AUT.360 Distributed Control and Optimization of Cyber-Physical Systems

Homework 11

Valtteri Nikkanen 282688

Riku Pekkarinen 267084

Table of contents

Problem 1

We are asked to consider an undirected graph G = (V, E). The graph G is given in figure 1. We are to assume that there is one malicious node in the graph.

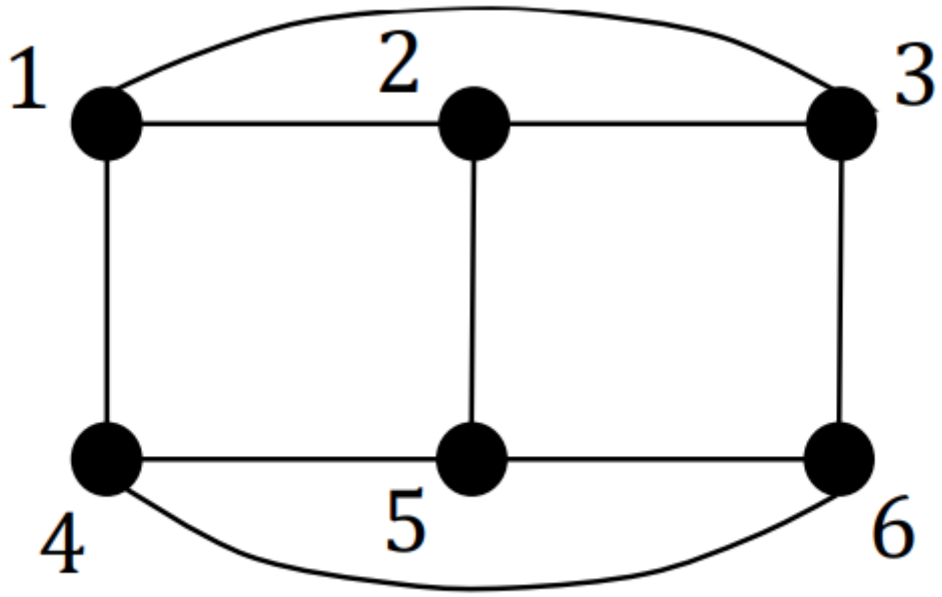

Figure 1: Graph G for problem 1

The starting conditions for nodes are $x_i(0) = 2$ for all $i \in \{1,2,3\}$ and $x_i(0) = 0$ for all $i \in \{4,5,6\}$.

If all the agents update their states according to the local filtering-based consensus protocol where each agent disregards the largest and smallest values when compared to its own value, we need to show that consensus cannot be reached. This is of course inevitable as each node in set $\{1,2,3\}$ will disregard the value of the node in $\{4,5,6\}$ and vice versa, because they start from different values. So, the sets will only take the values from another node in the same set as itself.

The simulation results are presented in figure 2. From the results we can see that our assumption before was correct, and the agents did not reach consensus but kept their original values.

