ARS1 Project 1: Rendezvous Problem

We suppose the case of six robots, where we can control directly their speed. The objective is to let the robots meet each other. The initial positions of the robots are: (-4, 8); (-3, -4); (3, -1); (6, 3); (2, -5); (3, 7). We suppose that all the robots communicate between themselves.

- *Plot the graph of the system*

An arrow of the graph from i to j means that the robot j receives information (about position and speed) from the robot i. In the case that all the robots communicate between themselves, each robot sends and receives information from any other robots. Thus the graph of the system is the following:

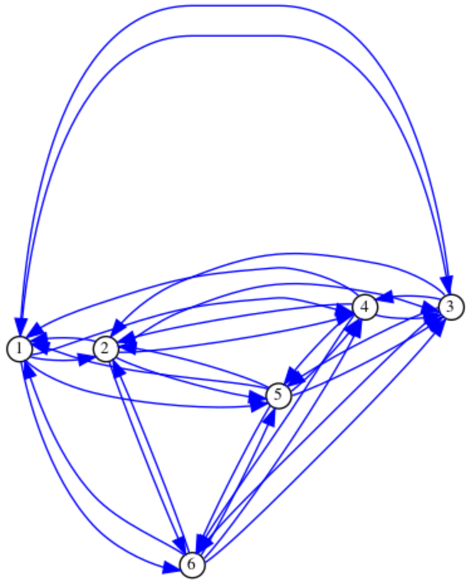
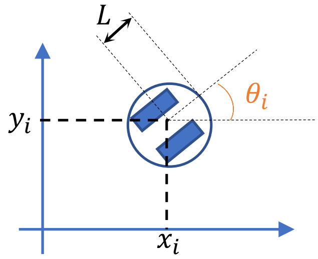


Figure 1: Graph of case 1 (all the robots communicate between themselves)

- *Propose a consensus-based control law that let the agents meet together*

We suppose the kinematic equations of the robot i ($i = 1, \dots, 6$):

$$\begin{cases} \dot{x}_i = v_i \cos\theta_i \\ \dot{y}_i = v_i \sin\theta_i \\ \dot{\theta}_i = \omega_i \end{cases}$$



Where:

- $(x_i; y_i)$: position of robot i
- θ_i : yaw angle of robot i
- v_i and ω_i : linear and angular velocities of robot i

Non-holonomic constraints: the robots cannot move sideways

$$\dot{x}_i.\sin\theta_i - \dot{y}_i.\cos\theta_i = 0$$

$$\begin{pmatrix} \theta_i = 0 \;\Rightarrow\; \dot{y}_i = 0 \\ \theta_i = \frac{\pi}{2} \;\Rightarrow\; \dot{x}_i = 0 \end{pmatrix}$$

It is difficult to design v_i and w_i such that $x_i \to 0$, $y_i \to 0$, $\theta_i \to 0$.

<u>Feedback linearization</u>: We use feedback linearization to simplify this problem. We define a point $(x_{h_i}; y_{h_i})$ at the front of the robot i:

$$\begin{cases} x_{h_i} = x_i + L\cos\theta_i \\ y_{h_i} = y_i + L\sin\theta_i \end{cases}$$

$$\Rightarrow \begin{pmatrix} \dot{x_{h_i}} \\ \dot{y_{h_i}} \end{pmatrix} = \begin{pmatrix} \cos\theta_i & -L\sin\theta_i \\ \sin\theta_i & L\cos\theta_i \end{pmatrix} \begin{pmatrix} v_i \\ \omega_i \end{pmatrix} = \begin{pmatrix} u_{x_i} \\ u_{y_i} \end{pmatrix}$$

$(u_{x_i}; u_{y_i})$ is the input of the closed loop system. We define the following input in order to have exponential convergence of the robots' position coordinates:

$$\begin{cases} u_{x_i} = -\sum_{j=1}^{6} g_{ij} k_{ij} \left(x_{h_i} - x_{h_j}\right) \\ u_{y_i} = -\sum_{j=1}^{6} g_{ij} k_{ij} \left(y_{h_i} - y_{h_j}\right) \end{cases}$$

Given that $i \neq j$, $g_{ij} = 1$ if robot i receives information from robot j, otherwise $g_{ij} = 0$. k_{ij} is the gain of the control term between robot i and robot j. It allows us to increase or decrease the speed of convergence of robot i towards the position of robot j.

