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

Homework 10

Valtteri Nikkanen 282688

Riku Pekkarinen 267084

Table of contents

We have four omnidirectional robots with the following dynamics:

$$\dot{x}_i = u_i,\ \ i \in \{1, \dots, 4\} \tag{1}$$

Where $x_i = [p_{x,i} \quad p_{y,i}]^T \in \mathbb{R}^2$ which represents the position of the robot and $u_i = [u_{x,i} \quad u_{y,i}]^T \in \mathbb{R}^2$ is the control input that is to be designed. Each robot is given an initial location $x_i(0)$ and a goal position $x_{g,i}$. And they need to move toward the goal position by following a go-to-goal controller $u_{gtg,i} = \left(x_{g,i} - x_i\right)$. Let us denote the combined state, control input and goal positions for all robots as:

$$x = [x_1^T \dots x_4^T]^T \in \mathbb{R}^8,\ u = [u_1^T \dots u_4^T]^T \in \mathbb{R}^8,\ x_g = \left[x_{g,1}^T \dots x_{g,4}^T\right]^T \in \mathbb{R}^8 \tag{2}$$

With the initial and goal positions for the robots are at

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

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

Problem 1

We need to create a centralized optimization-based safety controller as follows:

$$u = \underset{u_i, i=\{1,\dots,4\}}{\arg\min} \ \sum_{i=1}^{4} \left\| u_{gtg,i} - u_i \right\|^2$$

$$s.t. \left(\frac{\partial h_o(x_i,x_j)}{\partial x_i}\right)^T u_i + \left(\frac{\partial h_o(x_i,x_j)}{\partial x_j}\right)^T u_j \geq -2\gamma h_o^3\left(x_i, x_j\right),\ \forall j \neq i \tag{3}$$

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

Given that the general form of quadratic programming is

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

$$s.t. Hu \leq b \tag{4}$$

We need to compute Q, c, H and b from the given equation 3.

Let us start by calculating the Q and the c from equation for the control signal u.

$$u = \underset{u_i, i=\{1,\dots,4\}}{\arg\min} \ \sum_{i=1}^{4} \left\| u_{gtg,i} - u_i \right\|^2$$

$$u = \underset{u}{\arg\min} \left\| u_{gtg} - u \right\|^2$$

$$u = \left(u_{gtg} - u\right)^T \left(u_{gtg} - u\right)$$

$$u = u_{gtg}^T u_{gtg} - 2u_{gtg}^T u + u^T u$$

$$u = \frac{1}{2} u^T 2Iu - 2u_{gtg}^T u$$

From this representation we can read that Q needs to be an identity matrix multiplied by 2 and that c is negative two times the go-to-goal controller input signal. We also know from before that u is has eight elements from the four robots, so Q and c are:

$$Q = \begin{bmatrix} 2 & & & & & & & \\ & 2 & & & & & & \\ & & 2 & & & & & \\ & & & 2 & & & & \\ & & & & 2 & & & \\ & & & & & 2 & & \\ & & & & & & 2 & \\ & & & & & & & 2 \end{bmatrix}$$

$$c = \begin{bmatrix} -2u_{x,1}^{gtg} & -2u_{y,1}^{gtg} & -2u_{x,2}^{gtg} & -2u_{y,2}^{gtg} & -2u_{x,3}^{gtg} & -2u_{y,3}^{gtg} & -2u_{x,4}^{gtg} & -2u_{y,4}^{gtg} \end{bmatrix}^T$$

We can calculate that for 4 agents we will need 6 constraints so that all the robots can be aware of one another's position. With this information we know that H will be a 6x8 matrix and b will be a 6x1 matrix. We need to calculate from equation 3 the constraint for all the robot pairs that we can make from our 4 agents.

Now from the constraint we can calculate the H and b for our first agent pair i and j.

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

$$\left(\frac{\partial \left\|x_i - x_j\right\|^2 - 0.2^2}{\partial x_i}\right)^T u_i + \left(\frac{\partial \left\|x_i - x_j\right\|^2 - 0.2^2}{\partial x_j}\right)^T u_j \geq -2\gamma \left(\left\|x_i - x_j\right\|^2 - 0.2^2\right)^3, \forall j \neq i$$

$$2(x_i - x_j)u_i - 2(x_i - x_j)u_j \geq -2\gamma(\left\|x_i - x_j\right\|^2 - 0.2^2)^3,\ \forall j \neq i$$

$$-2(x_i - x_j)u_i - 2(x_j - x_i)u_j \leq 2\gamma(\left\|x_i - x_j\right\|^2 - 0.2^2)^3,\ \forall j \neq i$$

From this we can read the H and b as we know that in the H matrix if the agent in question isn't one of the i or j agents derivative will be equal to 0. Which leads us to the following:

$$H = \begin{bmatrix} -2(p_{x,1}-p_{x,2}) & -2(p_{y,1}-p_{y,2}) & -2(p_{x,2}-p_{x,1}) & -2(p_{y,2}-p_{y,1}) & 0 & 0 & 0 & 0 \\ -2(p_{x,1}-p_{x,3}) & -2(p_{y,1}-p_{y,3}) & 0 & 0 & -2(p_{x,3}-p_{x,1}) & -2(p_{y,3}-p_{y,1}) & 0 & 0 \\ -2(p_{x,1}-p_{x,4}) & -2(p_{y,1}-p_{y,4}) & 0 & 0 & 0 & 0 & -2(p_{x,4}-p_{x,1}) & -2(p_{y,4}-p_{y,1}) \\ 0 & 0 & -2(p_{x,2}-p_{x,3}) & -2(p_{y,2}-p_{y,3}) & -2(p_{x,3}-p_{x,2}) & -2(p_{y,3}-p_{y,2}) & 0 & 0 \\ 0 & 0 & -2(p_{x,2}-p_{x,4}) & -2(p_{y,2}-p_{y,4}) & 0 & 0 & -2(p_{x,4}-p_{x,2}) & -2(p_{y,4}-p_{y,2}) \\ 0 & 0 & 0 & 0 & -2(p_{x,3}-p_{x,4}) & -2(p_{y,3}-p_{y,4}) & -2(p_{x,4}-p_{x,3}) & -2(p_{y,4}-p_{y,3}) \end{bmatrix}$$