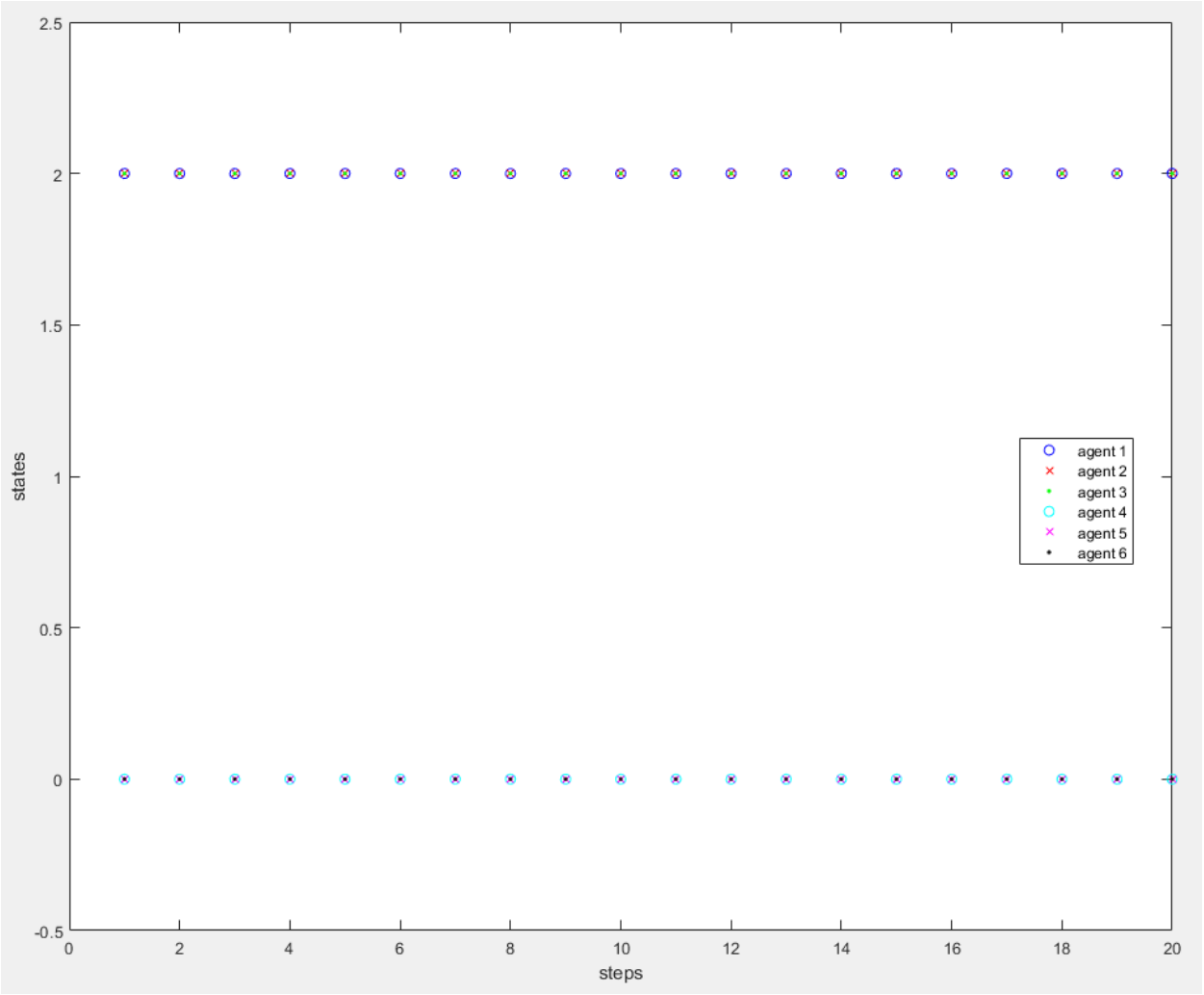

Figure 2: Simulation of the graph G in figure 1 with local filtering-based consensus protocol with the assumption of 1 malicious agent

We are then asked to modify the graph G in figure 1 to be a 2F+1 -robust graph. Given that there is 1 malicious node so that F = 1. Our graph needs to be modified to be 3-robust. A graph is r-robust if for any two, non-empty and disjoint subsets, at least one of the sets is r-reachable. In other words, no matter how the graph is divided into two subsets at least one node needs to have r neighbours outside of that subset. The updated graph is given in figure 3.

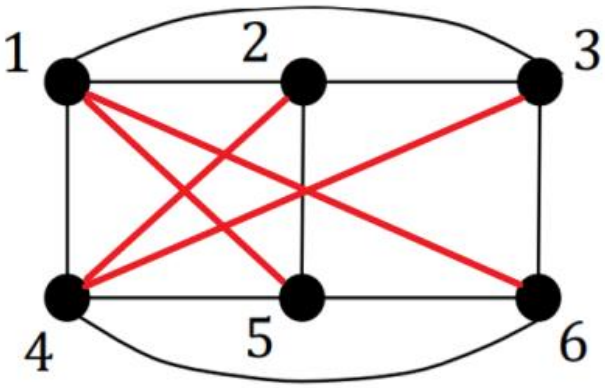

Figure 3: 3-robust graph modified from the original graph in figure 1.

By using the following graph in our simulation, we were able to produce the results in figure 4. We can see that the network reached consensus. The consensus was not exactly average but close to it. As the average would have been 1 and the graph reached a consensus value of 0.9.

