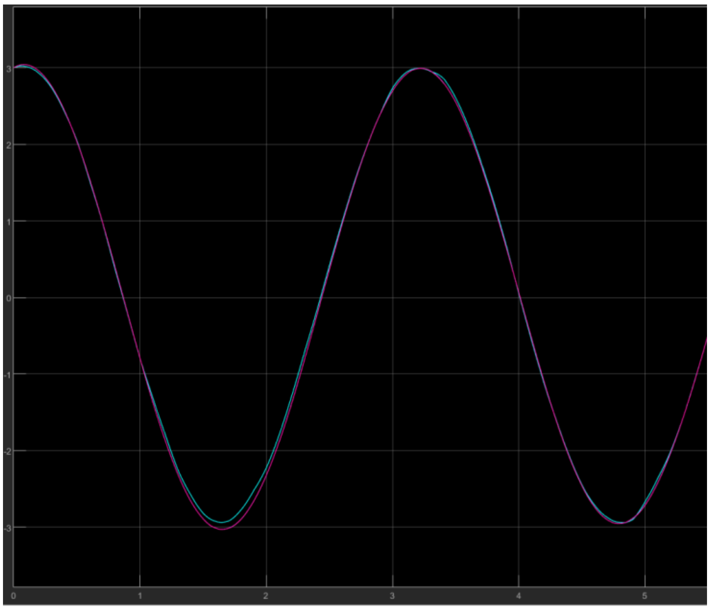
Global



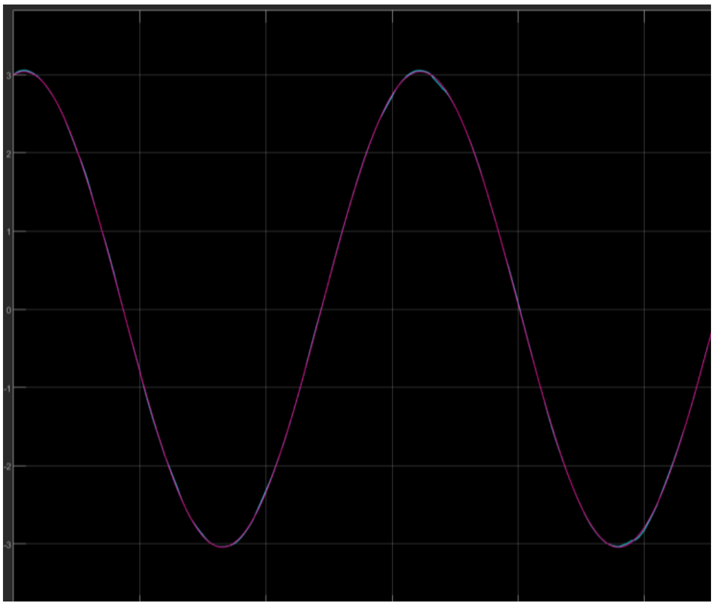
Local

Similar reasonings can be done for the ramp case, the noise in the local control protocol affects all the signals.

- Tracking to a sinusoidal signal, we used [-2j, 2j] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a sinusoidal signal. We tried also with other values (for example [-10j, 10j]) and we didn't notice any difference except for the frequency of the signal that is due to the magnitude of the eigenvalues.