$$b = 2\gamma \begin{bmatrix} (p_{x,1}-p_{x,2})^2 + (p_{y,1}-p_{y,2})^2 - 0.2^2 \\ (p_{x,1}-p_{x,3})^2 + (p_{y,1}-p_{y,3})^2 - 0.2^2 \\ (p_{x,1}-p_{x,4})^2 + (p_{y,1}-p_{y,4})^2 - 0.2^2 \\ (p_{x,2}-p_{x,3})^2 + (p_{y,2}-p_{y,3})^2 - 0.2^2 \\ (p_{x,2}-p_{x,4})^2 + (p_{y,2}-p_{y,4})^2 - 0.2^2 \\ (p_{x,3}-p_{x,4})^2 + (p_{y,3}-p_{y,4})^2 - 0.2^2 \end{bmatrix}^3$$

We can now implement the quadratic programming as we have all the parameters for the general form. This gives us the result which are presented in figure 1-3.

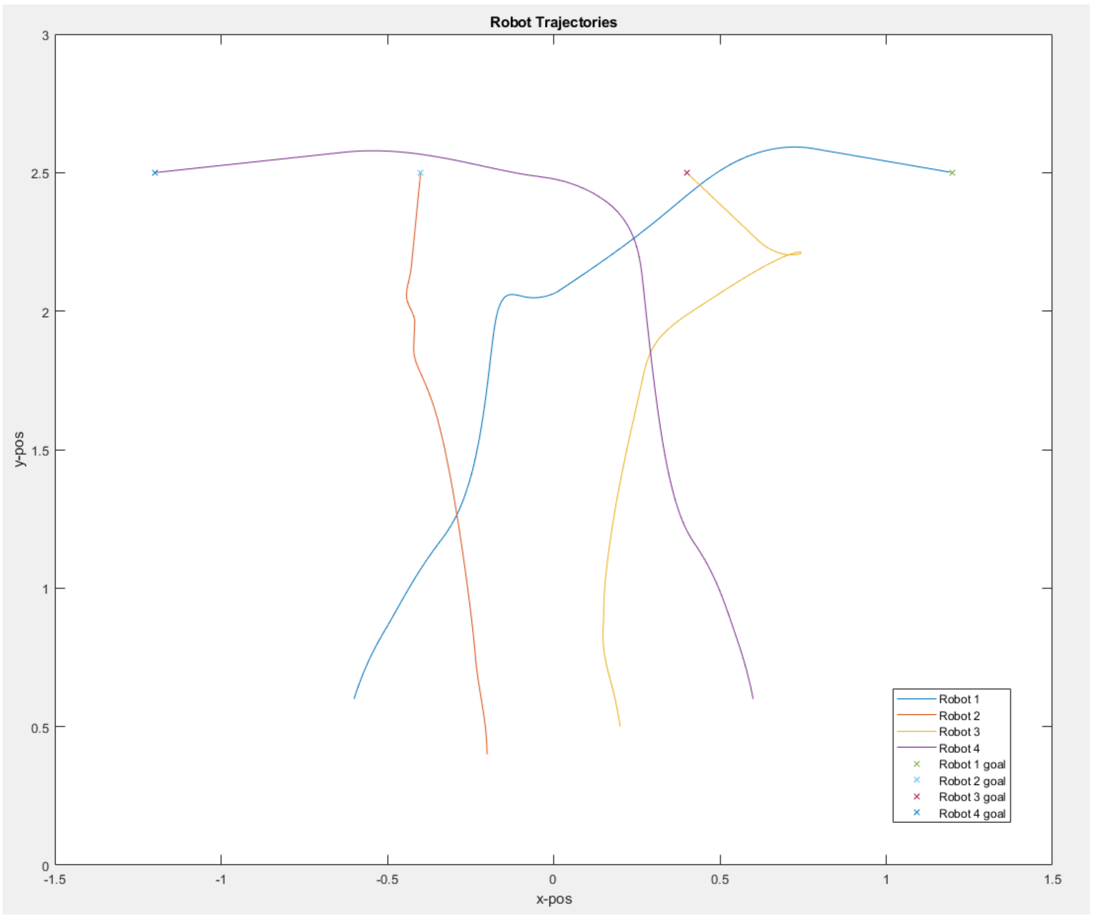

Figure 1: Robot trajectories for centralized control

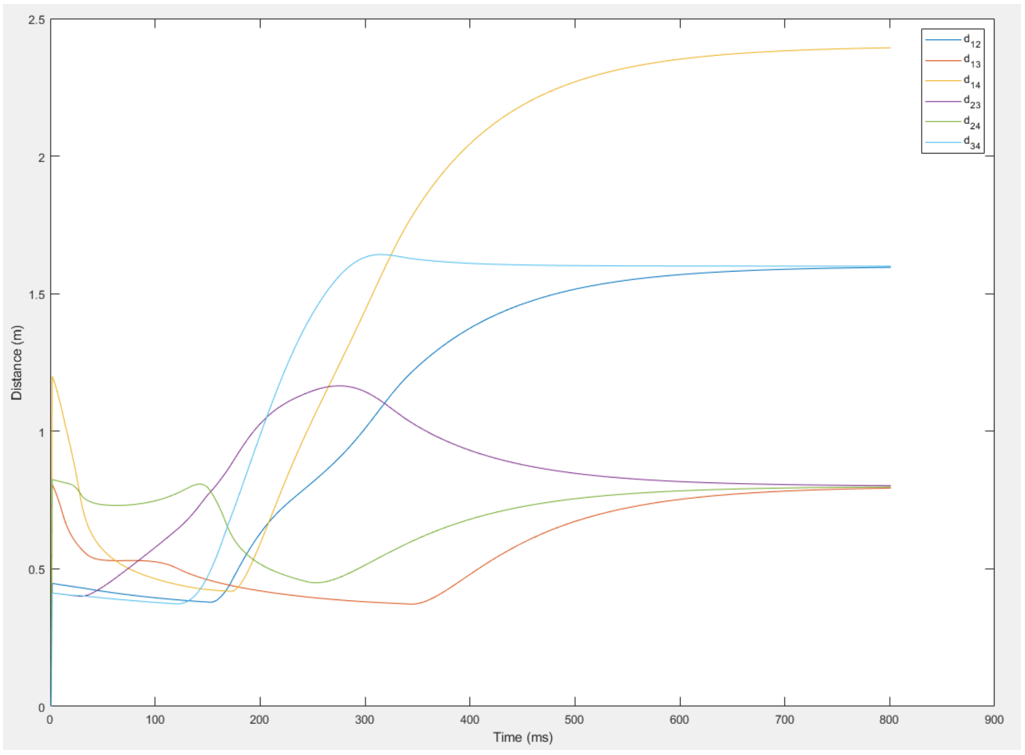

Figure 2: Time series of the distance between all pairs of robots for centralized control

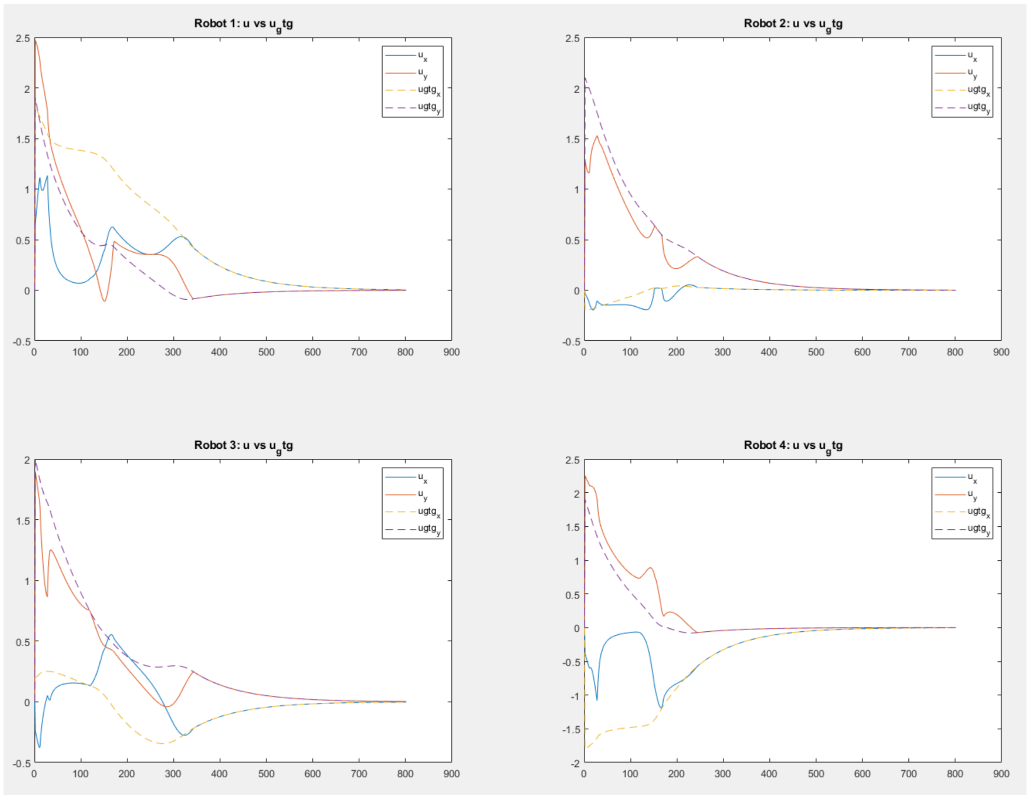

Figure 3: Comparison between the nominal on optimal control inputs for centralized control

Problem 2

We need to create a distributed optimization-based safety control as follows:

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

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

With $N_i^s = \left\{ j, j \neq i \,||\, \left\| x_i - x_j \right\|^2 \leq 1^2 \right\}$ we will use the same $h_o(x_i, x_j)$ and γ as in problem 1.

Given that the general form of quadratic programming for distributed case is

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

$$s.t.\ H_i u_i \ \leq b_i$$