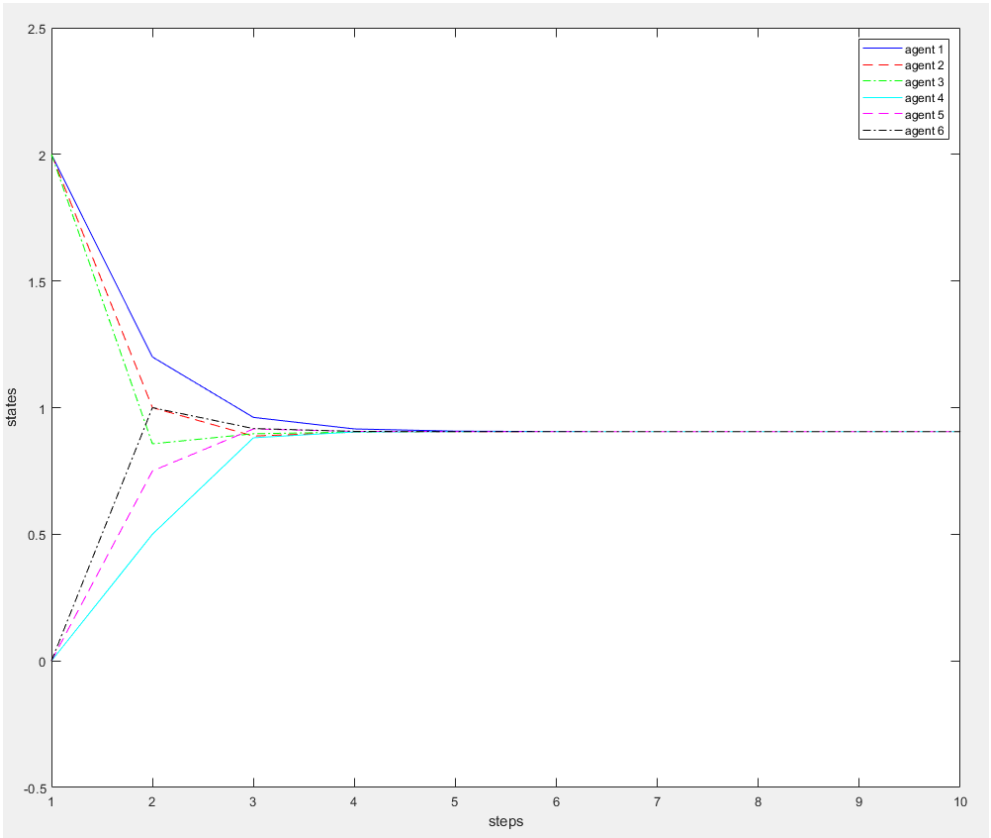

Figure 4: 2F+1 -robust graph G in figure 3

Now we update the last agent 6's state in the following way:

$$x_6(k+1) = x_6(k) + v \tag{1}$$

where v is Gaussian noise. All non-malicious agents know about a threat but not which agent it is. The results of this are presented in figure 5.

