Distributed Control and Optimization of Cyber-Physical Systems

Exercise 10

by:

Dong Le (151057054)

Manikkuge Chathuranga Priyadarsana De Silva (150963990)

Date: 18.04.2023

Given Scenario :

Control input is $\dot{x}_i = u_i$ where $i \in \{1, \dots, 4\}$.

$x_i = [p_{x,i}, p_{y,i}]^T$ and $u_i = [u_{x,i}, u_{y,i}]^T$

Let's take ;

$x = [x_1{}^T \dots x_4{}^T]^T$, $u = [u_1{}^T \dots u_4{}^T]^T$ and $x_g = [u_{g,1}{}^T \dots u_{g,4}{}^T]^T$

Robots' initial position and the goal locations are;

$x(0) = [-0.6\ 0.6\ -0.2\ 0.4\ \ 0.2\ 0.5\ \ 0.6\ 0.6]^T\ m$

$x_g = [1.2\ 2.5\ -0.4\ 2.5\ \ 0.4\ 2.5\ \ -1.2\ 2.5]^T\ m$

Problem 1

$u = \arg\min\limits_{u_i} \sum_{i=1}^{4} ||u_{gtg,i} - u_j||^2$

s.t.

$$\left(\frac{\partial h_o(x_i,x_j)}{\partial x_i}\right)^T u_i + \left(\frac{\partial h_o(x_i,x_j)}{\partial x_i}\right)^T u_j \geq -2\gamma h_o^3(x_i, x_j) \quad (1)$$

$\gamma = 10$ and $h_o(x_i, x_j) = ||x_i - x_j||^2 - 0.2^2$

Question 1

The quadratic programming :

$$u = \arg min \frac{1}{2} u^T Q u + c^T u$$

$$s.t. Hu \leq b \quad (2)$$

From formula (1), to transform it to the format of (2)

$$-\left(\frac{\partial h_o(x_i, x_j)}{\partial x_i}\right)^T u_i - \left(\frac{\partial h_o(x_i, x_j)}{\partial x_i}\right)^T u_j \leq 2\gamma h_o^3(x_i, x_j)$$

Therefore, H is a matrix 6 by 2, and b is 6 by 1

$$Q = 2 * I_8 = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$c = -2 * u_{gtg}$ where $u_{gtg} = [u_{gtg,1}{}^T \ldots\ldots u_{gtg,4}{}^T]^T$

$$c = -2 * \begin{bmatrix} u_{gtg,1x} \\ u_{gtg,1y} \\ u_{gtg,2x} \\ u_{gtg,2y} \\ u_{gtg,3x} \\ u_{gtg,3y} \\ u_{gtg,4x} \\ u_{gtg,4y} \end{bmatrix} = \begin{bmatrix} -2 * (x_{goal1x} - x_{1x}) \\ -2 * (x_{goal1y} - x_{1y}) \\ -2 * (x_{goal2x} - x_{2x}) \\ -2 * (x_{goal2} \quad - x_{2y}) \\ -2 * (x_{goal3x} - x_{3x}) \\ -2 * (x_{goal3y} - x_{3y}) \\ -2 * (x_{goal4x} - x_{4x}) \\ -2 * (x_{goal4y} - x_{4y}) \end{bmatrix}$$

$$H = -2 * \begin{bmatrix} x_{1x} - x_{2x} & x_{1y} - x_{2y} & x_{2x} - x_{1x} & x_{2y} - x_{1y} & 0 & 0 & 0 & 0 \\ x_{1x} - x_{3x} & x_{1y} - x_{3y} & 0 & 0 & x_{3x} - x_{1x} & x_{3y} - x_{1y} & 0 & 0 \\ x_{1x} - x_{4x} & x_{1y} - x_{4y} & 0 & 0 & 0 & 0 & x_{4x} - x_{1x} & x_{4y} - x_{1y} \\ 0 & 0 & x_{2x} - x_{3x} & x_{2y} - x_{3y} & x_{3x} - x_{2x} & x_{3y} - x_{2y} & 0 & 0 \\ 0 & 0 & x_{2x} - x_{4x} & x_{2y} - x_{4y} & 0 & 0 & x_{4x} - x_{2x} & x_{4y} - x_{2y} \\ 0 & 0 & 0 & 0 & x_{3x} - x_{4x} & x_{3y} - x_{4y} & x_{4x} - x_{3x} & x_{4y} - x_{3y} \end{bmatrix}$$

$b = 2 * 10 * h^3(x_i - x_j)$

$$h(x_i - x_j) = \begin{bmatrix} (x_{1x} - x_{2x})^2 + (x_{1y} - x_{2y})^2 - 0.2^2 \\ (x_{1x} - x_{3x})^2 + (x_{1y} - x_{3y})^2 - 0.2^2 \\ (x_{1x} - x_{4x})^2 + (x_{1y} - x_{4y})^2 - 0.2^2 \\ (x_{2x} - x_{3x})^2 + (x_{2y} - x_{3y})^2 - 0.2^2 \\ (x_{2x} - x_{4x})^2 + (x_{2y} - x_{4y})^2 - 0.2^2 \\ (x_{3x} - x_{4x})^2 + (x_{3y} - x_{4y})^2 - 0.2^2 \end{bmatrix}$$

So,

$$b = 20 * \begin{bmatrix} ((x_{1x} - x_{2x})^2 + (x_{1y} - x_{2y})^2 - 0.2^2)^3 \\ ((x_{1x} - x_{3x})^2 + (x_{1y} - x_{3y})^2 - 0.2^2)^3 \\ ((x_{1x} - x_{4x})^2 + (x_{1y} - x_{4y})^2 - 0.2^2)^3 \\ ((x_{2x} - x_{3x})^2 + (x_{2y} - x_{3y})^2 - 0.2^2)^3 \\ ((x_{2x} - x_{4x})^2 + (x_{2y} - x_{4y})^2 - 0.2^2)^3 \\ ((x_{3x} - x_{4x})^2 + (x_{3y} - x_{4y})^2 - 0.2^2)^3 \end{bmatrix}$$

Question 2

Figure 1 shows the trajectory of four agents, they starts from their initial positions, and reach the goals.