According to the previous graph, $g_{ij} = 1$ for all i and j. Consequently:

$$\dot{X} = AX$$

$$X = \begin{bmatrix} x_{h_1} \\ y_{h_1} \\ x_{h_2} \\ y_{h_2} \\ x_{h_3} \\ y_{h_3} \\ x_{h_4} \\ y_{h_4} \\ x_{h_5} \\ y_{h_5} \\ x_{h_6} \\ y_{h_6} \end{bmatrix} \text{ and } \dot{X} = \begin{bmatrix} \dot{x}_{h_1} \\ \dot{y}_{h_1} \\ \dot{x}_{h_2} \\ \dot{y}_{h_2} \\ \dot{x}_{h_3} \\ \dot{y}_{h_3} \\ \dot{x}_{h_4} \\ \dot{y}_{h_4} \\ \dot{x}_{h_5} \\ \dot{y}_{h_5} \\ \dot{x}_{h_6} \\ \dot{y}_{h_6} \end{bmatrix}$$

$$A = \begin{bmatrix}
-\sum_{j\neq 1}^{1\dots 6} k_{1j} & 0 & k_{12} & 0 & k_{13} & 0 & k_{14} & 0 & k_{15} & 0 & k_{16} & 0 \\
0 & -\sum_{j\neq 1}^{1\dots 6} k_{1j} & 0 & k_{12} & 0 & k_{13} & 0 & k_{14} & 0 & k_{15} & 0 & k_{16} \\
k_{21} & 0 & -\sum_{j\neq 2}^{1\dots 6} k_{2j} & 0 & k_{23} & 0 & k_{24} & 0 & k_{25} & 0 & k_{26} & 0 \\
0 & k_{21} & 0 & -\sum_{j\neq 2}^{1\dots 6} k_{2j} & 0 & k_{23} & 0 & k_{24} & 0 & k_{25} & 0 & k_{26} \\
k_{31} & 0 & k_{32} & 0 & -\sum_{j\neq 3}^{1\dots 6} k_{3j} & 0 & k_{34} & 0 & k_{35} & 0 & k_{36} & 0 \\
0 & k_{31} & 0 & k_{32} & 0 & -\sum_{j\neq 3}^{1\dots 6} k_{3j} & 0 & k_{34} & 0 & k_{35} & 0 & k_{36} \\
k_{41} & 0 & k_{42} & 0 & k_{43} & 0 & -\sum_{j\neq 4}^{1\dots 6} k_{4j} & 0 & k_{45} & 0 & k_{46} & 0 \\
0 & k_{41} & 0 & k_{42} & 0 & k_{43} & 0 & -\sum_{j\neq 4}^{1\dots 6} k_{4j} & 0 & k_{45} & 0 & k_{46} \\
k_{51} & 0 & k_{52} & 0 & k_{53} & 0 & k_{54} & 0 & -\sum_{j\neq 5}^{1\dots 6} k_{5j} & 0 & k_{56} & 0 \\
0 & k_{51} & 0 & k_{52} & 0 & k_{53} & 0 & k_{54} & 0 & -\sum_{j\neq 5}^{1\dots 6} k_{5j} & 0 & k_{56} \\
k_{61} & 0 & k_{62} & 0 & k_{63} & 0 & k_{64} & 0 & k_{65} & 0 & -\sum_{j\neq 6}^{1\dots 6} k_{6j} & 0 \\
0 & k_{61} & 0 & k_{62} & 0 & k_{63} & 0 & k_{64} & 0 & k_{65} & 0 & -\sum_{j\neq 6}^{1\dots 6} k_{6j}
\end{bmatrix}$$

- *<u>Implement the controlled system in MATLAB/Simulink and analyze the results function of the controllers' gains.</u>*

1) We first implement the most simple case, that is when $k_{ij} = 1$ for all i and j ($i \neq j$). We observe that x and y converge towards their final value in about 1 second.