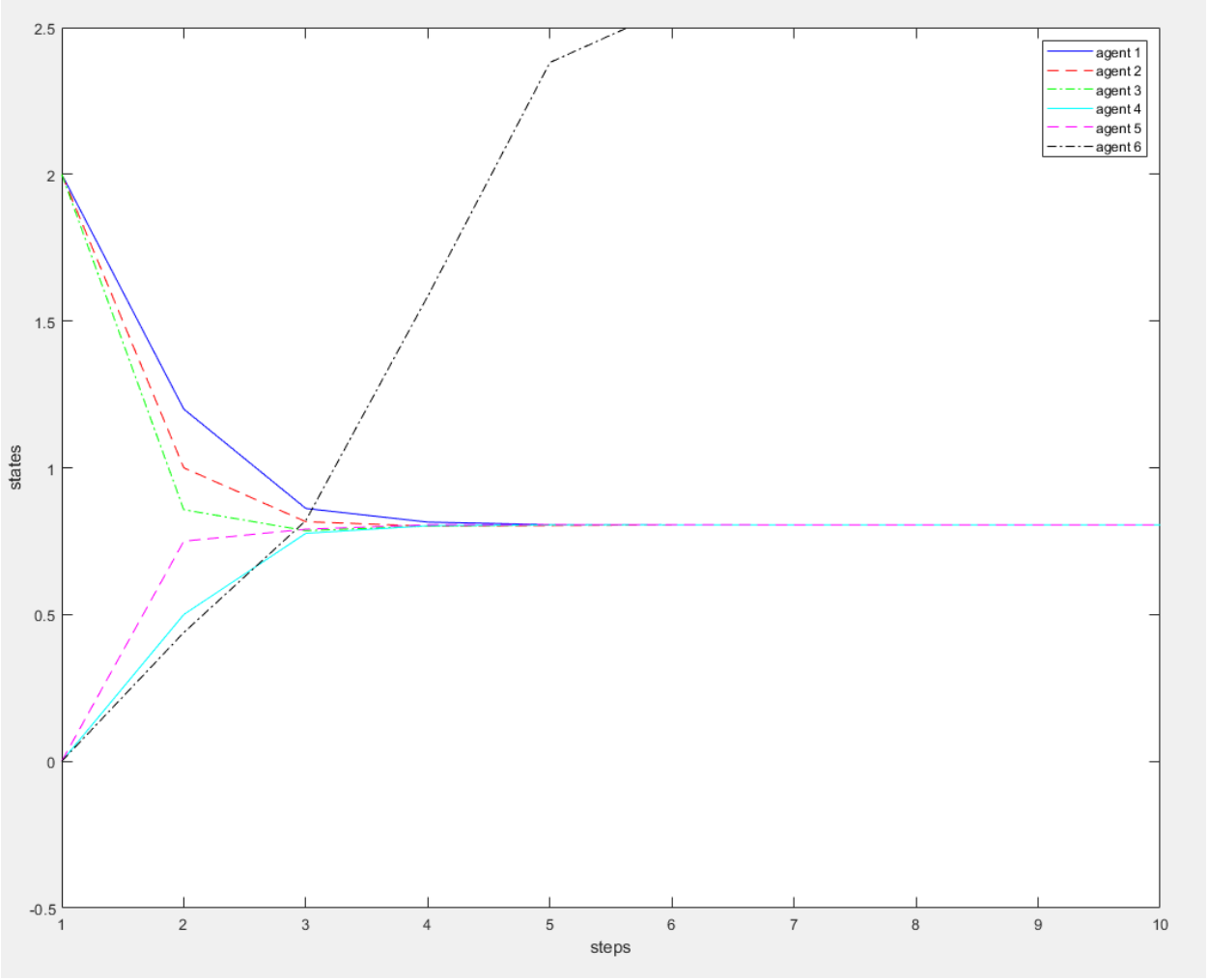

Figure 5: The robust graph with the filtering algorithm where agent 6 is not updating correctly but just receiving random gaussian noise.

As we can see from figure 5 the other agents reach consensus nicely as the filtering allows them to ignore the compromised node. The agent 6 then wanders wherever it is going but has no effect on the rest of the agents to perform safely when reaching consensus.

Problem 2

Now considering undirected ring topology of six agents where agents update their states according to the following consensus protocol:

$$\dot{x}_i = (x_{i-1} - x_i) + (x_{i+1} - x_i),\ modulo\ n, \text{for all } i\ \in \{1, \dots, 6\} \quad (2)$$

In vector form

$$\dot{x} = -Lx \quad (3)$$

Where L is the Laplacian matrix of the graph.

We assume a cyber-attack where the malicious information is injected on the communication link and the feedback loop of the agents.

$$\dot{x} = -L(x - d) \quad (4)$$

where d is any bounded value designed by the attacker to cause maximum dispersion from the original consensus values in the system.

The dynamics of the attack:

$$\dot{d} = -0.1Id + k_a Ix \quad (5)$$

where k_a is chosen by the attacker to produce maximum dispersion.

We need to design k_a to cause maximum dispersion in the system. From the equations 4 and 5 we can make the following:

$$\begin{aligned} \dot{x} &= -Lx + Ld \\ \dot{d} &= -0.1Id + k_a Ix \end{aligned} \quad (6)$$

$$\begin{pmatrix} \dot{x} \\ \dot{d} \end{pmatrix} = \begin{pmatrix} -L & L \\ k_a I & -0.1I \end{pmatrix} \begin{pmatrix} x \\ d \end{pmatrix}$$

Where the matrix that is multiplying the x and d is also known as M. We also know that the L matrix is

$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

We can now calculate the eigenvalues

$$\det(\lambda I - M) = 0$$

$$det\begin{pmatrix} \lambda I + L & -L \\ -k_a I & \lambda I + 0.1I \end{pmatrix} = 0$$

$$\det\big((\lambda I + L)(\lambda I + 0.1I) - k_a L\big) = 0$$

$$\det(\lambda^2 I + \lambda(0.1I + L) + (0.1 - k_a)L) = 0$$

The eigenvalues of L are 0 ,1, 1, 3, 3, and 4.