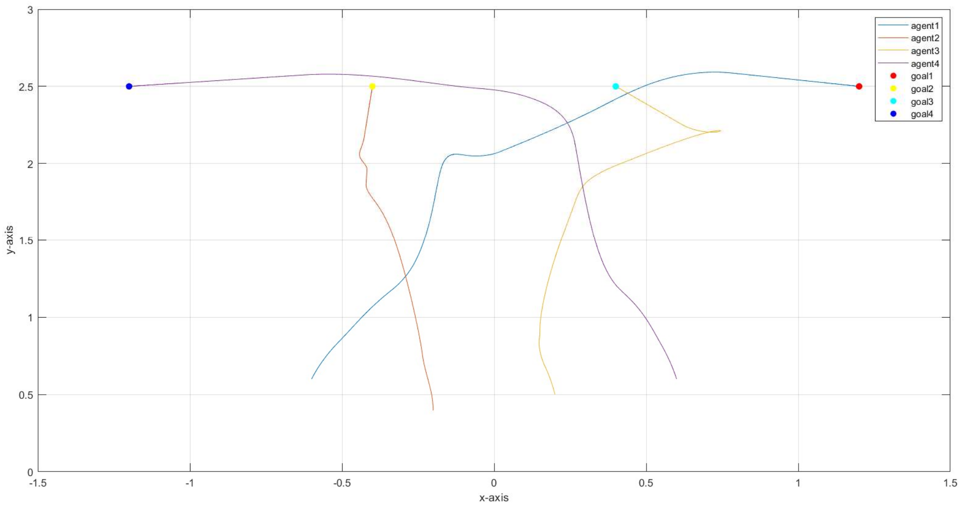

Figure 1 : Prob1 - XY tracjectory

Figure 2 displays the error between x_goal and x_agent, so after 6.5 sec, all four agents reach their goals so the errors went to 0.

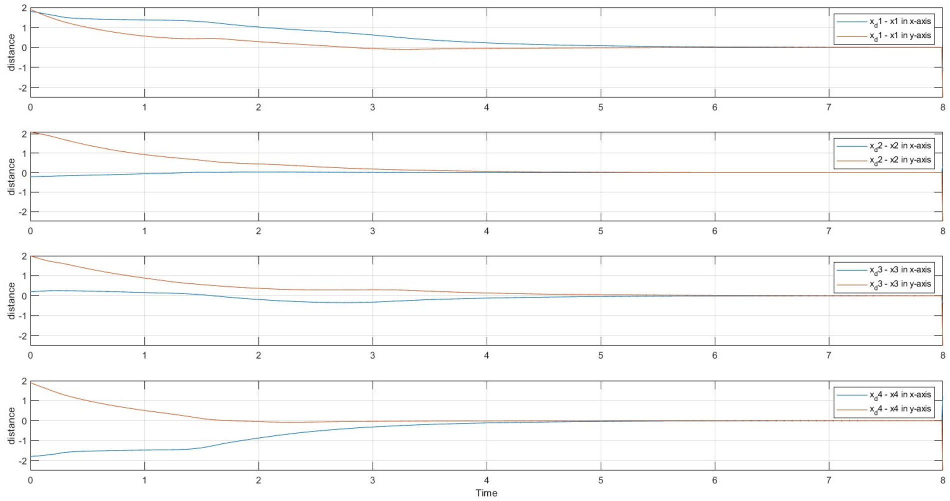

Figure 2 : Prob1 - Time Series of (Xd – X)

To ensure that the controller did the right job, figure 3 shows the distance between each agents, so if any distance goes below 0.2, which means collision. However, based on the plots below, none of the distance went to 0.2, so all robots towards the goals without colliding with each other.

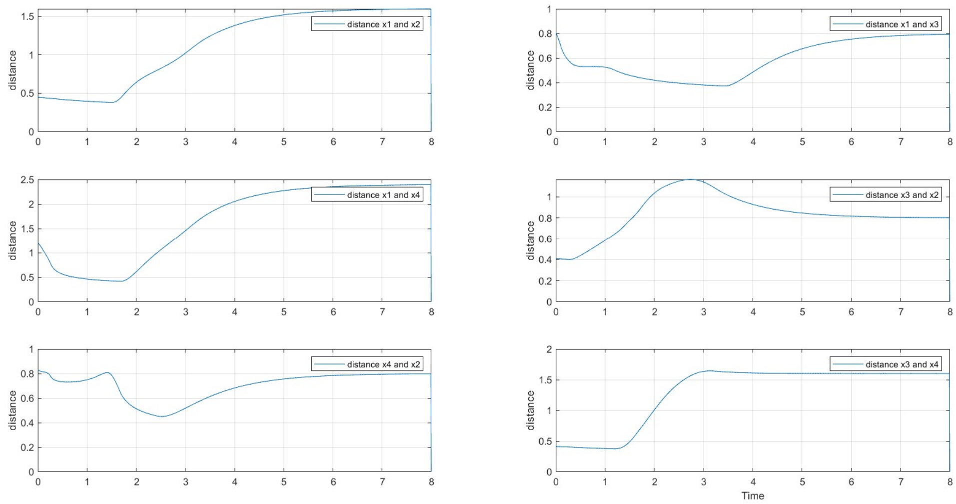

Figure 3 : Prob1 - Time Series of the distance between all pairs of robots

Figure 4 shows the control inputs of each agents, after 6 sec, they all went to 0, meaning the robot stop at its destination.

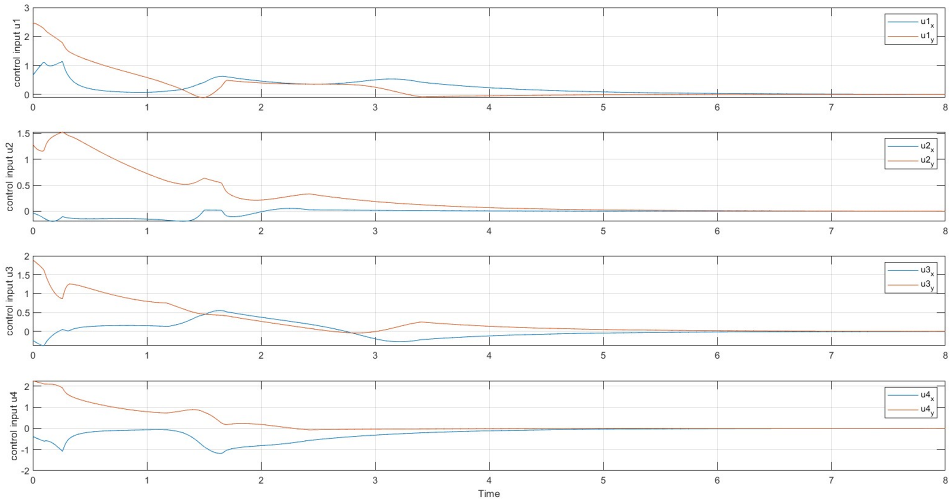

Figure 4 : Prob1 - Time Series of control inputs

Finally, figure 5 shows the time series of goal and x. As the result showed in above graphs, the robots reach the goals, so at the end, the goal lines and x lines merge into one line.