Global

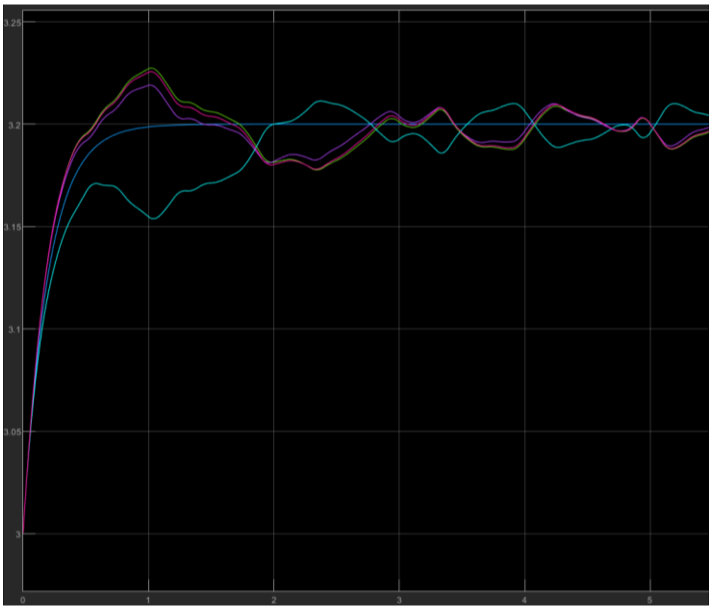


Local

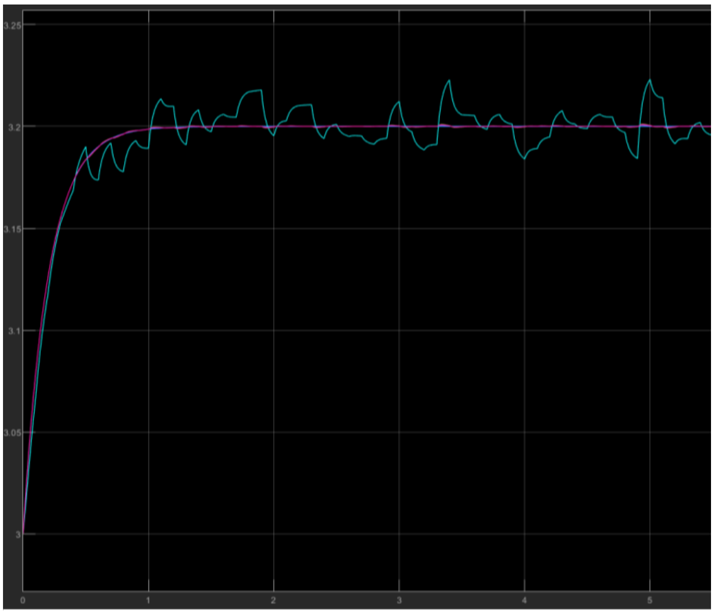
Also in this tracking case the signals in the local graph are more corrupted by noise. It can still be seen a difference between the two control protocols as the global case has more noise than the local one (this is true for all the tracking cases).

- We tried changing the values of Q, R and c
    - **R=100, Q=1*eye(2).**

    This time we can notice that all the signals (except for the noisy ones) have a similar behaviour that is due to the fact that the weights are all equal. In the local graph it is not so easy to notice the same behaviour but it can be seen that there is more noise than in the last topology.



