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

Homework 4

Valtteri Nikkanen 282688

Riku Pekkarinen 267084

Table of contents

Problem 1

We are given a digraph presented in figure 1. [1] We know that the continuous-time dynamics of this systems agents are:

$$\dot{x} = Lx \tag{1}$$

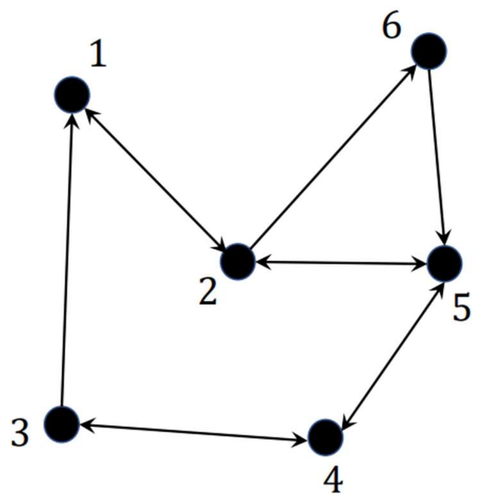

Figure 1: Strongly connected digraph.

We need to calculate the Laplacian matrix of the digraph in figure 1. This matrix is presented in table 1.

2	-1	-1	0	0	0
-1	2	0	0	-1	0
0	0	1	-1	0	0
0	0	-1	2	-1	0
0	-1	0	-1	3	-1
0	-1	0	0	0	1

Table 1: Laplacian matrix of the digraph in figure 1

The discrete-time variant of (1) is:

$$x(k+1) = x(k) - \epsilon L x(k), \tag{2}$$

where ϵ is a step-size.

We need to choose step-size so that (2) converges to consensus value and plot state trajectories of their initial conditions.

For a discrete-time system to reach a consensus value we can formulate a matrix P which is equal to:

$$P = I - \epsilon L \tag{3}$$

With this we can present the discrete-time dynamics of the system with the new matrix P as follows:

$$x(k+1) = Px(k) \tag{4}$$

A square matrix P is semi-convergent and not convergent if and only if:

i) 1 is eigenvalue,

ii) the algebraic multiplicity and geometric multiplicity of the eigenvalue 1 are equal.

which are always satisfied with any step size, and

iii) all other eigenvalues have magnitude less than 1,

which we will force by choosing step-size to be:

$$\epsilon \in (0, \frac{1}{\Delta}) \tag{5}$$

where Δ is $\max_i(\sum_{j \neq i} a_{ij})$, networks largest in-degree. From this range of (0, $\frac{1}{3}$), we chose 0.3 as our step-size.

With our step-size chosen we can formulate our matrix P. Matrix P was calculated with equation 3. The resulting matrix P is presented in table 2.

0.4	0.3	0.3	0	0	0
0.3	0.4	0	0	0.3	0
0	0	0.7	0.3	0	0
0	0	0.3	0.4	0.3	0
0	0.3	0	0.3	0.1	0.3
0	0.3	0	0	0	0.7

Table 2: Matrix P with step-size of 0.3 for digraph in figure 1

Now we can plot the trajectories of the agents in the discrete-time system. This is shown in figure 2. Random starting positions were used for the agents in the plotting. This simulation showed that the system is converging to the consensus value of about 0.51 with the initial condition in the figure.