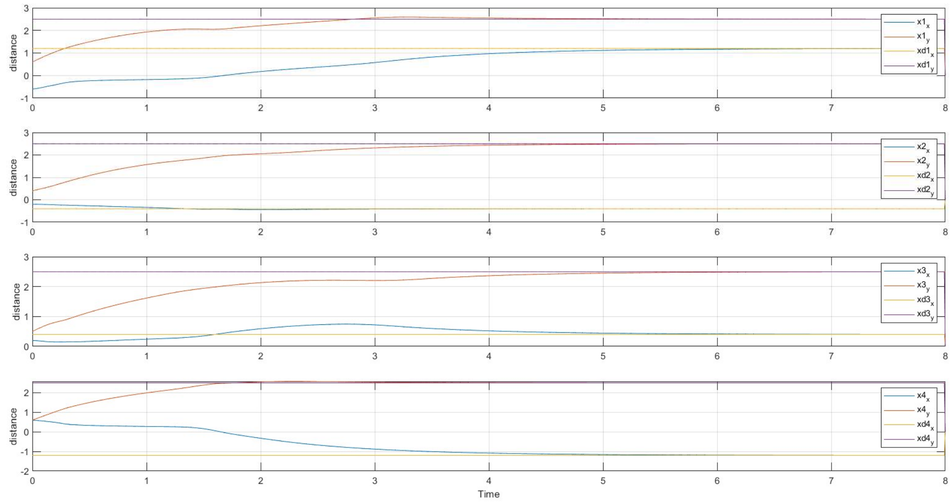

Figure 5 : Prob1 - Time Series of goal vs x

Problem 2

$u_i = \arg\min_{u_i} ||u_{gtg,i} - u_i||^2$

s.t.

$$\left(\frac{\partial h_o(x_i,x_j)}{\partial x_i}\right)^T u_i \geq -\gamma h_o^3(x_i, x_j), \qquad \forall j \in N_i^s \tag{3}$$

with $N_i^s = \{j, j \neq i \mid ||u_i - u_j||^2 \leq 1^2\}$

Question 1

The quadratic programming :

$$u_i = \arg min \frac{1}{2} u_i{}^T Q_i u_i + c_i^T u_i$$

$$s.t. H_i u_i \leq b_i \tag{4}$$

From fomular (3), to transform it to the format of (4)

$$-\left(\frac{\partial h_o(x_i, x_j)}{\partial x_i}\right)^T u_i \leq \gamma h_o^3(x_i, x_j)$$

Therefore, H is a matrix 3 by 2, and b is 3 by 1

Assume $N_i^s = \{2, 3, 4\}$

$$Q_1 = 2 * I_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$c_1 = -2 * u_{gtg1} = \begin{bmatrix} -2 * (x_{goal1x} - x_{1x}) \\ -2 * (x_{goal1y} - x_{1y}) \end{bmatrix}$$

$$H_1 = -2 * \begin{bmatrix} x_{1x} - x_{2x} & x_{1y} - x_{2y} \\ x_{1x} - x_{3x} & x_{1y} - x_{3y} \\ x_{1x} - x_{4x} & x_{1y} - x_{4y} \end{bmatrix}$$

$b_1 = 10 * h_1{}^3(x_i - x_j)$

$$h_1(x_i - x_j) = \begin{bmatrix} (x_{1x} - x_{2x})^2 + (x_{1y} - x_{2y})^2 - 0.2^2 \\ (x_{1x} - x_{3x})^2 + (x_{1y} - x_{3y})^2 - 0.2^2 \\ (x_{1x} - x_{4x})^2 + (x_{1y} - x_{4y})^2 - 0.2^2 \end{bmatrix}$$

So

$$b_1 = 10 * \begin{bmatrix} ((x_{1x} - x_{2x})^2 + (x_{1y} - x_{2y})^2 - 0.2^2)^3 \\ ((x_{1x} - x_{3x})^2 + (x_{1y} - x_{3y})^2 - 0.2^2)^3 \\ ((x_{1x} - x_{4x})^2 + (x_{1y} - x_{4y})^2 - 0.2^2)^3 \end{bmatrix}$$

Question 2

Figure 6 shows the trajectory of four agents, they starts from their initial positions, and reach the goals.

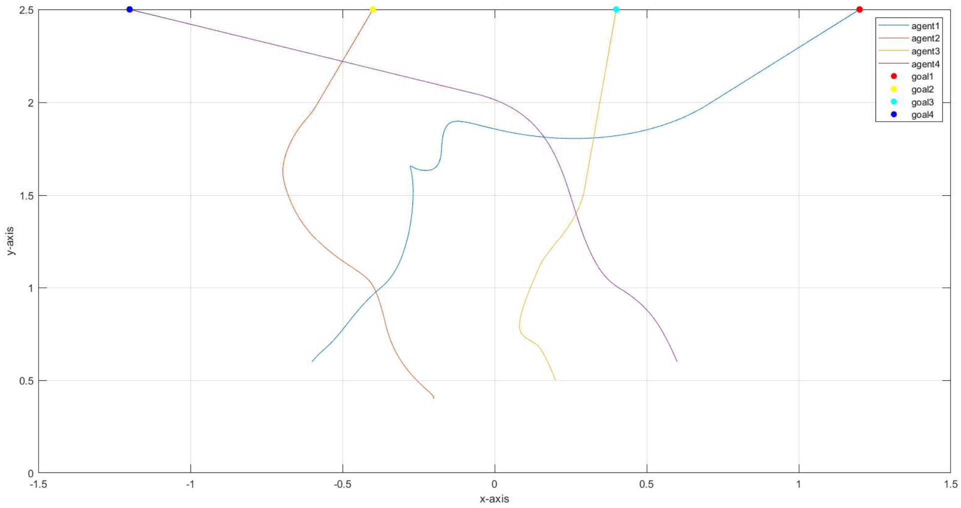

Figure 6 : Prob2 - XY trajectory

Figure 7 displays the error between x_goal and x_agent, so after 6 sec, all four agents reach their goals so the errors went to 0.