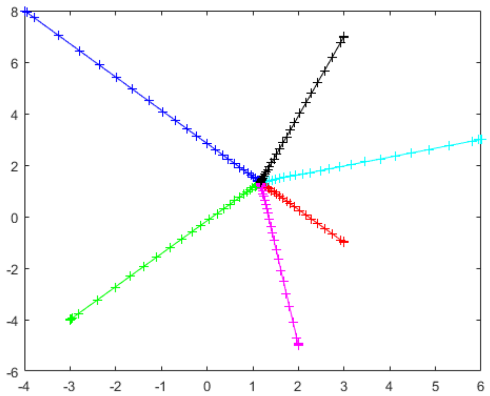


Figure 2 : $k_{ij} = 1$ for all i and j ($i \neq j$), space trajectory of the 6 rovers

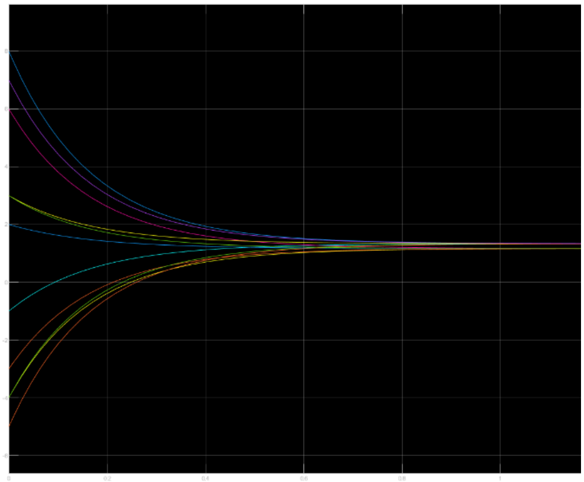


Figure 3.1 : $k_{ij} = 1$ for all i and j ($i \neq j$), time variation of x and y

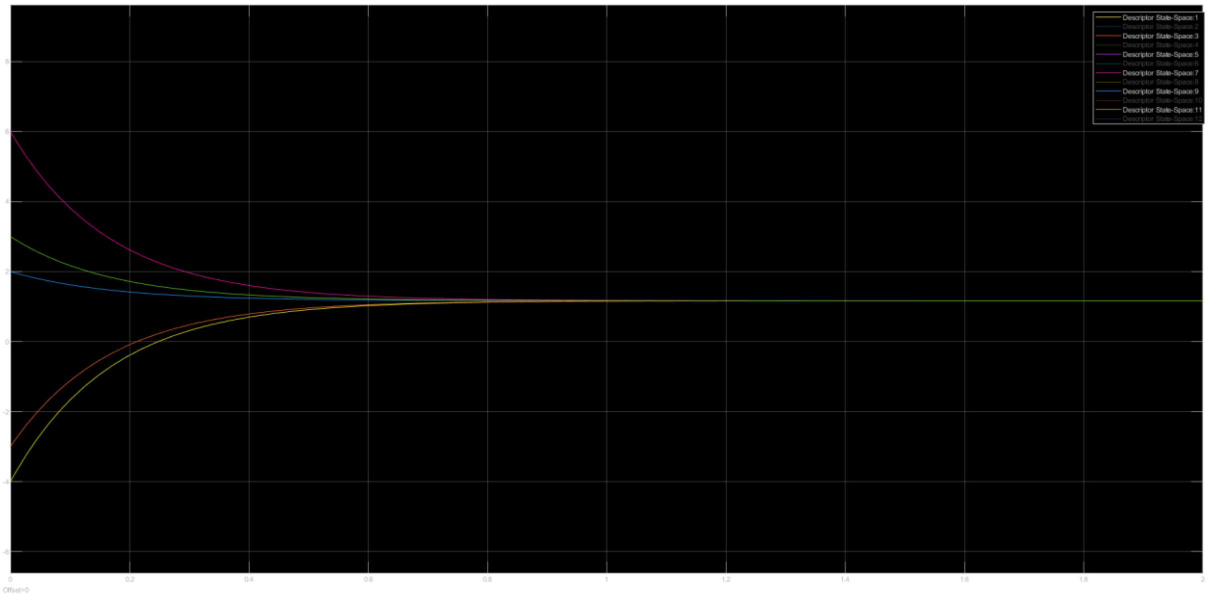


Figure 3.2 : $k_{ij} = 1$ for all i and j ($i \neq j$), time variation of x ($x_{h_i} \to 1.166$)