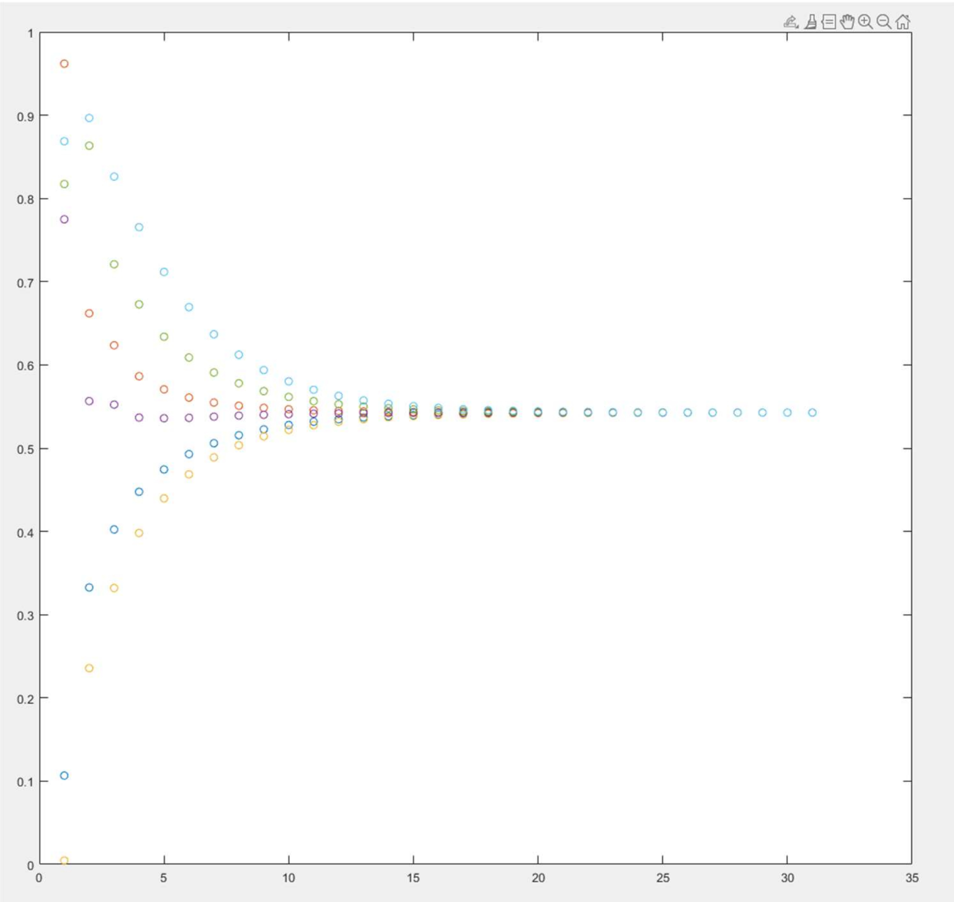

Figure 2: Plotted trajectories for the discrete-time system

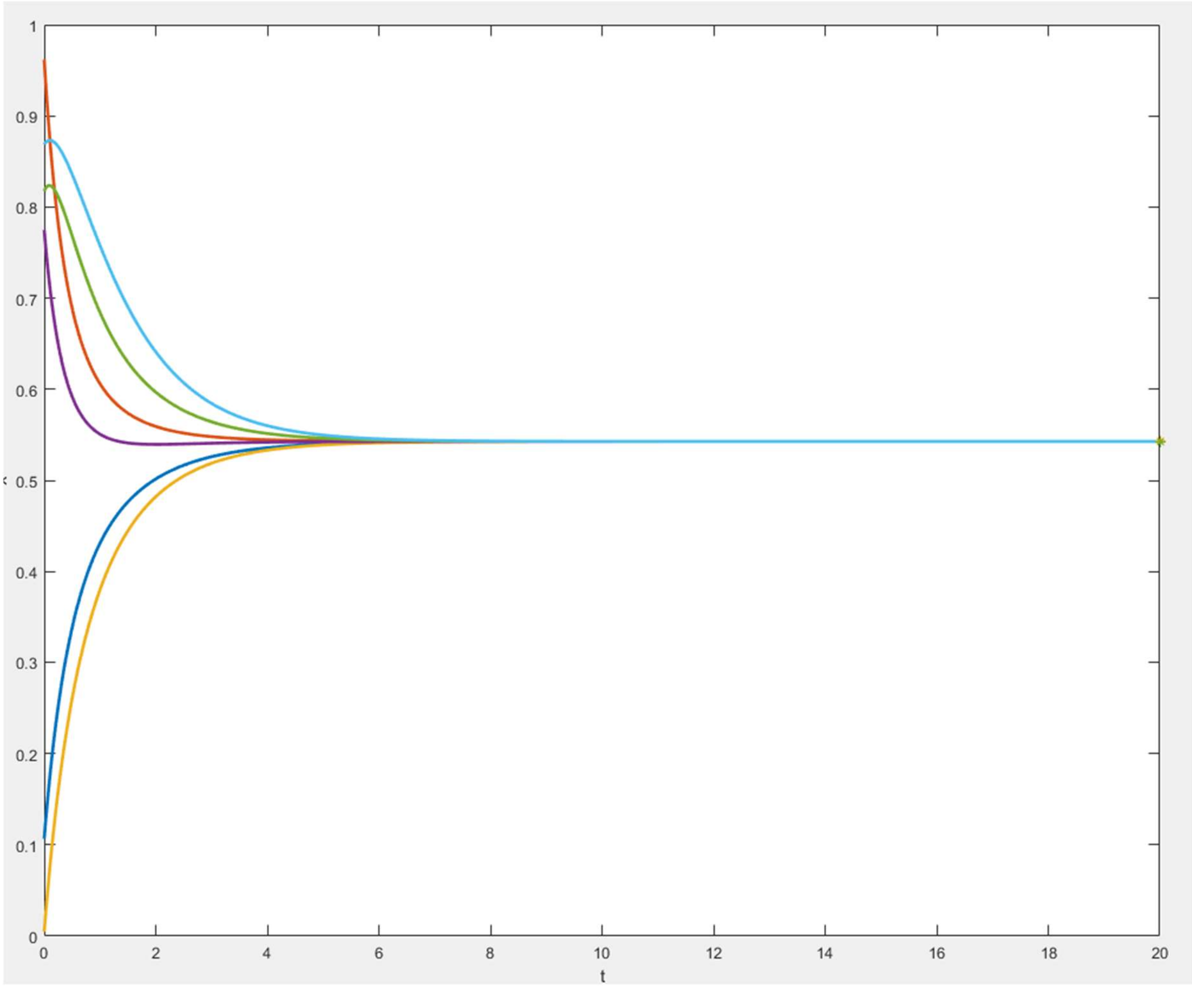

Figure 3: Same initial conditions for a continuous-time system

From the figure 2 and 3 we can see that the consensus value of discrete-time dynamics (4) is the same as continuous dynamics (1). [2]

Problem 2

Six agents in a certain area measuring temperature of the surrounding. Communication between the agent is directed so that if Agent I sends to Agent j, Agent j doesn't need to send information back. We will consider that the agents have the dynamics given in (1) and so also (2).

To have a network that converges to the average of the agents we need a balanced graph that is also strongly connected. For example, a simple cycle digraph should suffice for our needs. This network for our sensors is presented in figure 4.