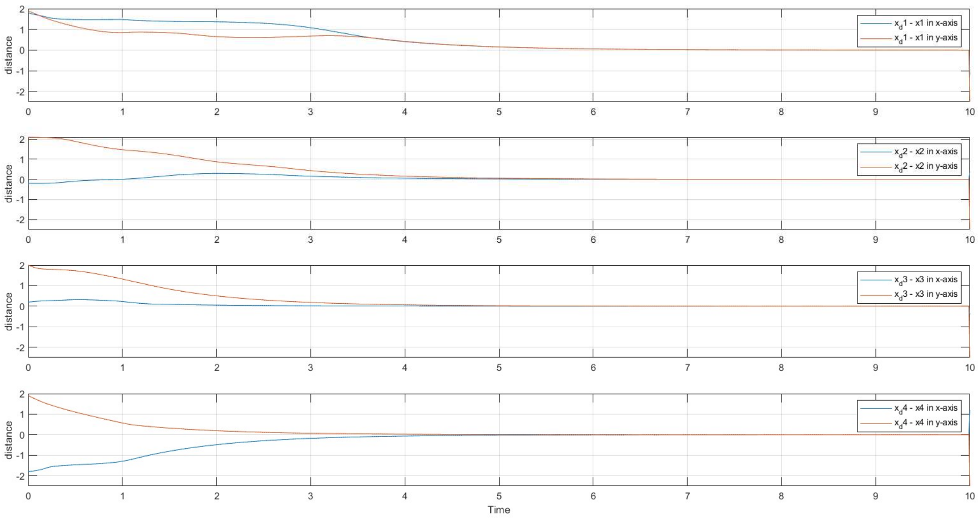

Figure 7: Prob2 - Time Series of (Xd – X)

To ensure that the controller did the right job, figure 8 shows the distance between each agents, so if any distance goes below 0.2, which means collision. However, based on the plots below, none of the distance went to 0.2, so all robots towards the goals without colliding with each other.

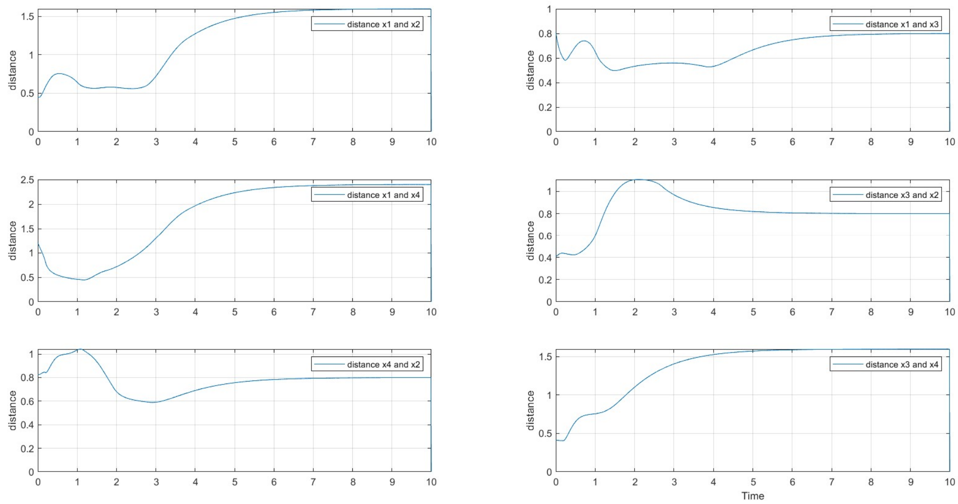

Figure 8: Prob2 - Time Series of the distance between all pairs of robots

Figure 9 shows the control inputs of each agents, after 6 sec, they all went to 0, meaning the robot stop at its destination.

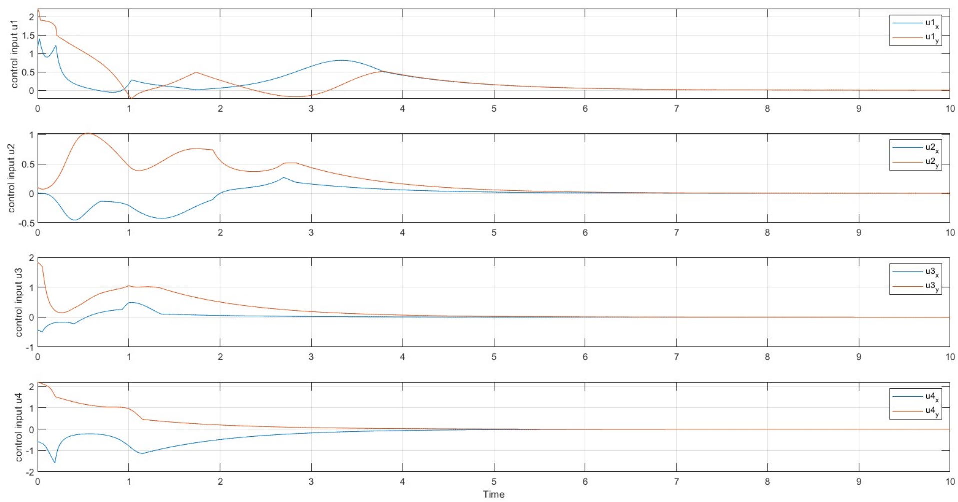

Figure 9: Prob2 - Time Series of control inputs

Finally, figure 10 shows the time series of goal and x. As the result showed in above graphs, the robots reach the goals, so at the end, the goal lines and x lines merge into one line.

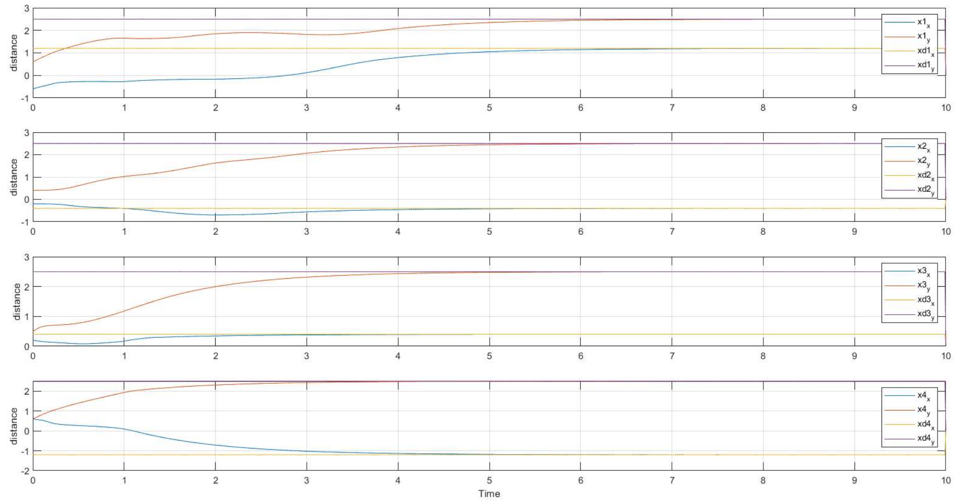

Figure 10: Prob2 - Time Series of goal vs x

Question 3

Compare the simulation result:

Both approaches' controller handles the problem well, all agents reach the goal. However, looking at the Time Series of the distance between all pairs of robots of figure 3 and figure 8, the agents in the distributed method keep the distance further than in the centralized method.