Global

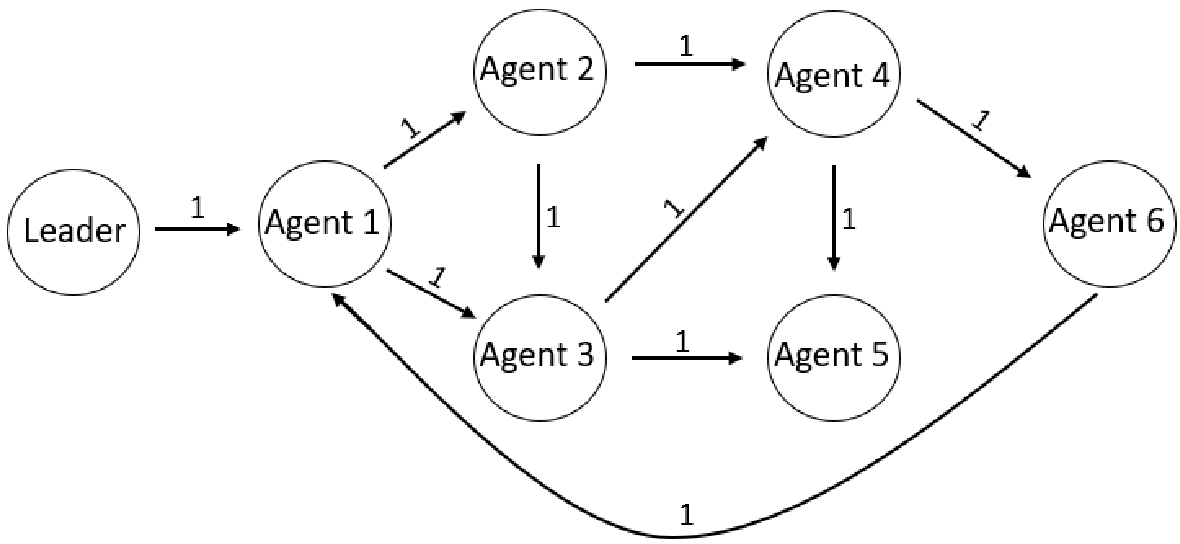


Local

We made the same trials with the parameters (R=1000,10 Q=100 c=100,1) and we saw that the behaviour was the same as in the second topology confirming that our thoughts in theory were right. The only thing that changed is that the signals' behaviours are similar because of the weights.

In conclusion we can say that this topology was interesting to study because we noticed how the weights affect the nodes. In the previous case the 4^{th} and the 6^{th} nodes received information from nodes with and without noise and thanks to the different weights in the links, they could reach a better estimation of the states. In this topology the weighting of the noise is not possible and each signal has more or less the same behaviour.

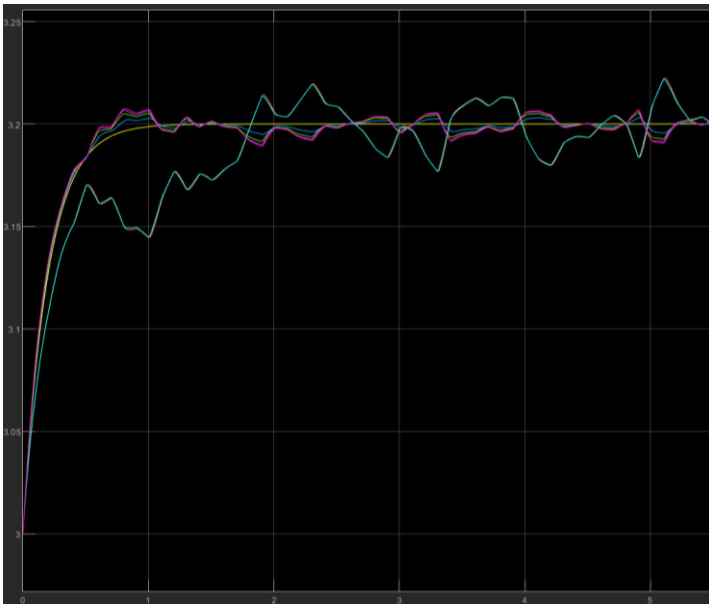
2.4. Topology 4



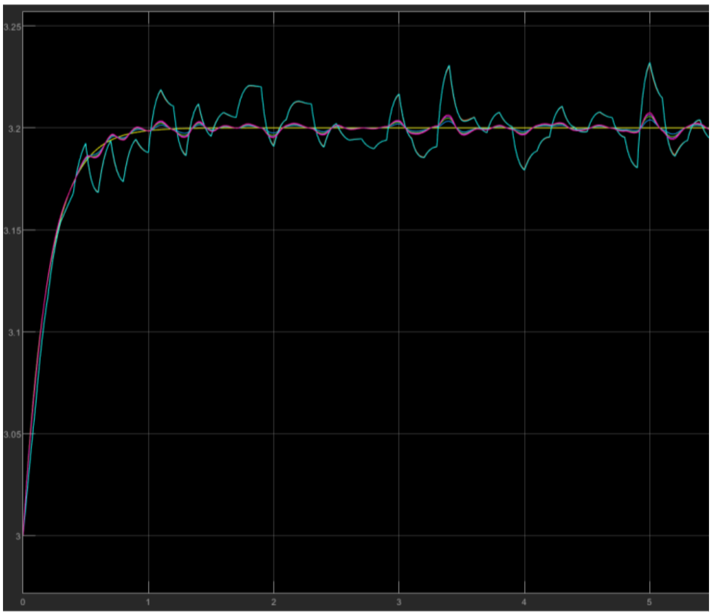
As fourth topology we decided to use a topology in which the only agent that receives information from the leader, gets also information from the last agent.

We analysed the behaviour of the topology without noise, and we noticed that there were no differences in what we obtained (exactly as in the previous topologies) so we decided not to insert the images and to avoid commenting on them.