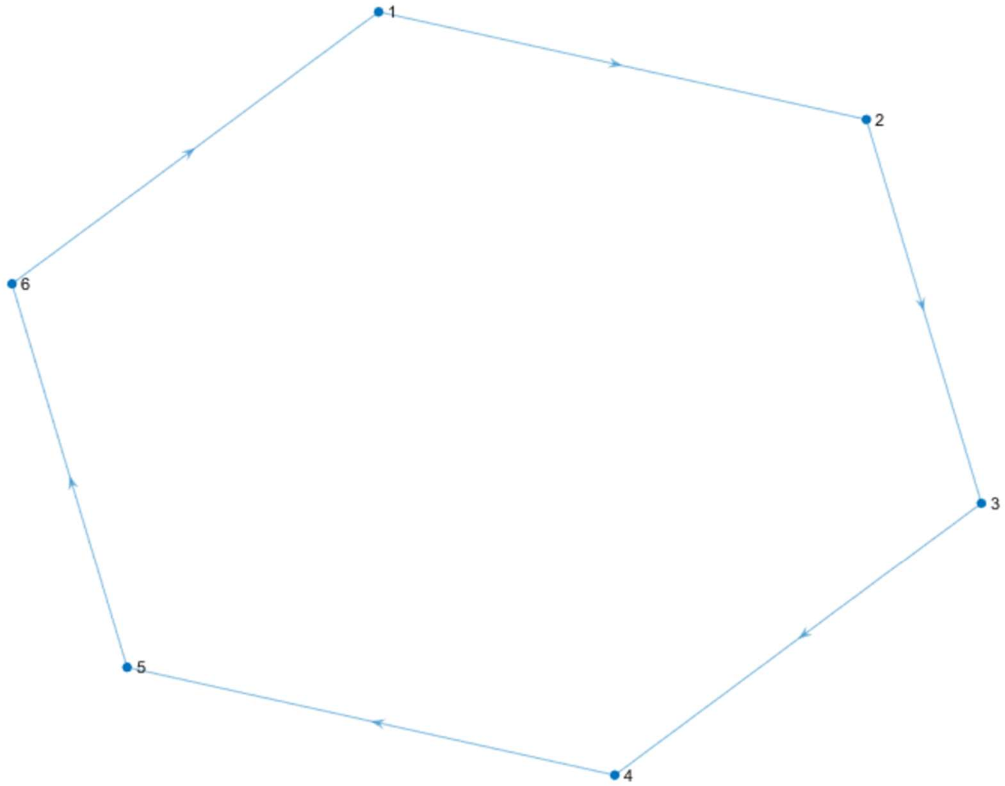

Figure 4: Digraph for our 6-agent sensor network.

We need to determine a good step-size for the discrete-time system. For this we need to create the Laplacian matrix and the P matrix, these are shown in tables 3 and 4 accordingly. In the adjacency matrix we have only ones in each of the matrix's rows, so the step-size of the system according to (5) needs to be between 0 and 1. We chose 0.9 to be our step-size.

1	0	0	0	0	-1
-1	1	0	0	0	0
0	-1	1	0	0	0
0	0	-1	1	0	0
0	0	0	-1	1	0
0	0	0	0	-1	1

Table 3: Laplacian matrix for Figure 3.

0.1	0	0	0	0	0.9
0.9	0.1	0	0	0	0
0	0.9	0.1	0	0	0
0	0	0.9	01	0	0
0	0	0	0.9	0.1	0
0	0	0	0	0.9	0.1

Table 4: P matrix with a steps-size of 0.9 for Figure 3.

We will simulate the network by plotting the trajectories of the agents with random initial conditions as in the problem 1. This trajectory plot simulation is shown in figure 5.