Assuming a case where robot 1 is nearby all other robots i.e., $N_i^s = \{2, 3, 4\}$, we can compute Q_1, c_1, H_1 and b_1.

Let us again start by calculating the Q_1 and c_1.

$$u_1 = \arg\min_{u_1} \left\| u_{gtg,1} - u_1 \right\|^2$$

$$u_1 = \left(u_{gtg,1} - u_1\right)^T \left(u_{gtg,1} - u_1\right)$$

$$u_1 = u_{gtg,1}^T u_{gtg,1} - 2u_{gtg,1}^T u_1 + u_1^T u_1$$

$$u_1 = \frac{1}{2} u_1^T 2I u_1 - 2u_{gtg,1}^T u_1$$

From this we can see the Q_1 and c_1 given that $u_1 = [u_{x,1} \quad u_{y,1}]^T \in \mathbb{R}^2$

$$Q_1 = \begin{bmatrix} 2 & \\ & 2 \end{bmatrix}$$

$$c_1 = \begin{bmatrix} -2u_{x,1}^{gtg} & -2u_{y,1}^{gtg} \end{bmatrix}^T$$

Now we need to compute the H_1 and b_1 parameters. We know that we have all the other agents nearby, so we need to take them into account. This means that H will be of size 3x2 and b 3x1.

$$H_1 = \begin{bmatrix} -2(p_{x,1} - p_{x,2}) & -2(p_{y,1} - p_{y,2}) \\ -2(p_{x,1} - p_{x,3}) & -2(p_{y,1} - p_{y,3}) \\ -2(p_{x,1} - p_{x,4}) & -2(p_{y,1} - p_{y,4}) \end{bmatrix}$$

$$b_1 = \gamma \begin{bmatrix} (p_{x,1} - p_{x,2})^2 + (p_{y,1} - p_{y,2})^2 - 0.2^2 \\ (p_{x,1} - p_{x,3})^2 + (p_{y,1} - p_{y,3})^2 - 0.2^2 \\ (p_{x,1} - p_{x,4})^2 + (p_{y,1} - p_{y,4})^2 - 0.2^2 \end{bmatrix}^3$$

We can now try and implement the quadratic programming for the distributed version as we have some knowledge about what we need to do. This gives us the result which are presented in figure 4-6.