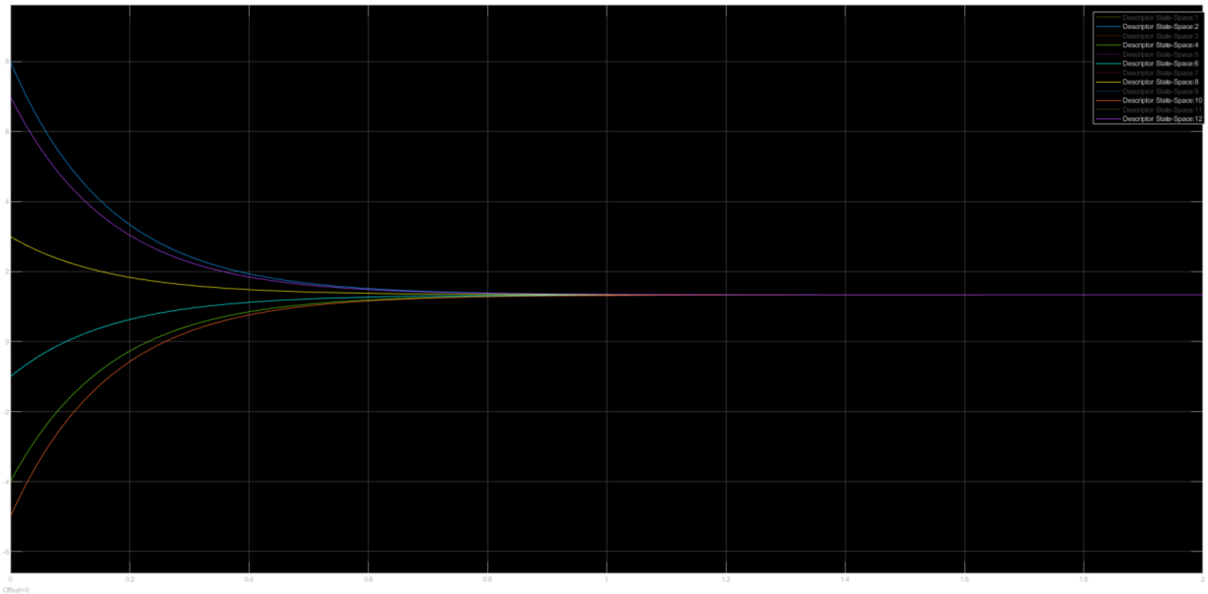


Figure 3.3 : $k_{ij} = 1$ for all i and j ($i \neq j$), time variation of y ($y_{h_i} \to 1.334$)

2) Then we want to observe the evolution of the time of convergence when we increase the controllers' gains. We run 2 simulations:
   - For the case $k_{ij} = 2$ for all i and j ($i \neq j$), x and y converge towards their final value in about 0.5 seconds.
   - For the case $k_{ij} = 3$ for all i and j ($i \neq j$), x and y converge towards their final value in about 0.4 seconds.

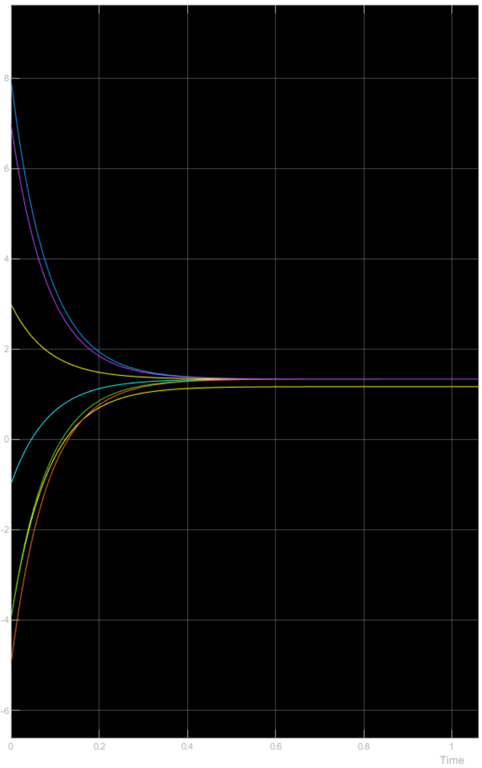


Figure 4.1 : $k_{ij} = 2$ for all i and j ($i \neq j$), time variation of x and y