$$\lambda_i = \frac{-(0.1 + \lambda_i(L)) \pm \sqrt{(0.1 + \lambda_i(L))^2 - 4(0.1 - k_a)\lambda_i(L)}}{2}$$

$$\lambda_i(L) = 0 \rightarrow k_a \text{ has no role}$$

$$\lambda_i(L) = 1 \rightarrow \frac{-1.1 \pm \sqrt{4k_a + 0.81}}{2}$$

$$\lambda_i(L) = 3 \rightarrow \frac{-3.1 \pm \sqrt{12k_a + 8.41}}{2}$$

$$\lambda_i(L) = 4 \rightarrow \frac{-4.1 \pm \sqrt{16k_a + 15.21}}{2}$$

From these we can calculate that if $k_a > 0.1$ the eigenvalues will go above 0 resulting in instability in the system. In figure 6 the network is run with the $k_a = 0$, in figure 7 the same simulation is run with $k_a = 0.1$ and lastly in figure 8 it is simulated with $k_a = 0.2$. From these simulations we can see that our calculations were correct, and we can cause the network to run into problems with this attack.

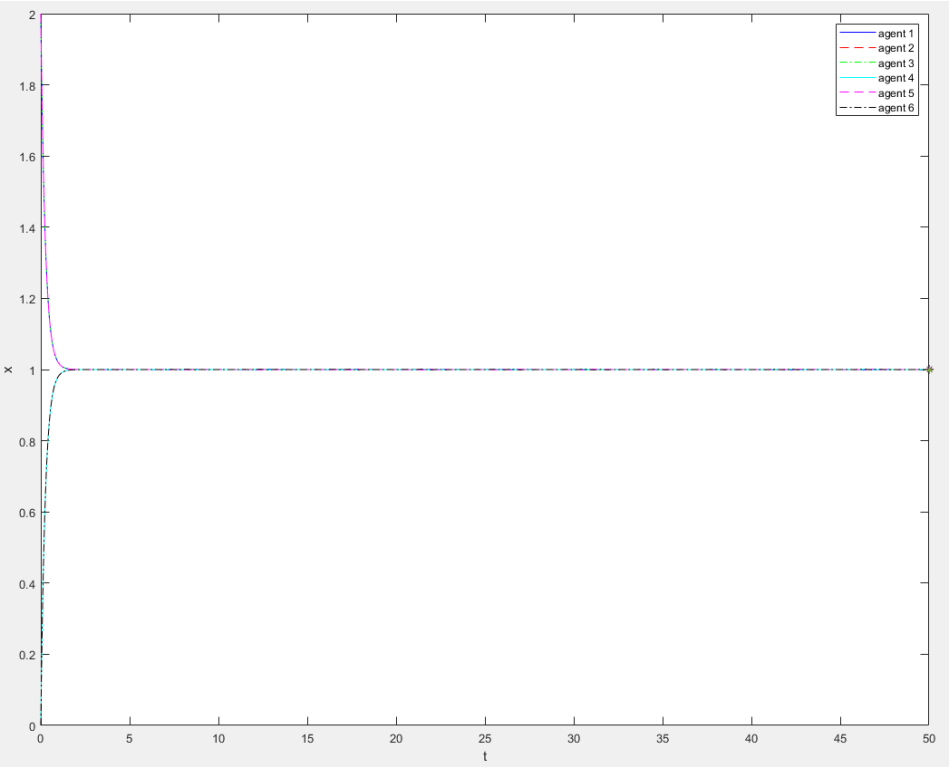

Figure 6: Network simulation when $k_a = 0$

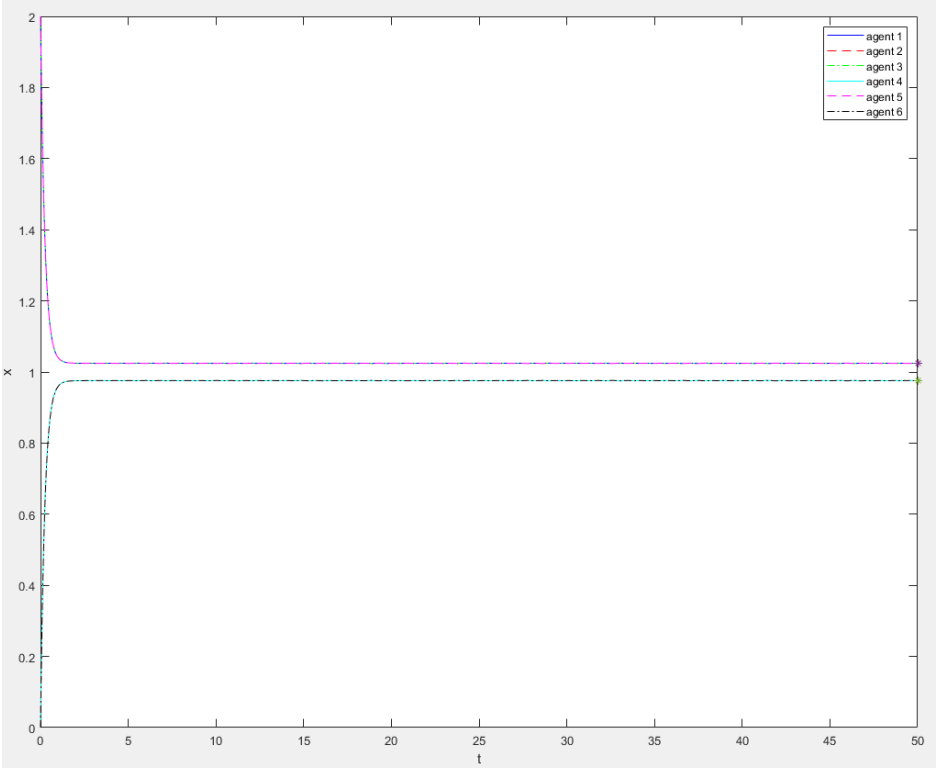

Figure 7: Network simulation when $k_a = 0.1$

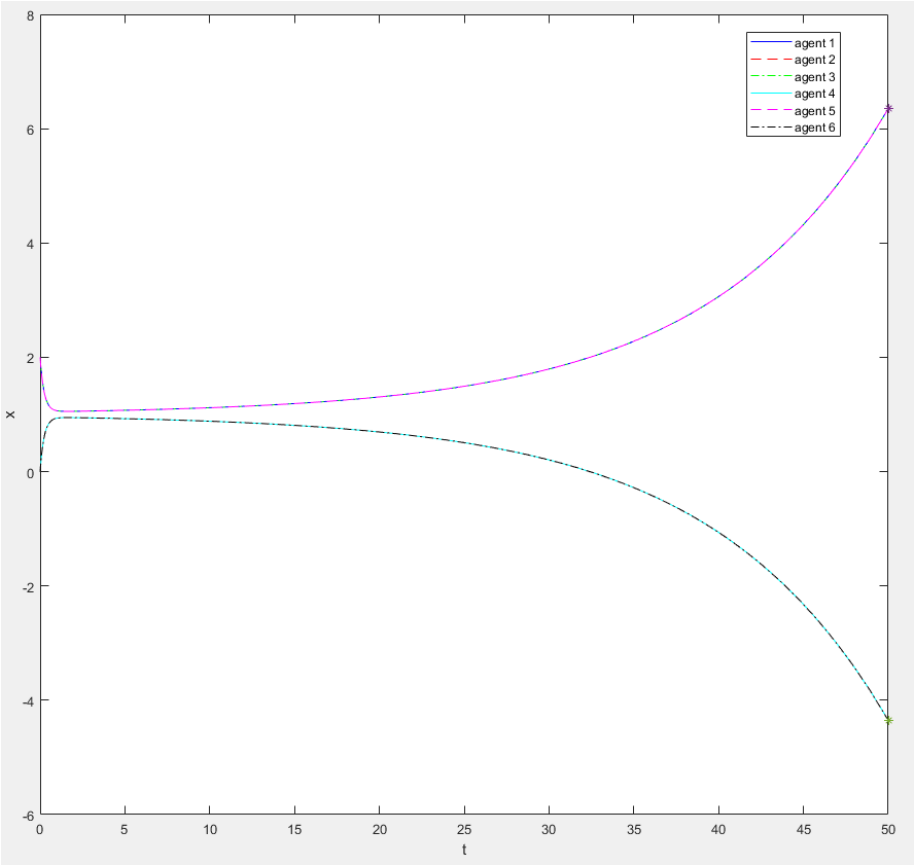

Figure 8: Network simulation when $k_a = 0.2$

Now we design a virtual network-based consensus protocol z, that enables agents to achieve consensus in absence of attacks.

We need to add to the equation 6 dynamics for the virtual network z that will help us counter the attacks on the system.

$$\begin{aligned} \dot{x} &= -Lx + \beta Lz + Ld \\ \dot{z} &= -\beta Lx - Lz \\ \dot{d} &= -0.1Id + k_a Ix \end{aligned} \tag{7}$$

Which gives us the following vector form:

$$\begin{pmatrix} \dot{x} \\ \dot{z} \\ \dot{d} \end{pmatrix} = \begin{pmatrix} -L & \beta L & L \\ -\beta L & -L & 0 \\ k_a I & 0 & -0.1I \end{pmatrix} \begin{pmatrix} x \\ z \\ d \end{pmatrix}$$

In equation 7 the β is a scalar gain larger than zero. And the initial values for z are arbitrary. The results for the new virtual network-based consensus protocol while not under attack are plotted in figure 9 and when we put the physical network under attack with the $k_a = 0.2$ which previously caused the network to not reach consensus the network now reaches consensus. This is shown in figure 10.