- Tracking to constant value, we used [0, -5] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a constant value. We tried with different values (for example [0, -3] or [0, -10]) and we saw that nothing changed except for the convergence time due to the different values of the eigenvalues.



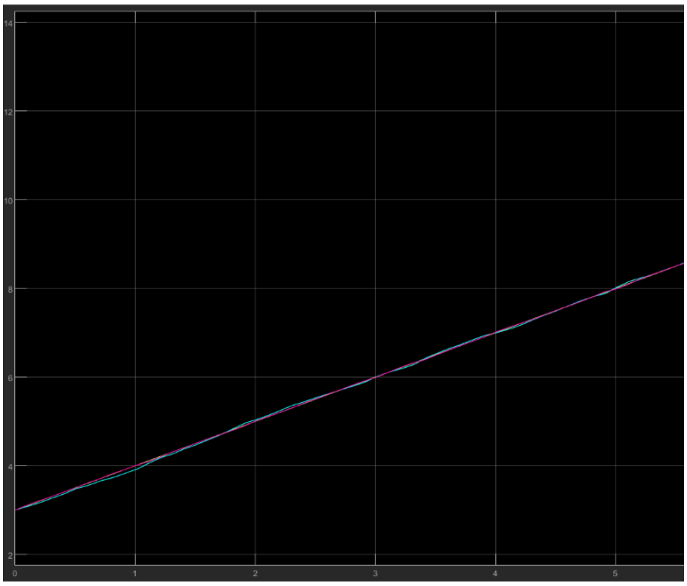
Global



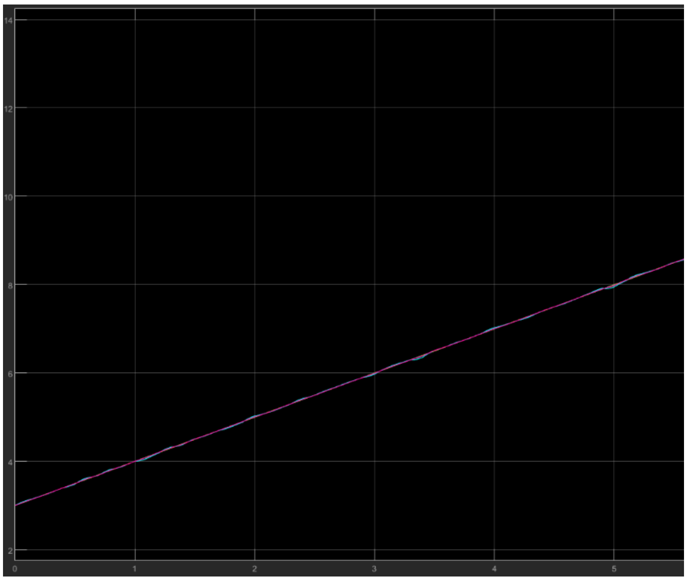
Local

We can notice that each node is affected by noise and follows more or less the same path. The link between the 6th and the 1st node propagates back the noise so neither the first agent is able to follow exactly the leader.

- Tracking to a ramp, we used [0, 0] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a ramp.