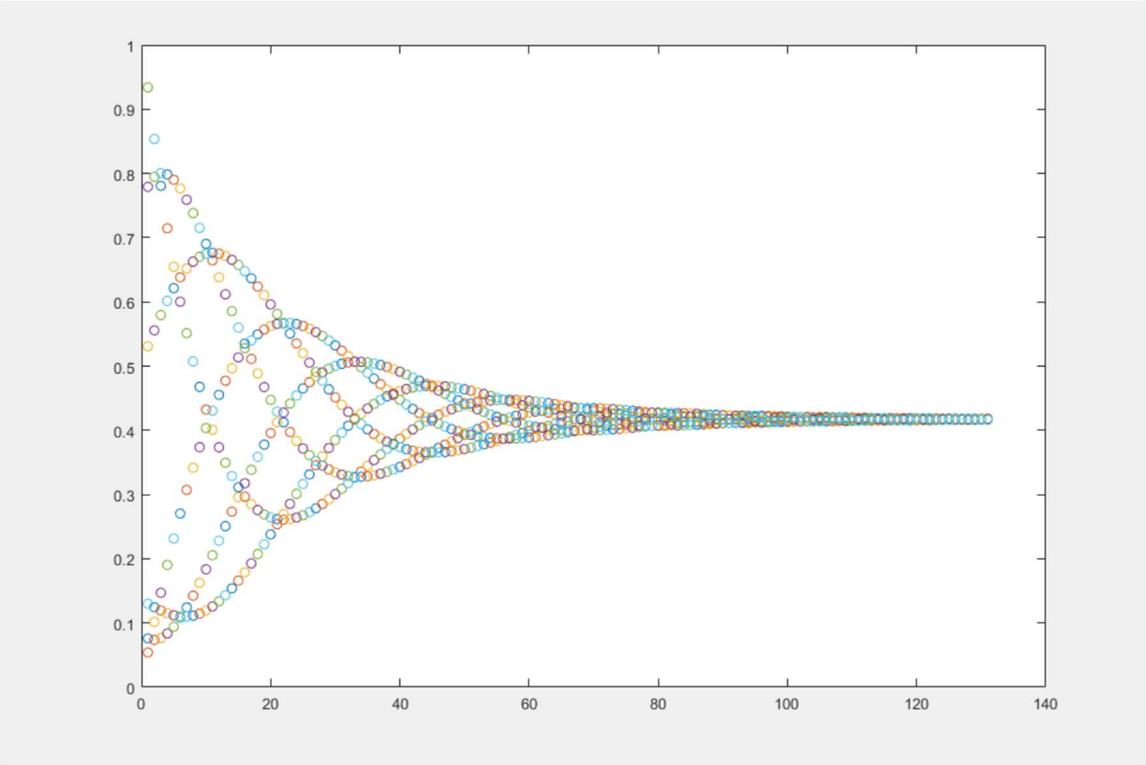

Figure 5: Simulation of the temperature sensor network with step size 0.9.

This network is simple and works but is the convergence is slow, adding communications links between agents would speed the process. But the network presented in figure 4 accomplishes the requirements and is extremely simple.

Problem 3

Considering a network of three mobile robots which are required to form a line formation between two points $p_{b1} = (1, 0)$ and $p_{b2} = (5, 0)$.

Considering that the nodes 1 and 5 are the two boundary points p_{b1} and p_{b2} the network digraph is presented in figure 6. In the digraph boundary point are nodes 1 and 5 and node 2 is agent 1, node 3 is agent 2 and node 4 is agent 3.

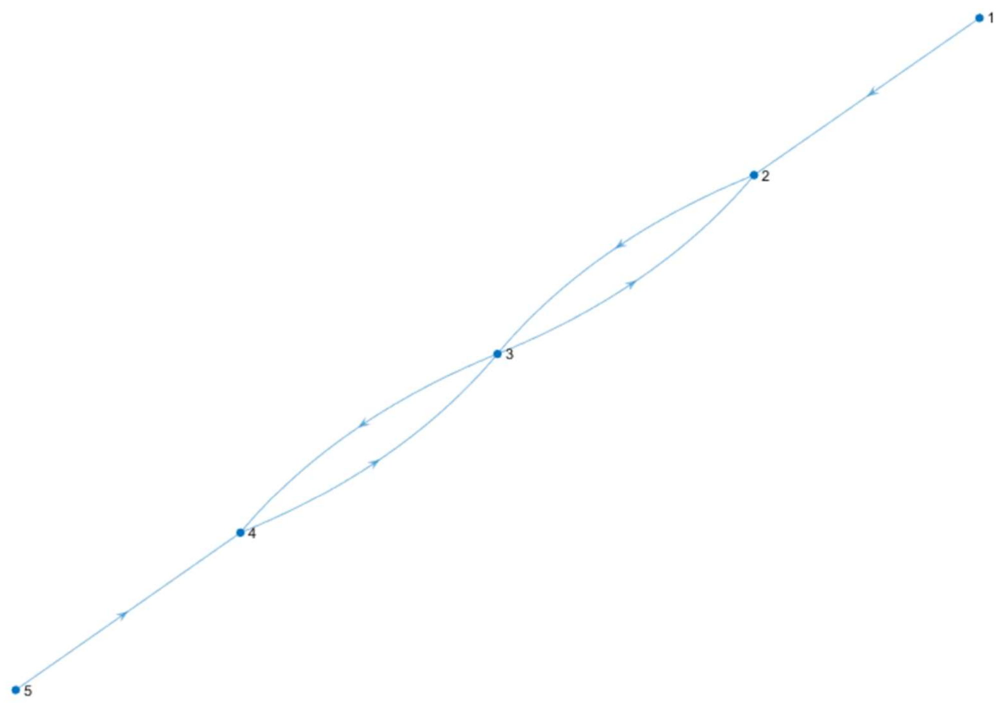

Figure 6: Digraph of the mobile robot network

Determining a step-size for a discrete-time system we need to know largest in-degree of the adjacency matrix, which will be 2, so our step-size will be between $(0, \frac{1}{2})$ and we chose 0.4 as our step-size. Here we have our Laplacian and P matrices.