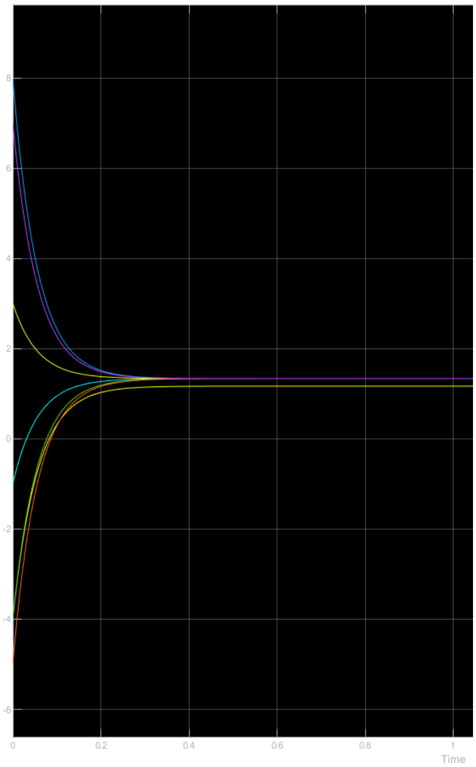


Figure 4.2 : $k_{ij} = 3$ for all i and j ($i \neq j$), time variation of x and y

By implementing a loop on Matlab which runs the Simulink simulation for different values of k_{ij} ($k_{ij} = 1, \dots, 5$), we can observe the evolution of the time of convergence (criterion of convergence : relative error for each x_{h_i} must be strictly inferior to 1%).

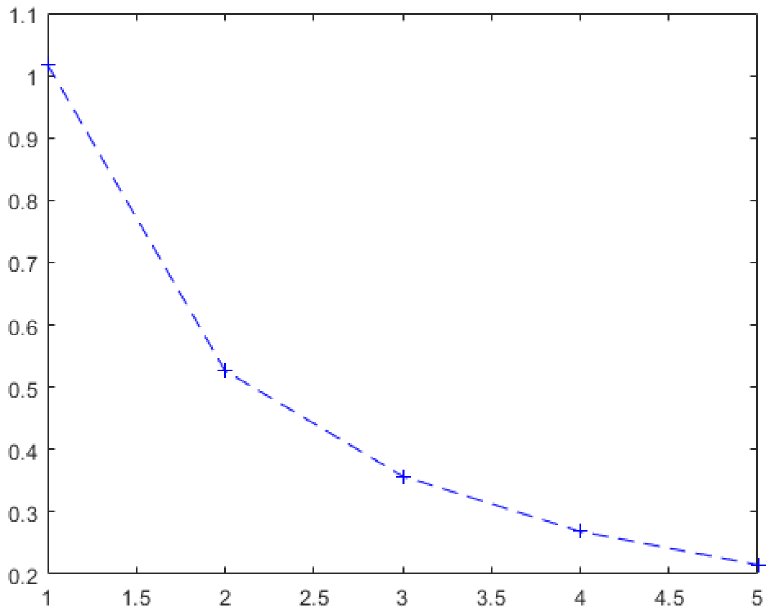


Figure 5 : Time of convergence for $k_{ij} = 1, \dots, 5$

3) Finally, we want to study a more global and difficult case, that is when k_{ij} can take different values depending on the values of i and j. For example, we consider the case in which the control term k_{1j} ($j \neq 1$) is higher than the other control terms ($k_{ij}, i \neq 1, i \neq j$). The dynamics of robot 1 is faster and it converges more rapidly towards the meeting point. However, this case is not optimal because the robots 2, 3, 4, 5 and 6 converge slowly. The time of convergence of x and y are approximately equal ($t = 0.8\ s$)