Advantages of Centralized Approach:

- Better coordination: In centralized approach, central controller has the global view of the system, which can lead to better coordination between the robots. Therefore, robots' movement is smoother and efficient.
- Resources are efficiently used: Use of resources (for example, battery) can be optimized by taking into account the state of all the robots.
- Easier to implement: A centralized approach is comparatively easier to implement than a distributed approach.

Disadvantages of Centralized Approach:

- Robustness issue: A centralized approach is prone to a single point of failure. All the robots will be affected if the controller fails.
- Scalability: Centralized approach is not good in terms of scalability. When the number of robots increases, the controller become much more complicated and will eventually fail.
- Communication difficulties: A centralized approach requires communication between the controller and all the robots. This can create communication issues, which can slow down the system.

Advantages of Distributed Approach:

- Robustness: Distributed approach can be more robust than a centralized approach, as each robot has its own controller. If one controller fails, the other robots can continue to function.
- Scalability: distributed approach is more scalable as each robot is a one equal building block in a system. This allows for more robots to be added to the system without affecting the performance of the other robots and the whole system.
- Communication is simple: In distributed approach communication is simple, as each robot only needs to communicate only with neighbors.

Disadvantages of Distributed Approach:

- Lack of coordination: distributed approach can lead to a lack of coordination between the robots, as each robot only has access to local information. This can result in inefficient movements of the robots.
- Inefficient use of resources: distributed approach may not optimize the use of resources as each robot is only aware of its own state and probably some information about its neighbors. This can result in inefficient use of energy and other resources.
- Implementation difficulty: Sometimes, distributed approach can be more complex than a centralized approach, as each robot has its own controller that needs to be designed and implemented.

Problem 3

A new constraint is added which is to maintain the network connectivity. Therefore, the optimization problem is as below.

$u_i = \arg\min_{u_i} ||u_{gtg,i} - u_i||^2$

s.t.

$$\left(\frac{\partial h_o(x_i,x_j)}{\partial x_i}\right)^T u_i \ \geq \ -\gamma h_o^3(x_i, x_j), \qquad \forall j \ \in N_i^s \tag{5}$$

$$\left(\frac{\partial h_o(x_i,x_j)}{\partial x_i}\right)^T u_i \ \geq \ -\gamma h_c^3(x_i, x_j), \qquad \forall j \ \in N_i^c \tag{6}$$

With new h function $0.9^2 - \ ||x_i - x_j||^2$

Question 1

To transforme 5 and 6 into QP distributed format 4

$$-\left(\frac{\partial h_o(x_i,x_j)}{\partial x_i}\right)^T u_i \leq \gamma h_o^3(x_i, x_j)$$

$$-\left(\frac{\partial h_o(x_i,x_j)}{\partial x_i}\right)^T u_i \leq \gamma h_c^3(x_i, x_j)$$

Assume $N_i^s = \{2, 3, 4\}$ and $N_i^c = \{2, 3, 4\}$, so agent 1 implements two controllers maintain the connectivity and avoid collision. Therefore, H_i is a matrix 6-by-2, and h_i is 6-by-1

$$Q_1 = 2 * I_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$c_1 = -2 * u_{gtg1} = \begin{bmatrix} -2 * (x_{goal1x} - x_{1x}) \\ -2 * (x_{goal1y} - x_{1y}) \end{bmatrix}$$

$$H_1 = \begin{bmatrix} -2 * (x_{1x} - x_{2x}) & -2 * (x_{1y} - x_{2y}) \\ -2 * (x_{1x} - x_{3x}) & -2 * (x_{1y} - x_{3y}) \\ -2 * (x_{1x} - x_{4x}) & -2 * (x_{1y} - x_{4y}) \\ 2 * (x_{1x} - x_{2x}) & 2 * (x_{1y} - x_{2y}) \\ 2 * (x_{1x} - x_{3x}) & 2 * (x_{1y} - x_{3y}) \\ 2 * (x_{1x} - x_{4x}) & 2 * (x_{1y} - x_{5y}) \end{bmatrix}$$

$b_1 = 10 * {h_1}^3(x_i - x_j)$

$$h_1(x_i - x_j) = \begin{bmatrix} (x_{1x} - x_{2x})^2 + (x_{1y} - x_{2y})^2 - 0.2^2 \\ (x_{1x} - x_{3x})^2 + (x_{1y} - x_{3y})^2 - 0.2^2 \\ (x_{1x} - x_{4x})^2 + (x_{1y} - x_{4y})^2 - 0.2^2 \\ 0.9^2 - ((x_{1x} - x_{2x})^2 + (x_{1y} - x_{2y})^2) \\ 0.9^2 - ((x_{1x} - x_{3x})^2 + (x_{1y} - x_{3y})^2) \\ 0.9^2 - ((x_{1x} - x_{4x})^2 + (x_{1y} - x_{4y})^2) \end{bmatrix}$$

So,

$$b_1 = 10 * \begin{bmatrix} ((x_{1x} - x_{2x})^2 + \left(x_{1y} - x_{2y}\right)^2 - 0.2^2)^3 \\ ((x_{1x} - x_{3x})^2 + (x_{1y} - x_{3y})^2 - 0.2^2)^3 \\ ((x_{1x} - x_{4x})^2 + (x_{1y} - x_{4y})^2 - 0.2^2)^3 \\ (0.9^2 - ((x_{1x} - x_{2x})^2 + \left(x_{1y} - x_{2y}\right)^2))^3 \\ (0.9^2 - ((x_{1x} - x_{3x})^2 + \left(x_{1y} - x_{3y}\right)^2))^3 \\ (0.9^2 - ((x_{1x} - x_{4x})^2 + \left(x_{1y} - x_{4y}\right)^2))^3 \end{bmatrix}$$

Question 2

The purpose of the controller is to ensure safety between robots, to remain the connectivity, and to reach the goals.

Each robot connected to all other robots, so figure 11 shows the graph of the system.