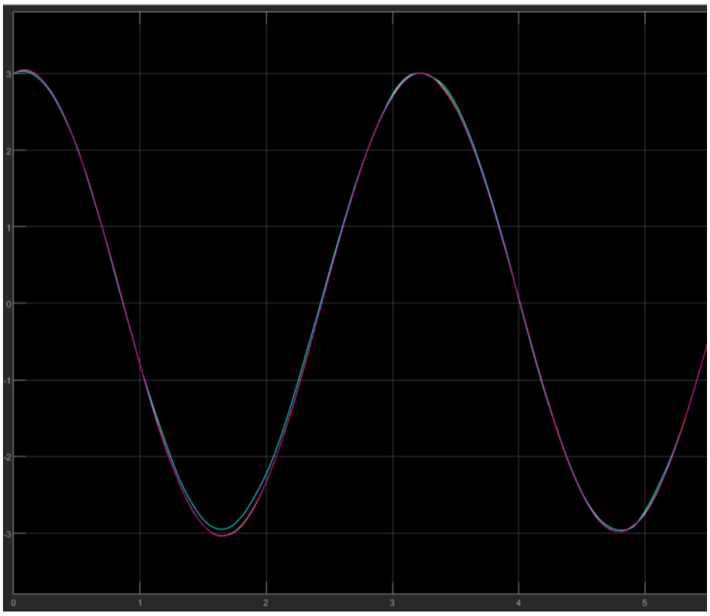
Global



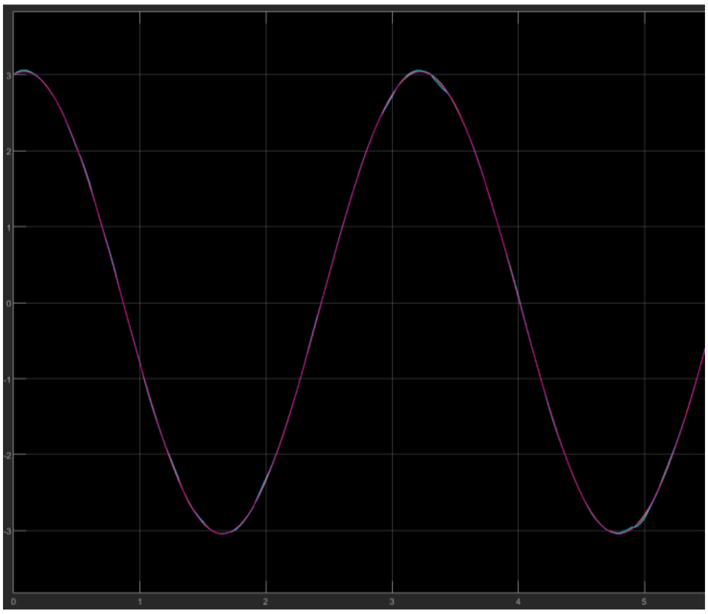
Local

The lines are shaky for the same reasons explained before.

- Tracking to a sinusoidal signal, we used [-2j, 2j] as eigenvalues to be placed in the matrix K in order to make the free response of the leader a sinusoidal signal. We tried also with other values (for example [-10j, 10j]) and we didn't notice any difference except for the frequency of the signal that is due to the magnitude of the eigenvalues.

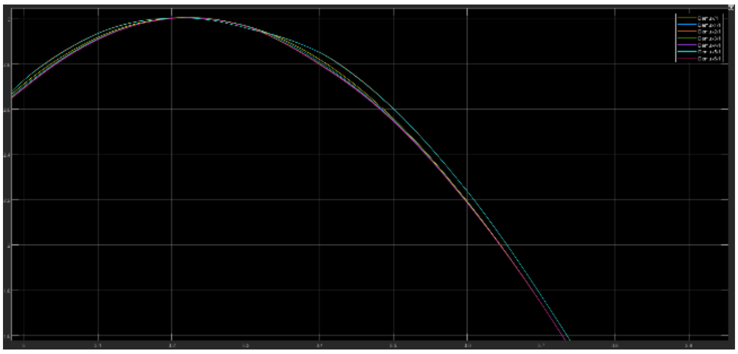


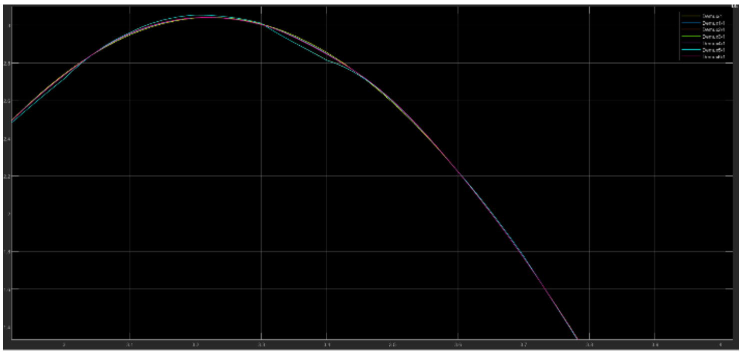
Global



Local

The sinusoidal case seems to be a little bit better than the other cases but taking a closer look (in the zooms below) we can see that the signals differ from the track.