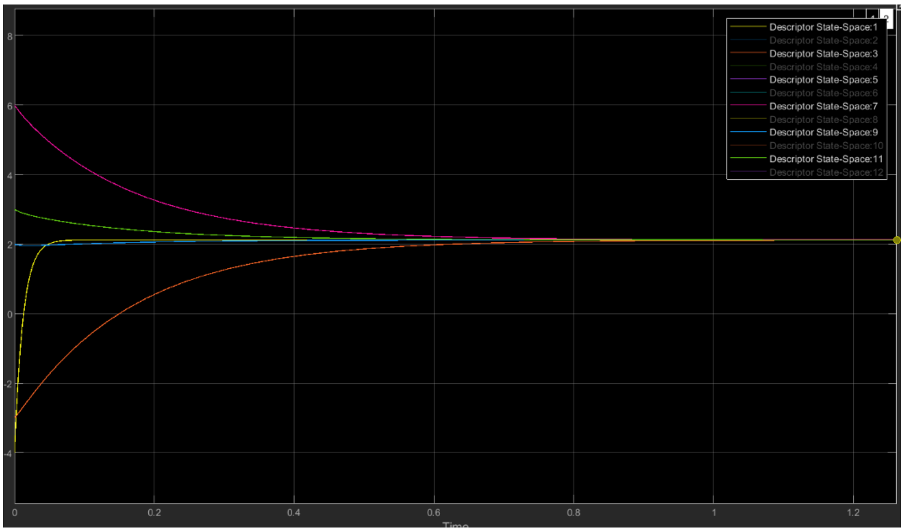


Figure 6.1: $k_{1j} = 16$ for all j ($j \neq 1$) and $k_{ij} = 1$ ($i \neq 1, i \neq j$), time variation of x ($x_{h_i} \rightarrow 2.123$)

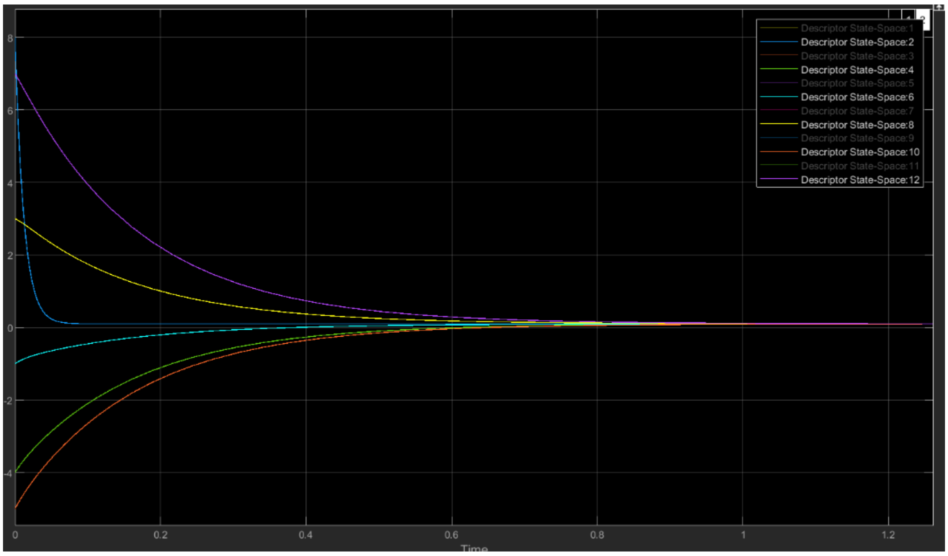


Figure 6.2: $k_{1j} = 16$ for all j ($j \neq 1$) and $k_{ij} = 1$ and ($i \neq 1, i \neq j$), time variation of y ($y_{h_i} \rightarrow 0.097$)

Another interesting example : we consider the case in which the control term k_{i1} ($i \neq 1$) is higher than the other control terms ($k_{ij}, j \neq 1, i \neq j$). The dynamics of robots 2, 3, 4, 5 are fast and the dynamics of robot 1 are slow so the robots converge towards a position close to the initial position of the robot 1. The case is optimal: the dynamics of robot 1 is the only slow dynamics so it does not cause an increase of the time of global convergence of the system. The time of convergence is $t = 0.3\ s$.