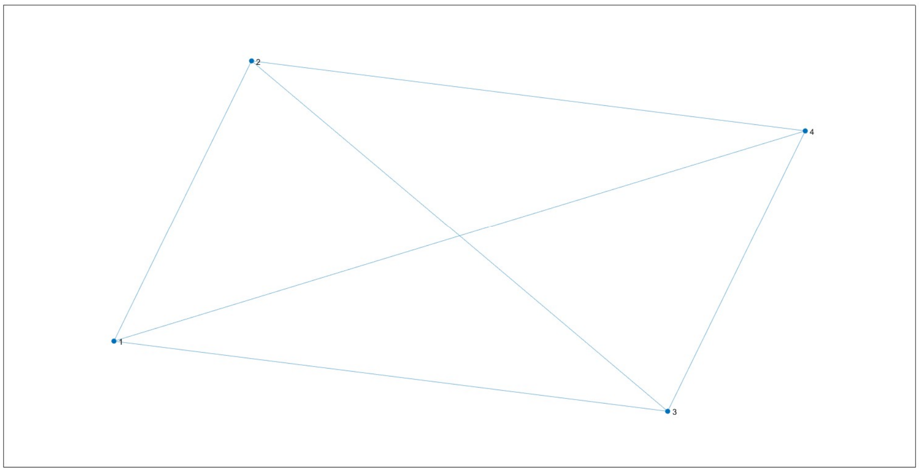

Figure 11: Connection between robots

After implement the controller, figure 12 displays the trajectory of the agents

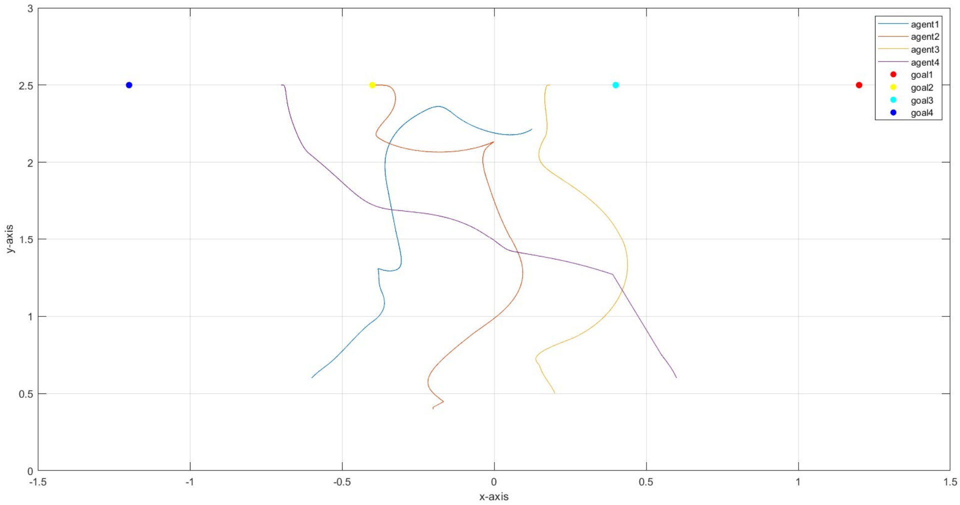

Figure 12 : Prob3_Q2 - XY trajectory

However, three agents did not reach the goals, only agent 2 did. To clarify the situation, let check another plot.

Figure 13 shows the error between goal and x, it only shows that agent 2 the error went to 0, meaning the goal was reached, the other agents, the error still larger than 0.

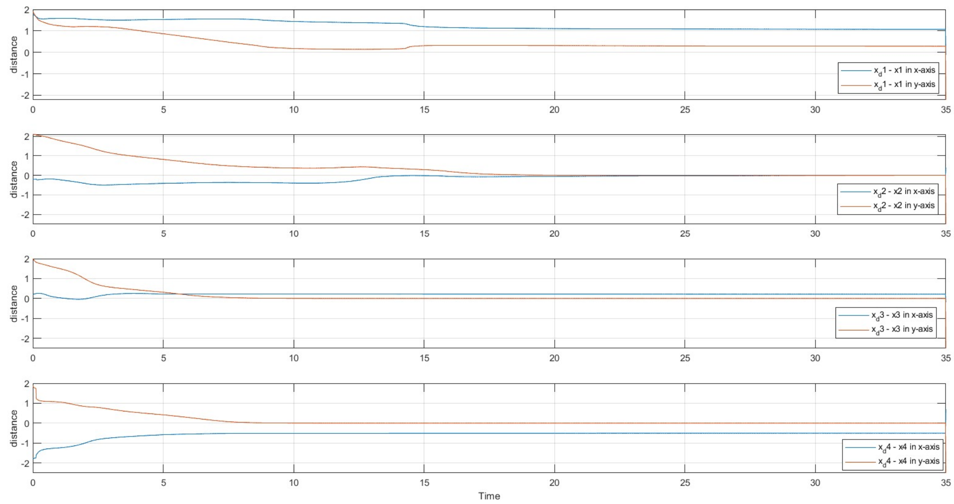

Figure 13 : Prob3_Q2 - Time Series of (Xd – X)

The good news is all distance between agents are larger than 0.2, menaing there was no collision between robots.

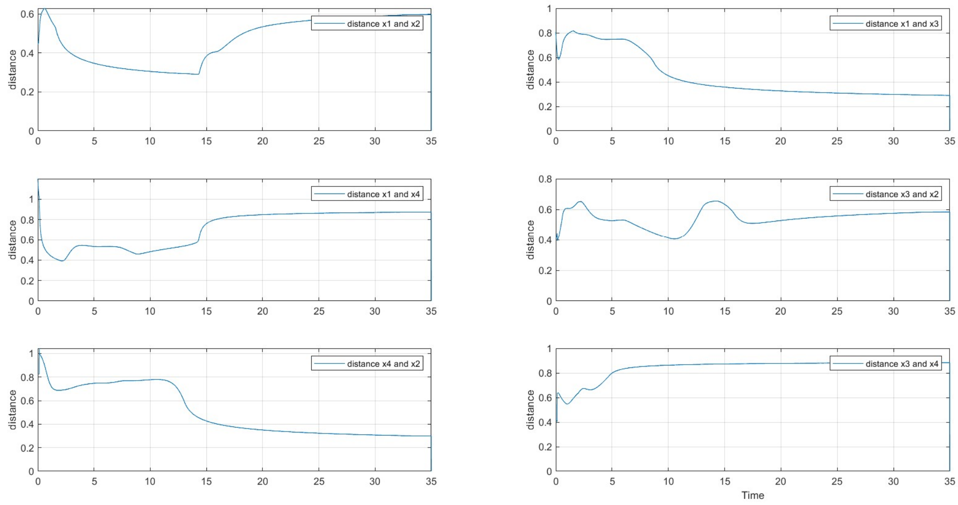

Figure 14 : Prob3_Q2 - Time Series of the distance between all pairs of robots

Figure 15 displays the control input of agents, after 25 sec, four agents stop moving even though most of it did not reach the goal. It happend because the agent 2 was at its goal, so due to the connectivity, the others have to be stopped.