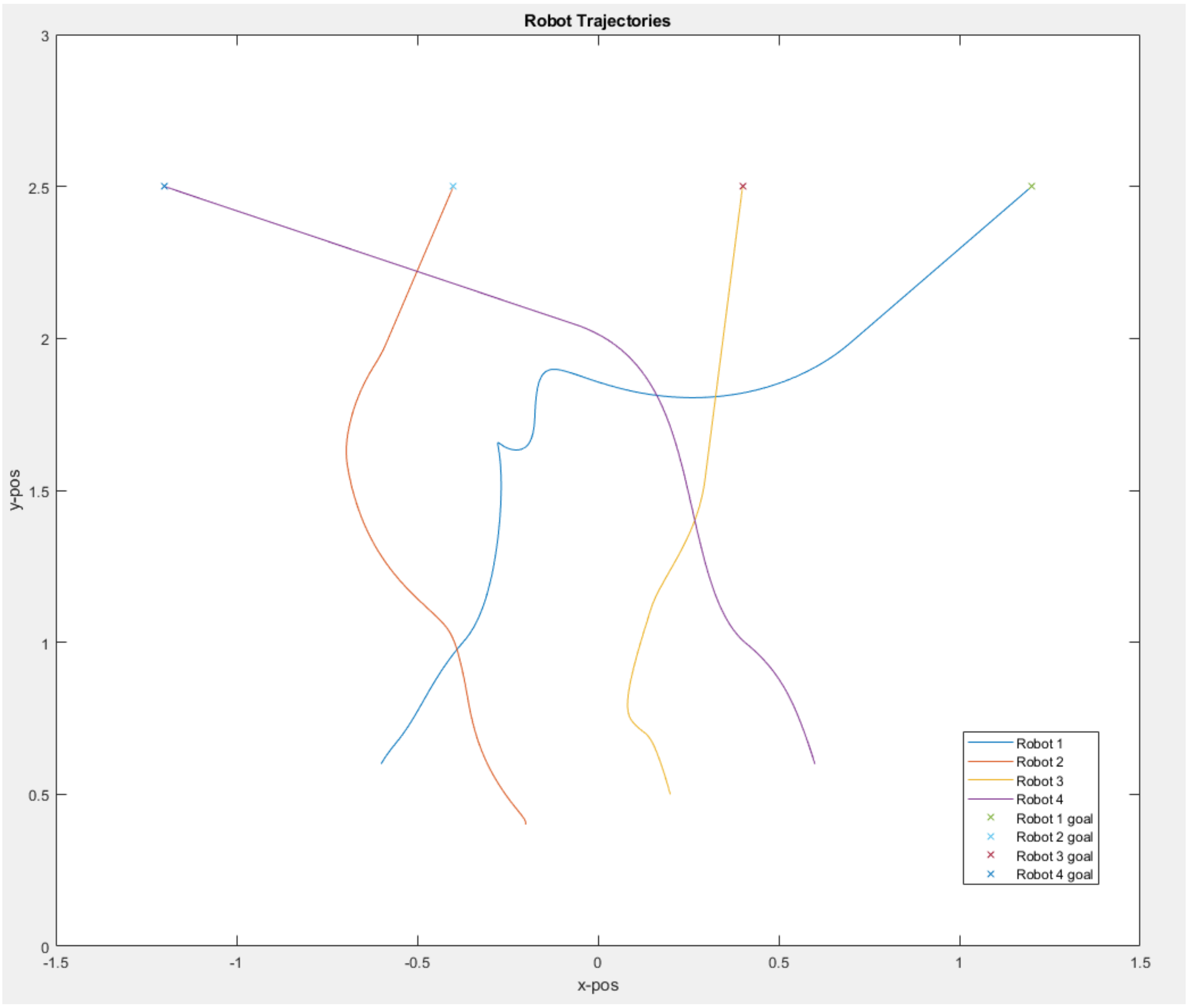

Figure 4: Robot trajectories for distributed control

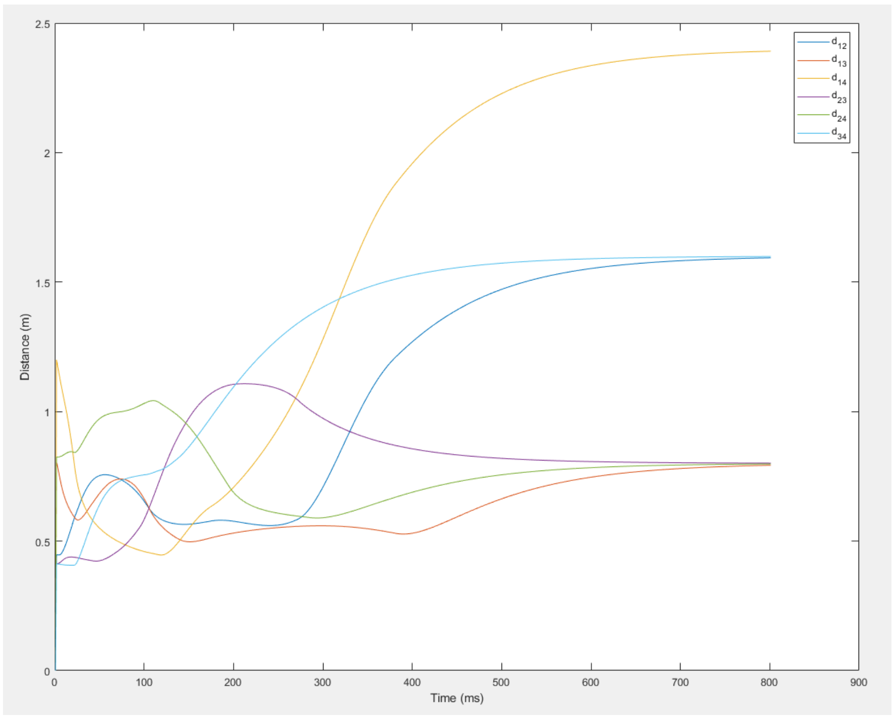

Figure 5: Time series of the distance between all pairs of robots for distributed control