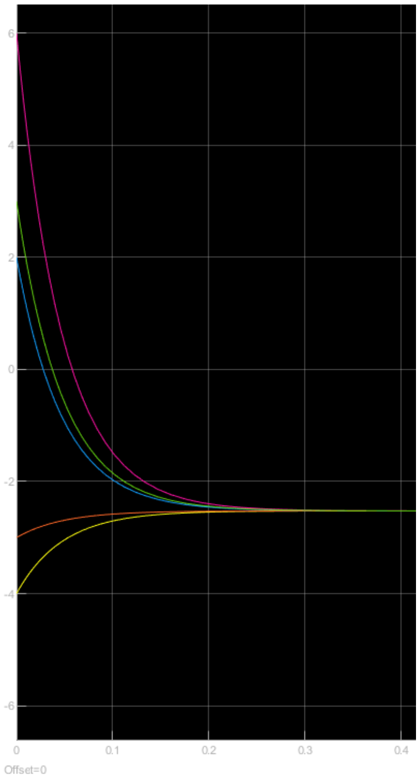


Figure 6.3: $k_{i1} = 16$ for all i ($i \neq 1$) and $k_{ij} = 1$ and ($j \neq 1, i \neq j$), time variation of x ($x_{h_i} \rightarrow -2.524$)

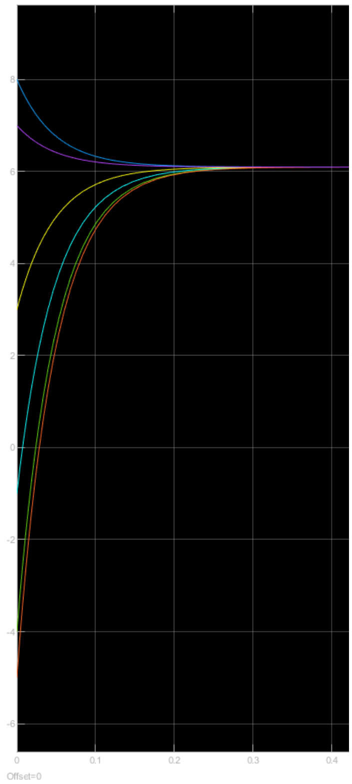


Figure 6.4: $k_{i1} = 16$ for all i ($i \neq 1$) and $k_{ij} = 1$ and ($j \neq 1, i \neq j$), time variation of y ($y_{h_i} \rightarrow 6.095$)

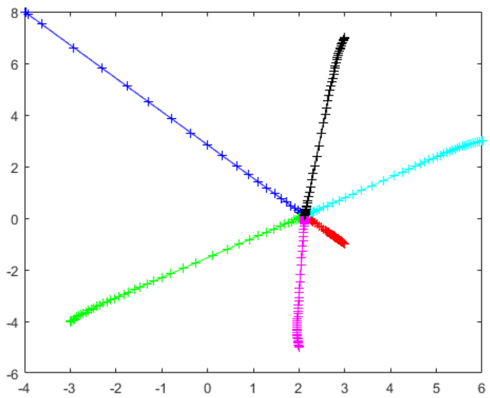


Figure 7.1: $k_{i1} = 16$ for all i ($i \neq 1$) and $k_{ij} = 1$ and ($j \neq 1, i \neq j$), trajectory of the 6 rovers

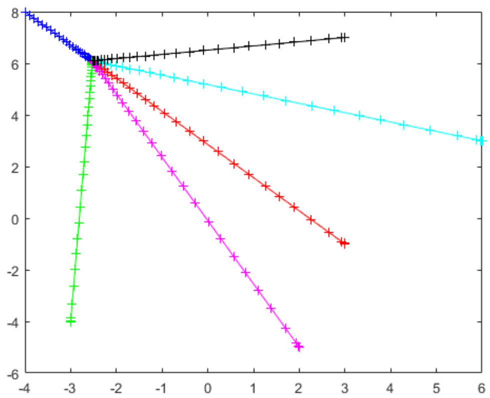


Figure 7.2: $k_{i1} = 16$ for all i ($i \neq 1$) and $k_{ij} = 1$ and ($j \neq 1, i \neq j$), trajectory of the 6 rovers

In conclusion of this part, it is a good idea to modify the control terms in order to have one robot that have different dynamics than the five others, it allows us to diminish the time of convergence of the global system.