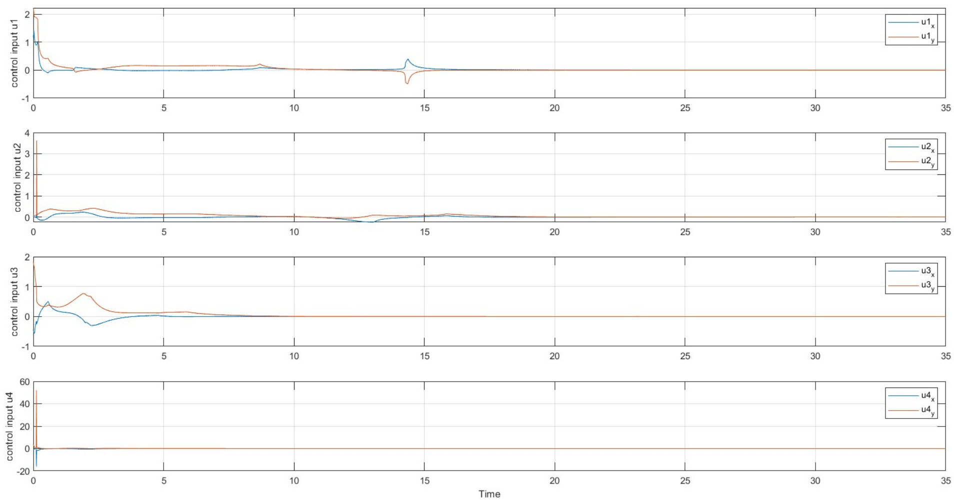

Figure 15 : Prob3_Q2 - Time Series of control inputs

To clarify the situation, figure 16 shows the time series between goal and x, except agent 2, none of the agents have the goal lines match the position line.

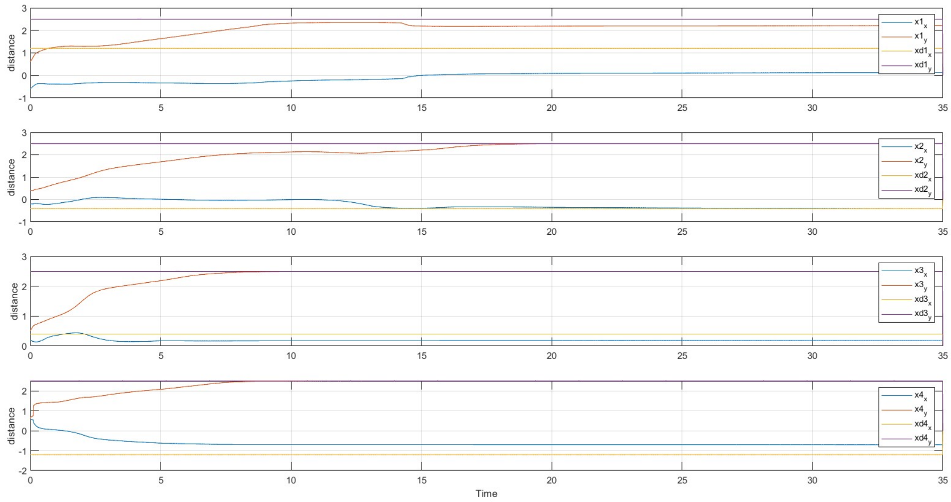

Figure 16 : Prob3_Q2 - Time Series of goal vs x

This situation occurs because of the connectivity remain $h_c(x_i, x_j) = 0.9^2 - \|x_i - x_j\|^2$ so the agents keep the furthest distance is 0.9.

However, as the goal of agents is $x_g = [1.2\ 2.5\ \ -0.4\ 2.5\ \ 0.4\ 2.5\ \ -1.2\ 2.5]^T$, distance between agents are much larger than 0.9, for example distance between agent 1 and agent 2 supports to be 1.6. So, when the agent 2 reaches the goal, all agents support to remain the connectivity, therefore it stops at the furthest it can go.

Question 3

To fix the situation, an approach is to change the topology of the system from fully connect to spanning tree. Figure 17 shows a new connectivity between agents

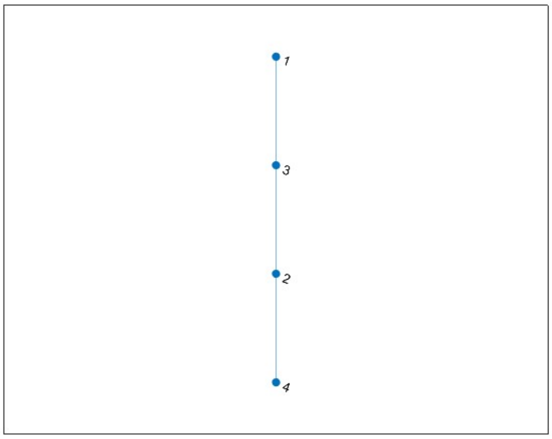

Figure 17: Spanning tree

So the Laplacian matrix of four agents is :

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

After implement a new controller, the result is showed in figure 18, the trajectory of four agents, and they did reach their goals.

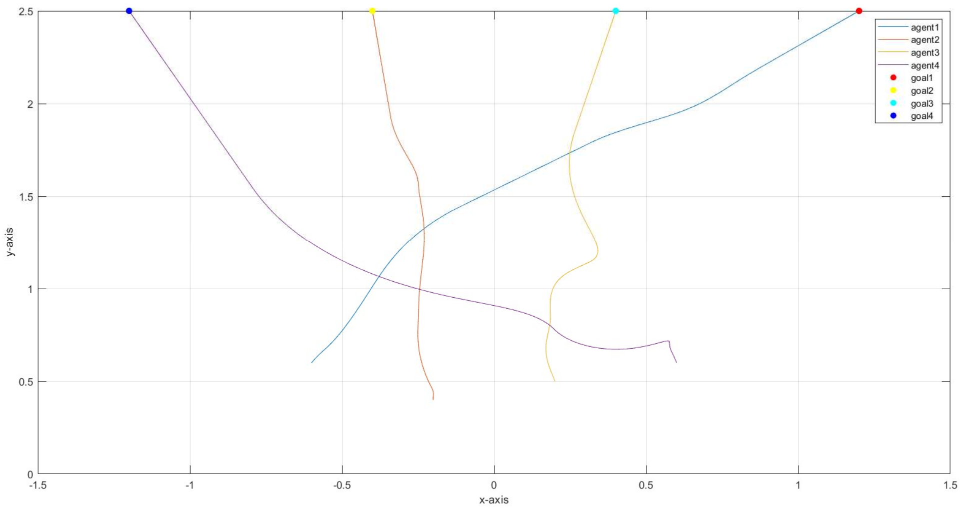

Figure 18 : Prob3_Q3 - XY trajectory

Figure 19 displays the error between goal and position of each robot, all error went to 0.