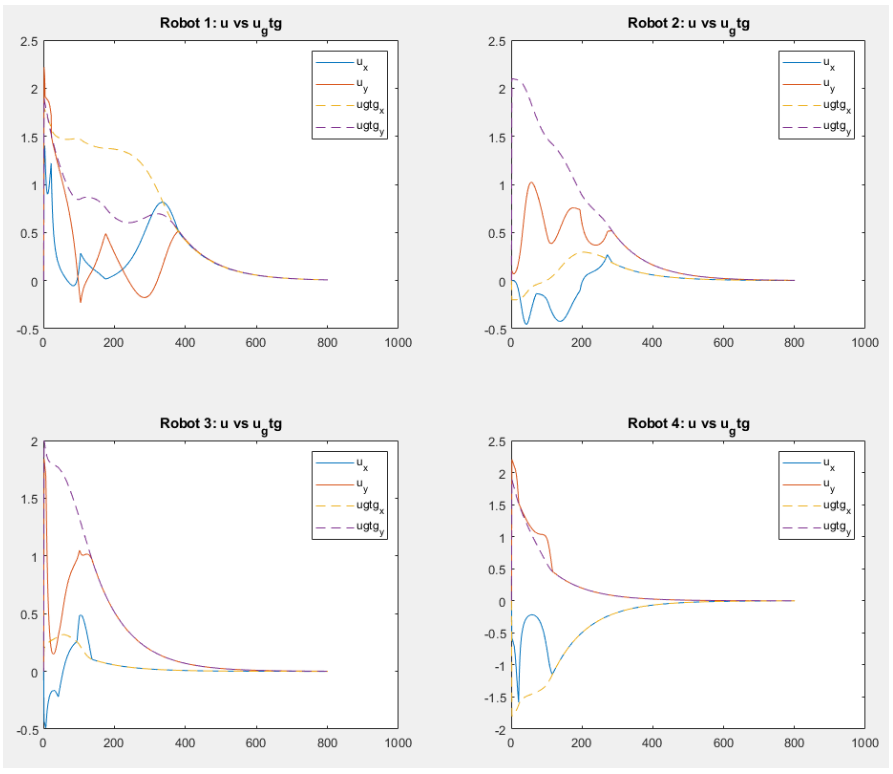

Figure 6: Comparison between the nominal on optimal control inputs for distributed control

When comparing the robot trajectories in figure 1 and figure 4 we can see that the robots take a much more linear path toward their goal than in figure 4 as the centralized controller always knows where every robot is and what it is going to do so it can find a more optimal path for the robots to take. Whereas in the figure 4 we can see that the robots take a more winding path as they don't know where every other robot is or what they are going to do. This leads to not as clean paths for the robots. In the distributed solution however, the robots would be fine to continue moving on their own even if one of the other robots malfunctioned. This is not the case with the centralized solution cause if the centralized controller malfunctioned all the robots would seize to do their job. This holds true for all distributed applications as they are more robust and more scalable than the centralized versions but can fail to achieve as good performance. Both solutions seem to achieve the goal destinations without collisions.

Problem 3

In this problem we have additional constraints where all agents need to maintain the network connectivity when moving, and the controller is then updated into:

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

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

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

Where $h_c(x_i,x_j) = 0.9^2 - \left\|x_i - x_j\right\|^2$ and N_i^c denotes the robot *i*'s neighbours defined by the prescribed graph communication topology.

Given the general form of quadratic programming for distributed case in problem 2. Assuming a case where robot 1 is nearby all other robots i.e. $N_i^s = \{2,3,4\}$ and its communication neighbours are specified by $N_i^c = \{2,3,4\}$. We can compute the new Q_1, c_1, H_1 and b_1.

We can once again start with computing the Q$_1$ and c$_1$ matrices. Because the only difference to problem 2 in the equation is that there is one more constraint equation the Q$_1$ and c$_1$ will be the same as in problem 2.

$$Q_1 = \begin{bmatrix} 2 & \\ & 2 \end{bmatrix}$$

$$c_1 = \begin{bmatrix} -2u_{x,1}^{gtg} & -2u_{y,1}^{gtg} \end{bmatrix}^T$$

When compared to problem 2 we can see that we have another constraint. Robot 1 is also nearby all other robots, so we need to take them into account as in problem 2. In addition, we are communication neighbours to all of them as well which means that we need 3 more rows in the H and b matrices. This will give our H$_1$ the size of 6x2 and our b$_1$ the size of 6x1. The first 3 rows of our H$_1$ and b$_1$ will be the same as in problem 2 and the other 3 rows will be made with the other constraint in mind giving us the following:

$$H_1 = \begin{bmatrix} -2(p_{x,1} - p_{x,2}) & -2(p_{y,1} - p_{y,2}) \\ -2(p_{x,1} - p_{x,3}) & -2(p_{y,1} - p_{y,3}) \\ -2(p_{x,1} - p_{x,4}) & -2(p_{y,1} - p_{y,4}) \\ 2(p_{x,1} - p_{x,2}) & 2(p_{y,1} - p_{y,2}) \\ 2(p_{x,1} - p_{x,3}) & 2(p_{y,1} - p_{y,3}) \\ 2(p_{x,1} - p_{x,4}) & 2(p_{y,1} - p_{y,4}) \end{bmatrix}$$

$$b_1 = \gamma \begin{bmatrix} (p_{x,1} - p_{x,2})^2 + (p_{y,1} - p_{y,2})^2 - 0.2^2 \\ (p_{x,1} - p_{x,3})^2 + (p_{y,1} - p_{y,3})^2 - 0.2^2 \\ (p_{x,1} - p_{x,4})^2 + (p_{y,1} - p_{y,4})^2 - 0.2^2 \\ 0.9^2 - (p_{x,1} - p_{x,2})^2 + (p_{y,1} - p_{y,2})^2 \\ 0.9^2 - (p_{x,1} - p_{x,3})^2 + (p_{y,1} - p_{y,3})^2 \\ 0.9^2 - (p_{x,1} - p_{x,4})^2 + (p_{y,1} - p_{y,4})^2 \end{bmatrix}^3$$

In the simulation the H matrix is of size 8x2 and the b matrix of size 8x1 for the ease of calculations but the rows corresponding to the robot itself will always be 0.