0	0	0	0	0
-1	2	-1	0	0
0	-1	2	-1	0
0	0	-1	2	-1
0	0	0	0	0

Table 5: Laplacian of figure 6

1	0	0	0	0
0.4	0.2	0.4	0	0
0	0.4	0.2	0.4	0
0	0	0.4	0.2	0.4
0	0	0	0	1

Table 6: P matrix with a step-size of 0.4 for Figure 6.

For the y-coordinate of the robots we can use the same dynamics as before (1) and (2) as all the robots need to achieve the same y-coordinate with the boundary points to be in line with them. These same dynamics also work for the x-coordinate. So, we can use the same dynamics from (1) and (2) for the mobile robots in this network.

As Agent 1 converges towards boundary1 and Agent 2, and Agent towards boundary 2 and Agent 2, as Agent 2 works as a bridge between those agents they find averages between those points and line up well.

From random starting positions the robots converge in a line on the x- and y-coordinates as shown in figures 7 and 8.

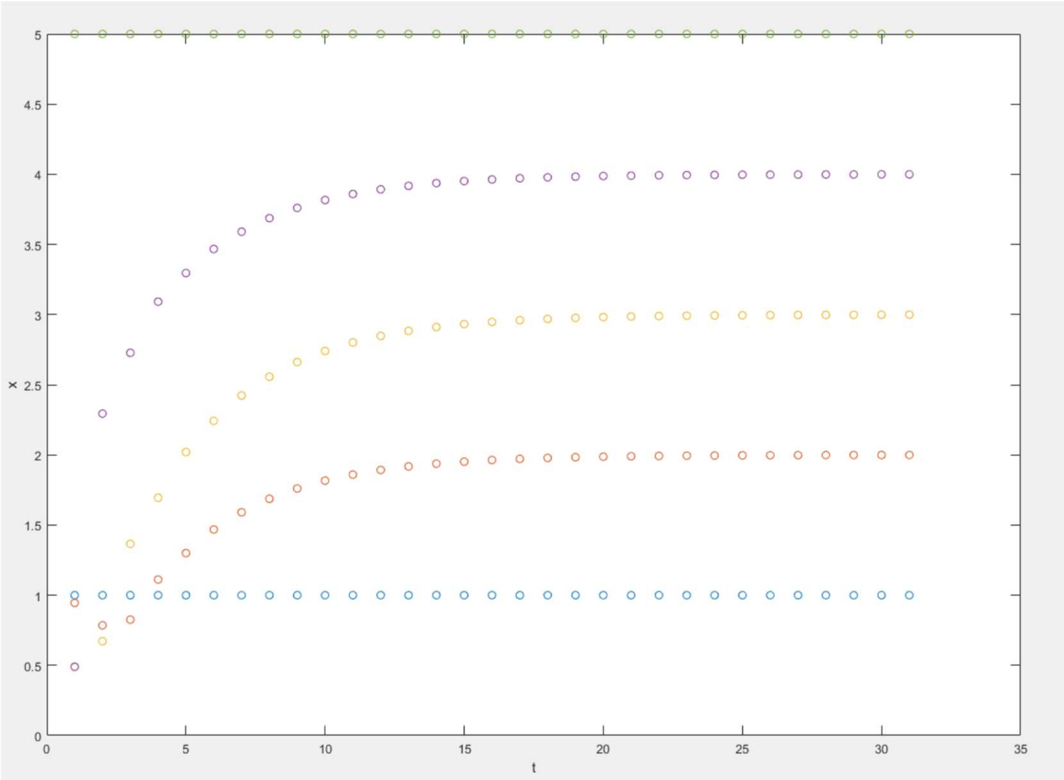

Figure 7: The x-coordinate trajectories of the mobile robots and the boundary points