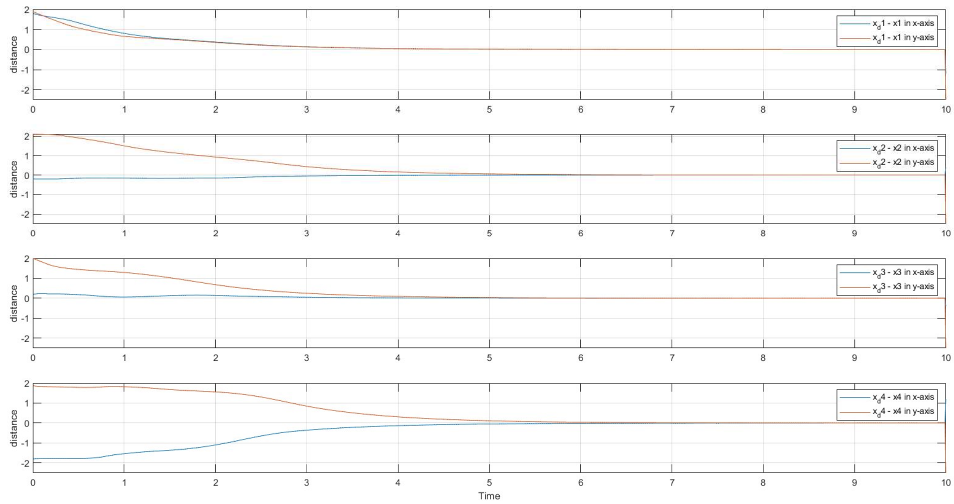

Figure 19 : Prob3_Q3 - Time Series of (Xd – X)

Figure 20 shows the distance between robots, due to the new spanning tree connectivity, the largest distance between two neighbor robots is not over 0.9. If two robots are not neighbor so they do not have to remain the distance 0.9, therefore they can reach the goals freely.

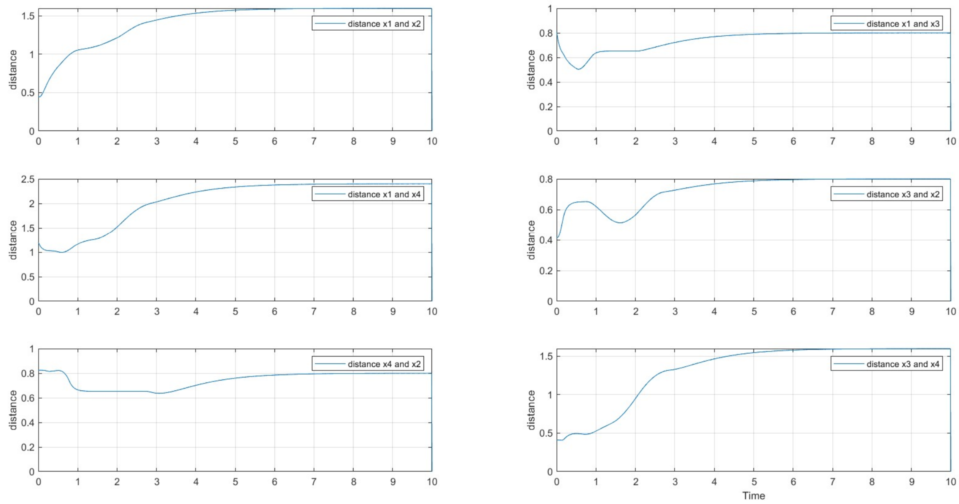

Figure 20 : Prob3_Q3 - Time Series of the distance between all pairs of robots

After about 6 sec, all control input went to 0 due to the fact that goals were reached, it is showed in figure 21.

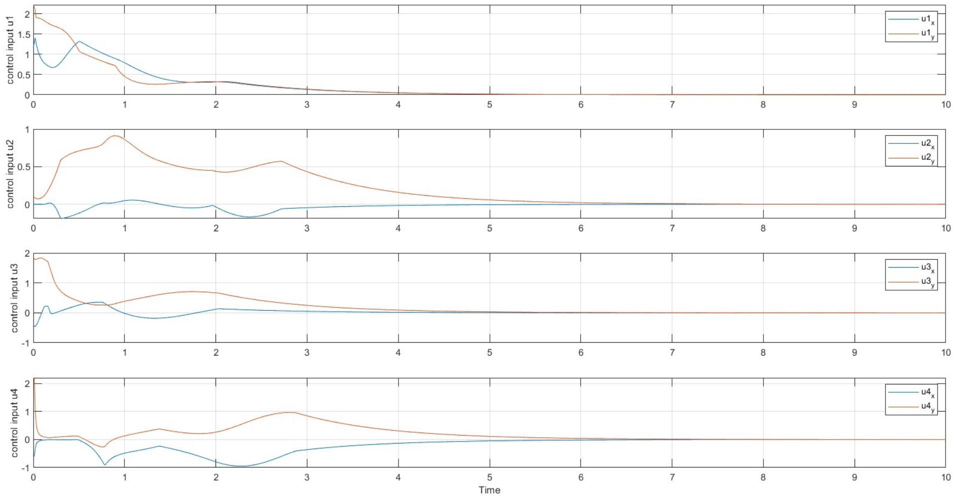

Figure 21 : Prob3_Q3 - Time Series of control inputs

Finally, figure 22 shows the time series of goal and x. As the result showed in above graphs, the robots reach the goals, so at the end, the goal lines and x lines merge into one line.

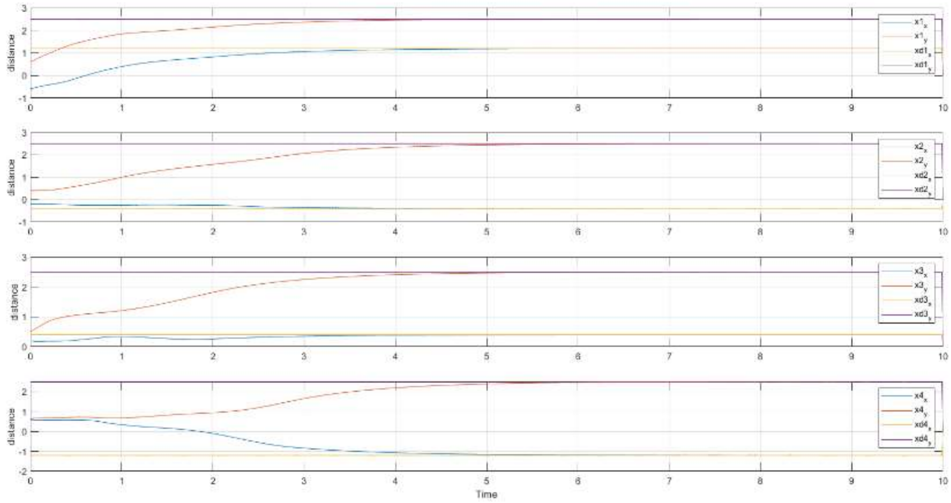

Figure 22 : Prob3_Q3 - Time Series of goal vs x