With this we can try to implement the network connectivity constraint to our network. This gives us the results which are provided in figure 7-9.

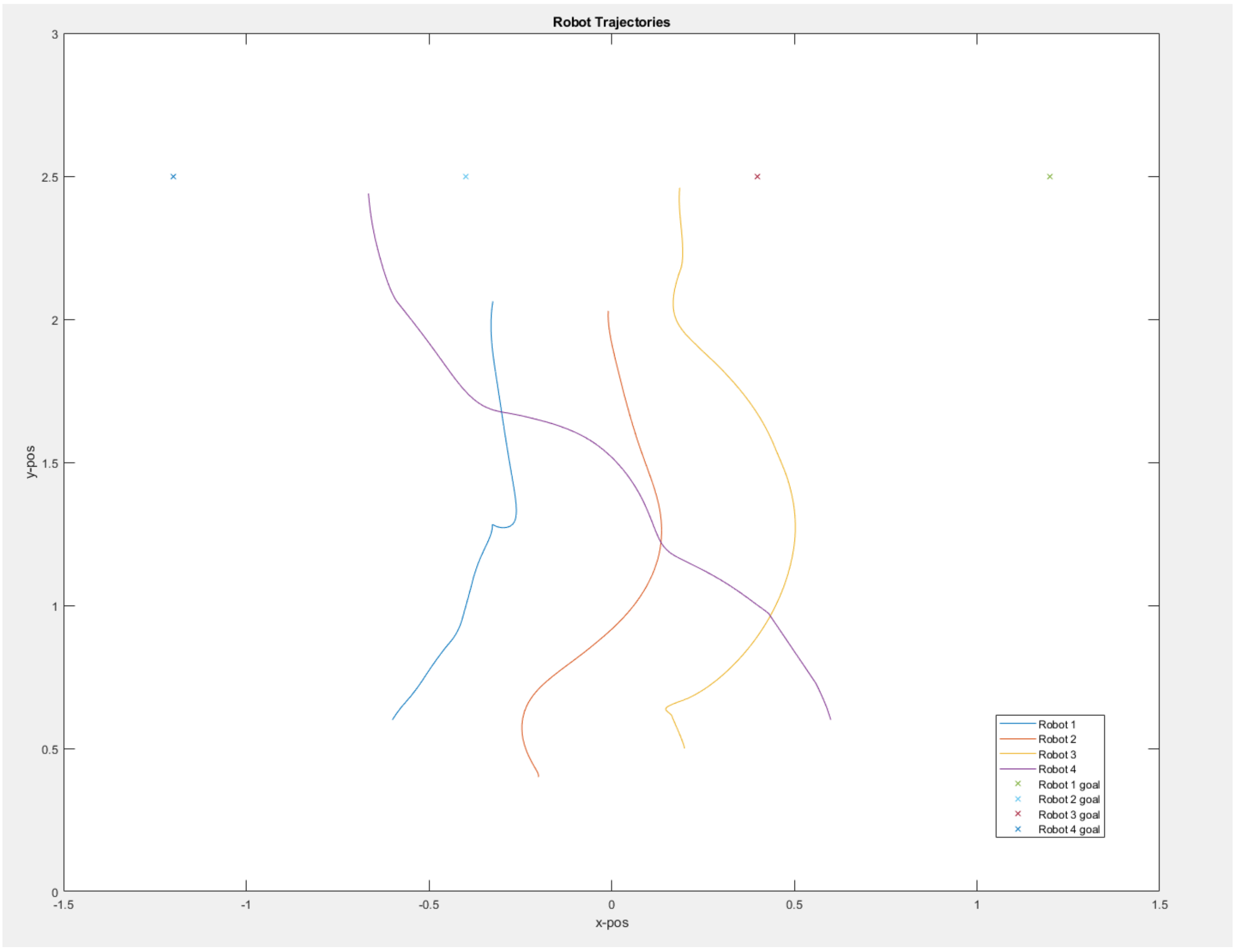

Figure 7: Robot trajectories for distributed control with full connection constraint

As we can see the robots do not reach the goal positions with this full connected network topology as the constraint requires all the robots to be always within 0.9 m from one another.

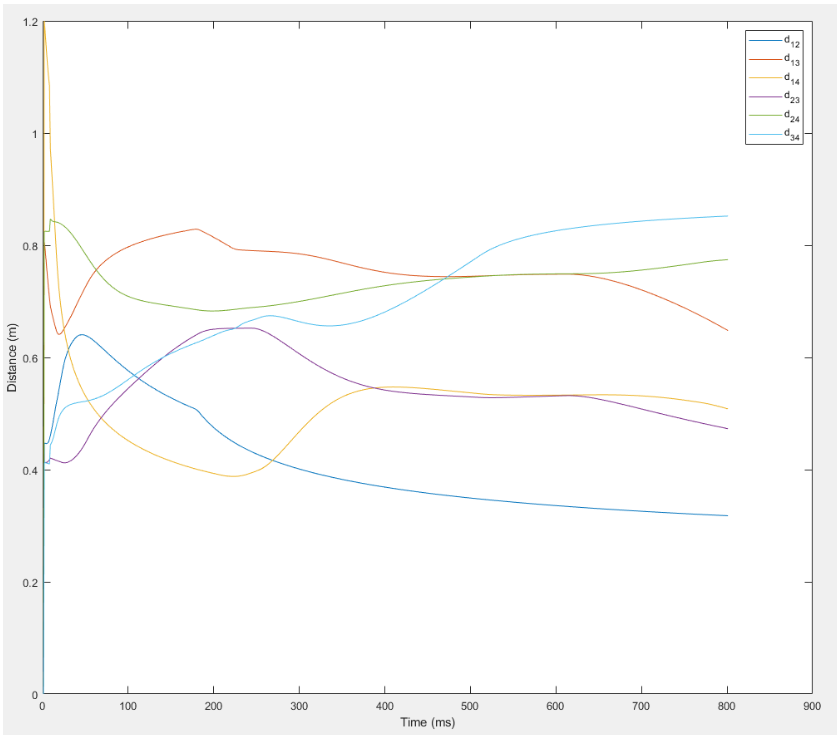

Figure 8: Time series of the distance between all pairs of robots for distributed control with full connection constraint