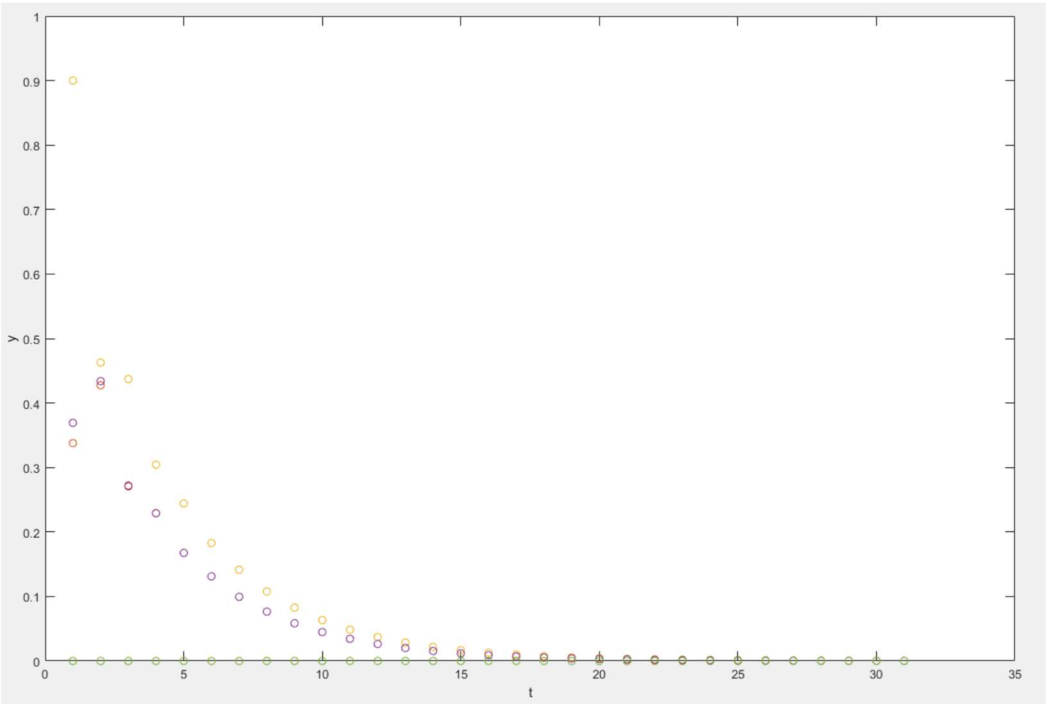

Figure 8: The y-coordinate trajectories of the mobile robots and the boundary points

Because we are unable to calculate eigenvectors for the graph, we will analytically demonstrate the positions of the robots using sensing/communication S-matrix in table 7 and its P-matrix in table 8, which is different from the P matrix calculated before using the step-size.

1	0	0	0	0
1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	0	0	1

Table 7: Sensing S matrix for figure 6.

1	0	0	0	0
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	0
0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0
0	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
0	0	0	0	1

Table 8: Sensing P matrix for figure 6.

From P matrix we can compute each nodes state:

$$x_1 = x_1$$

$$x_2 = \frac{x_1 + x_2 + x_3}{3}$$

$$x_3 = \frac{x_2 + x_3 + x_4}{3}$$

$$x_4 = \frac{x_3 + x_4 + x_5}{3}$$

$$x_5 = x_5$$

We can see that the local average for the nodes 1 and 5 are not changing as was supposed to. They are stable at the boundary points P_{b1} and P_{b2} in this $X_1 = P_{b1}$ and $X_5 = P_{b2}$. The moving robots' states are however changing as each of them computes the local average of its and its neighbours' states. Even from this we can deduce that as time moves forward the robots are going to move to the closest average point between the boundary points and each other that they can which is an equally spaced line between the boundary points.

Sources

(1) Homework 4, M. Iqbal, A. Gusrialdi, January 27, 2023. https://moodle.tuni.fi/pluginfile.php/3105374/mod_assign/introattachment/0/Homework4.pdf

(2) Lecture 4: Discrete-time consensus, A. Gusrialdi, February 4, 2023. https://moodle.tuni.fi/pluginfile.php/3099285/mod_resource/content/2/AUT360_lecture4.pdf