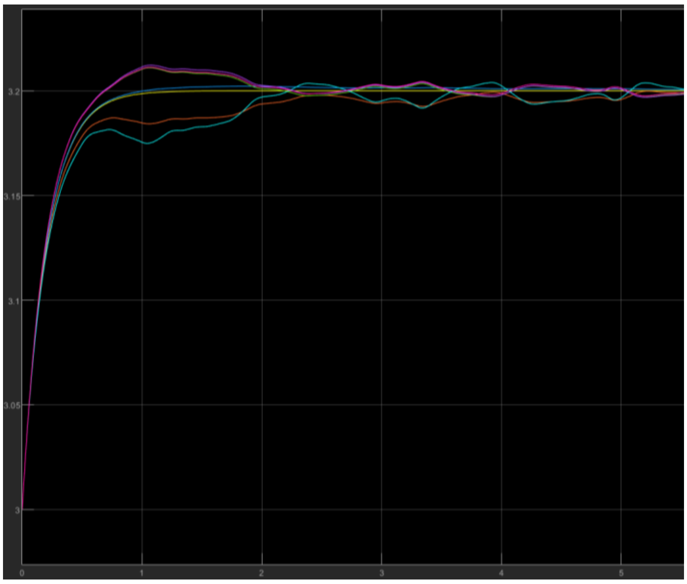




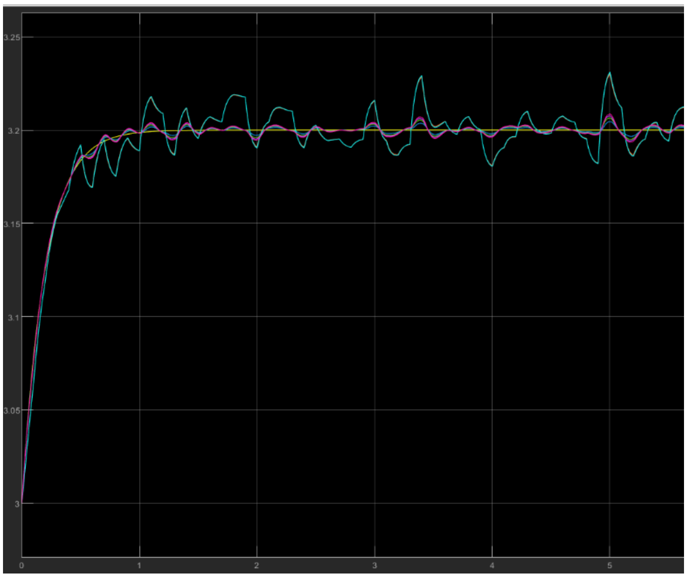
- We tried changing the values of Q, R and c
  - **R=100, Q=1*eye(2).**

  The behaviour with these parameters is interesting as it is the only case in which we can see a better performance of the global control protocol. In fact, looking at the global graph we notice that at the beginning the signals are far from the track but after a few seconds they get very close to it; the behaviour oscillates in time as after some seconds the signals are again far from the track and then closer.

  The local graph shows signals more affected by noise whose presence doesn't decrease in time, and we can also see that the fifth and the second signals are far from the track while in the global case they converge as the others.



Global



Local

We made the same trials with the parameters (R=1000,10 Q=100 c=100,1) and we saw that the behaviour was the same as in the second topology confirming that our thoughts in theory were right. The only thing that changes is the way in which the signals are corrupted by noise and their subsequent distance from the track.

In conclusion we can say that this topology was interesting to study because we noticed how the "back" link affected all the system. Because of this link, even the local control protocol was not able to eliminate the noise so the global was a little bit better in compensating for its influence.

3. Best Topology

We compared all the topologies with respect to:

- Noise management (propagation, rejection, corruption)
- Convergence time
- Response to changing parameters

We noticed that there wasn't a noticeably difference among all the topologies taking into account the convergence time. Considering the different behaviours associated to Q, R and c, the fourth topology shows a worsening in the local control protocol, while the other topologies do not differ very much. In practice we made our choice referring to the noise management, according to which we chose the 2^{nd} topology as the best one.

In fact, it is the topology that better avoids the noise propagation or at least is able to weigh its corruption on the nodes thanks to the different weight of each link and to their network's shape.