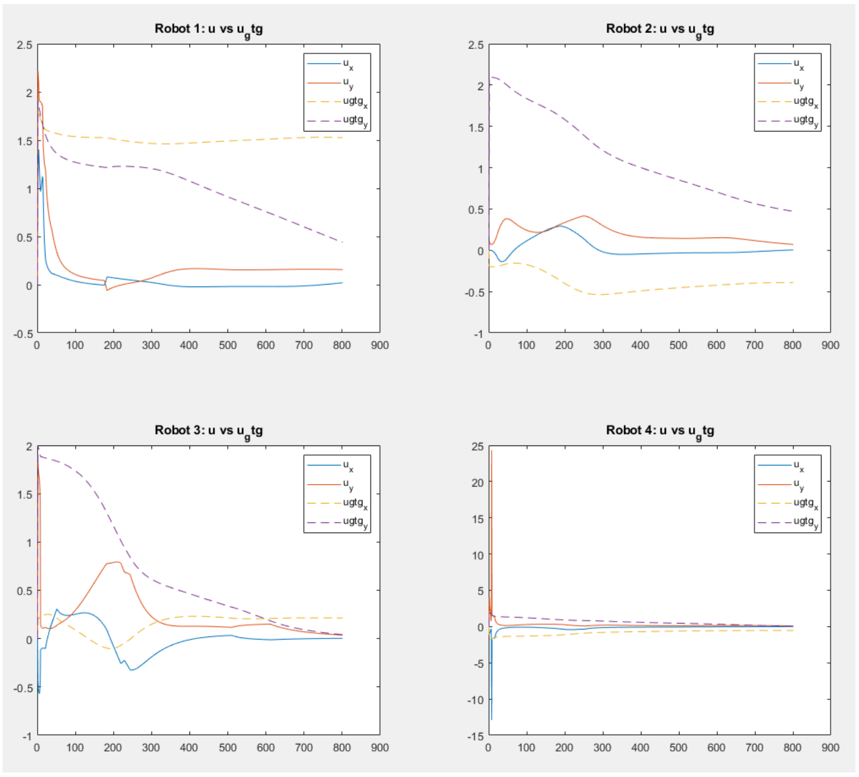

Figure 9: Comparison of control signals for distributed control with full connection constraint

The robots do not reach their own goals because the robots need to stay within 0.9m distance from one another but the goals are further than that apart from each other for some of the robots. And as every robot needs to be within the 0.9 m distance from every other robot the goal positions are not achievable for all the agents.

To achieve the desired goal positions a different communication network is required. We know that the robot goal positions are 0.8 m apart from one another in a line. So, the robots need to be in contact only with the ones whose goal is next to its goal. This leaves us with the following adjacency matrix for the communication network.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The results for this new communication network are presented in figures 10-12.

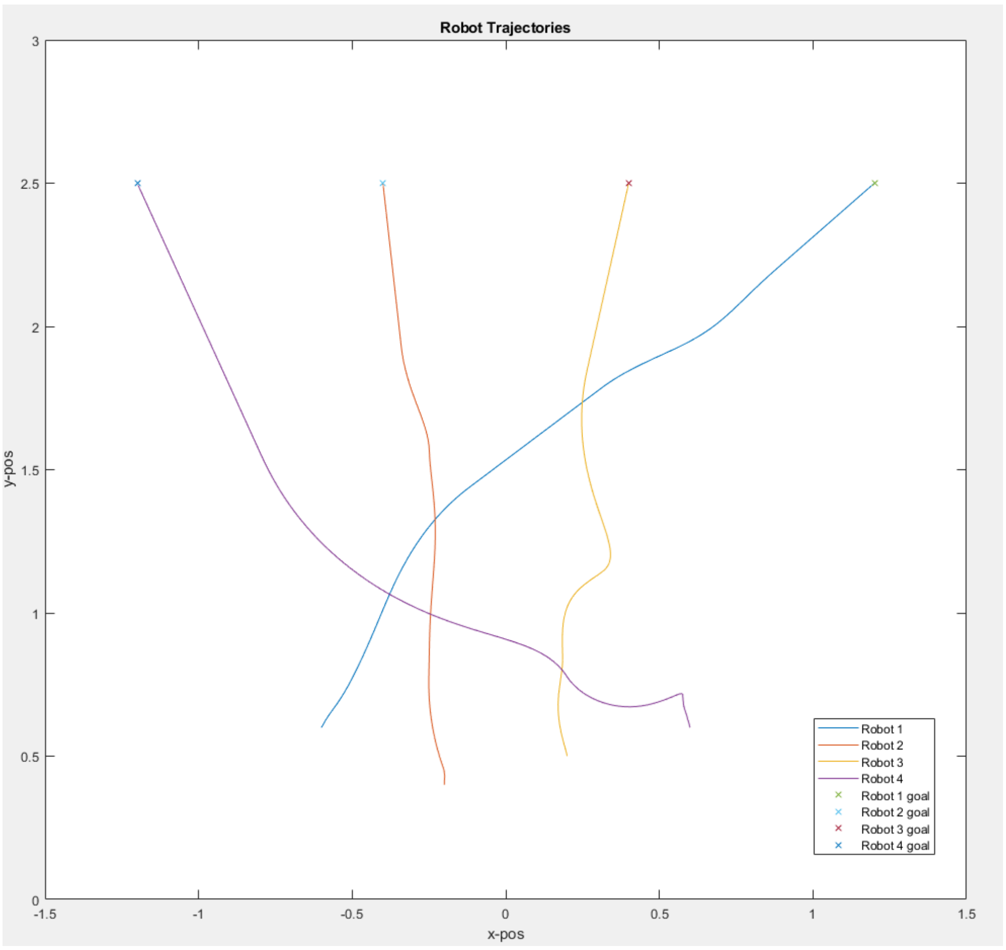

Figure 10: Robot trajectories for distributed control with new connection constraints