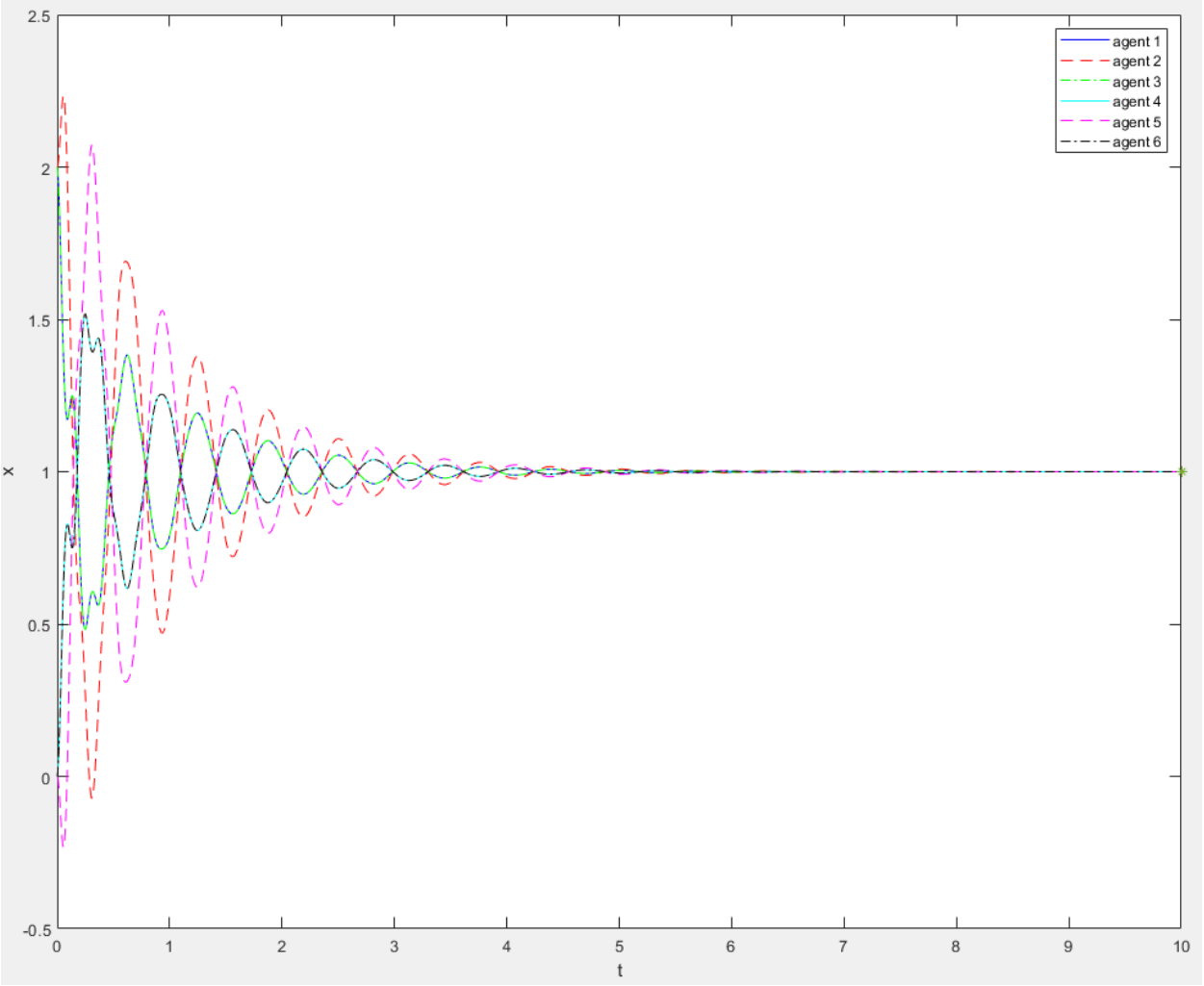

Figure 9: Virtual network-based consensus protocol when not under attack

We can see that the consensus value is the same as the consensus value without the virtual network as it should be.