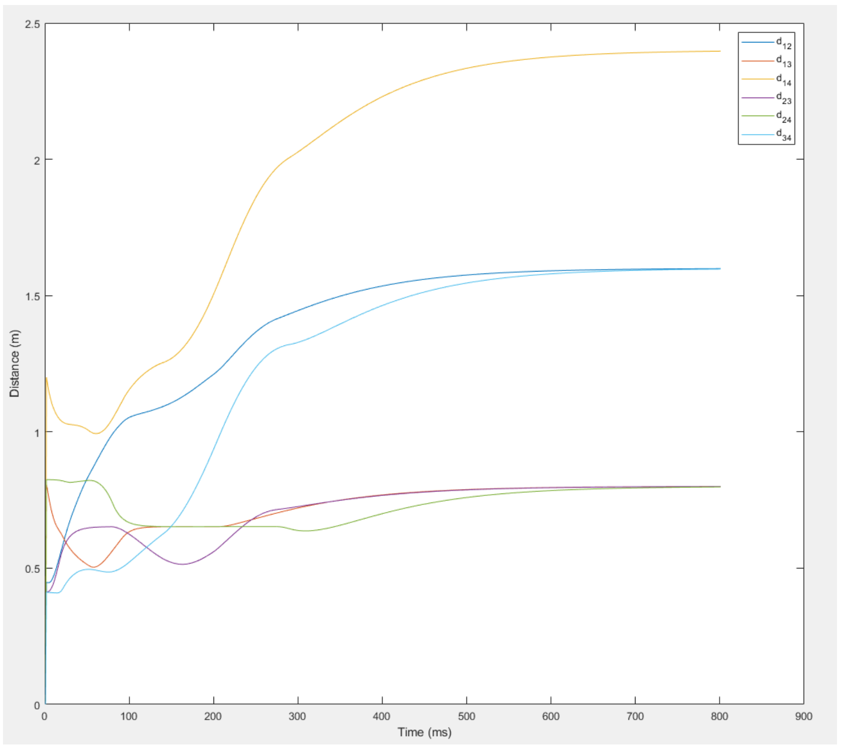

Figure 11: Time series of the distance between all pairs of robots for distributed control with new connection constraint

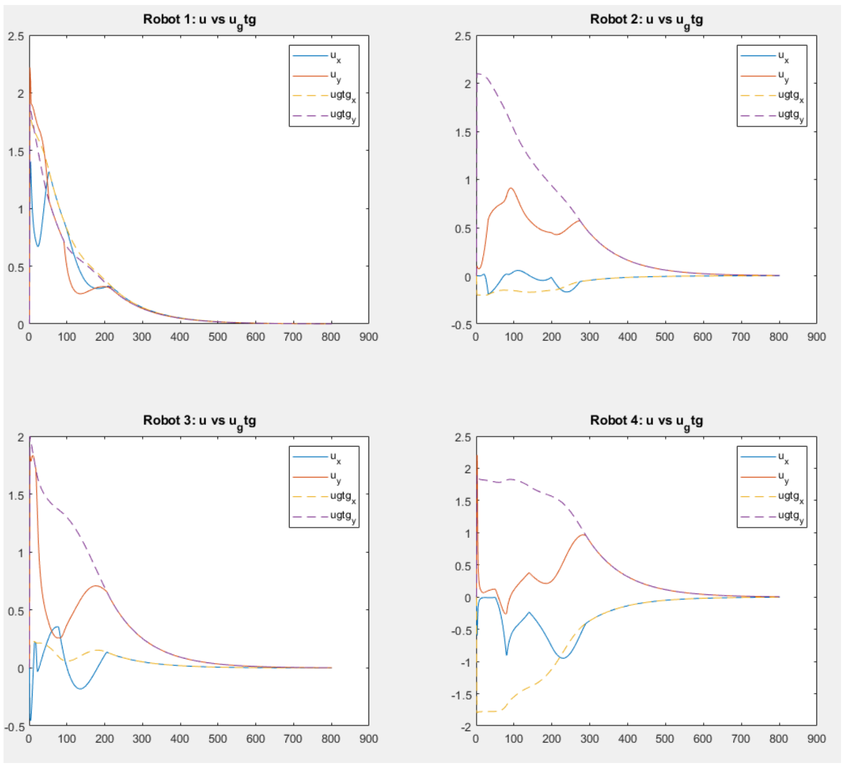

Figure 12: Comparison of control signals for distributed control with new connection constraint

As we can see the new network connection topology achieves the desired result. When the robots reach the connected robots, they are “connected” (within 0.9 m from one another) at all times to at least one other robot and they all reach the goal position without collisions. This new network topology achieves what was asked.

Sources

(1) Homework 10, M.W.S. Atman, M. Iqbal, A. Gusrialdi, April 14, 2023 https://moodle.tuni.fi/pluginfile.php/3216553/mod_resource/content/2/2023_DistControl_Problem_set10_updated.pdf
(2) Lecture 11 slides, A. Gusrialdi, April 14, 2023. https://moodle.tuni.fi/pluginfile.php/3213966/mod_resource/content/2/AUT360_lecture11_handout.pdf
(3) Exercise 11, M.W.S Atman, April 14, 2023 https://moodle.tuni.fi/pluginfile.php/3217031/mod_resource/content/1/Exercise_session_11.pdf