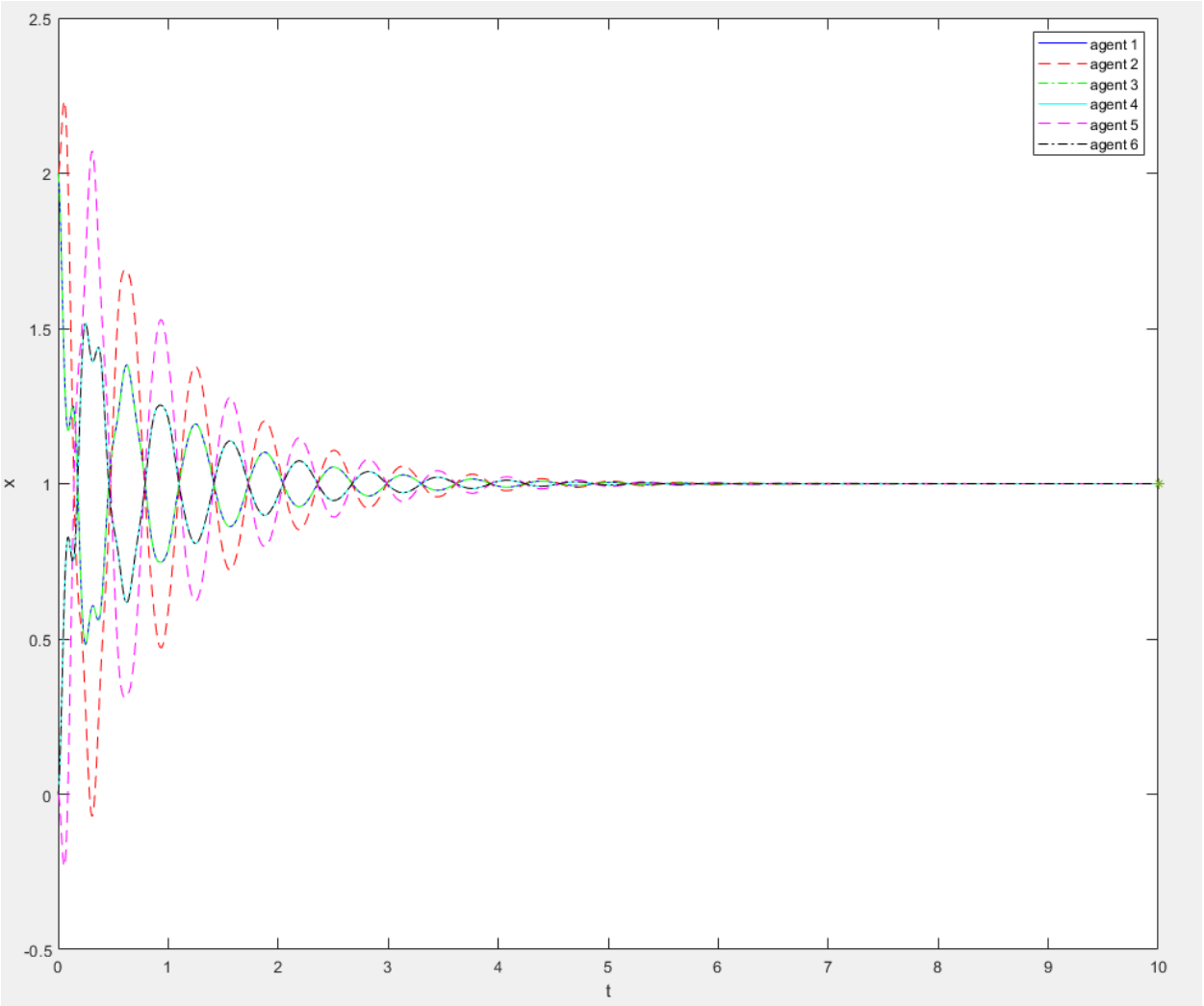

Figure 10: Virtual network-based consensus protocol when under attack

Under attack the network works almost the same and still reaches consensus.

Sources

(1) Homework 11, M. Iqbal, A. Gusrialdi, April 25, 2023
https://moodle.tuni.fi/pluginfile.php/3235043/mod_assign/introattachment/0/HW11.pdf
(2) Lecture 12 slides, A. Gusrialdi, April 25, 2023.
https://moodle.tuni.fi/pluginfile.php/3231245/mod_resource/content/3/AUT360_lecture12.pdf
(3) Exercise 12, M.W.S Atman, April 25, 2023
https://moodle.tuni.fi/pluginfile.php/3217031/mod_resource/content/1/Exercise_session_11.pdf