- *Propose another simplified graph with less connections, which responds to the rendezvous problem; then adapt the proposed controller and analyze the simulation results. Prove the convergence of the system.*

We want to use the minimum number of connections which allows the robots to meet together at any (x_f, y_f) point. Therefore, each robot must receive information from at least one robot. We choose the following simplified graph:

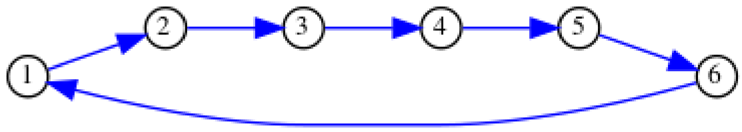


Figure 8: Graph of case 2 (each robot communicates with only one other robot)

In this case, the matrix A is:

$$A = \begin{bmatrix} -k_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{16} & 0 \\ 0 & -k_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{16} \\ k_{21} & 0 & -k_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{21} & 0 & -k_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{32} & 0 & -k_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{32} & 0 & -k_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{43} & 0 & -k_{43} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{43} & 0 & -k_{43} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{54} & 0 & -k_{54} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{54} & 0 & -k_{54} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{65} & 0 & -k_{65} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{65} & 0 & -k_{65} \end{bmatrix}$$

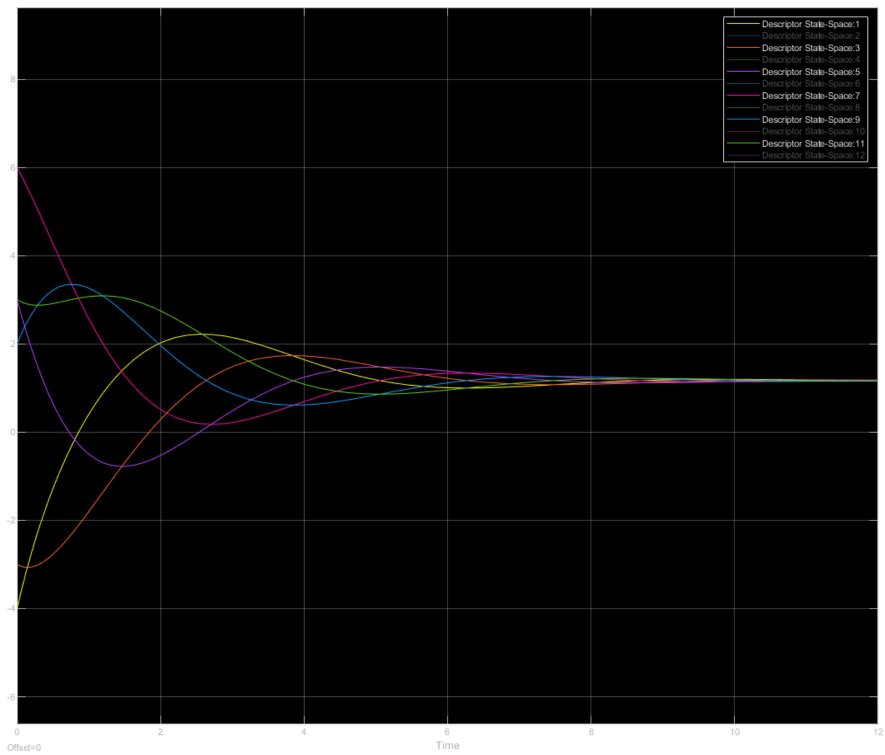


Figure 9.1 : time variation of x $(x_{h_i} \to 1.165$ *when* $t \simeq 15s)$

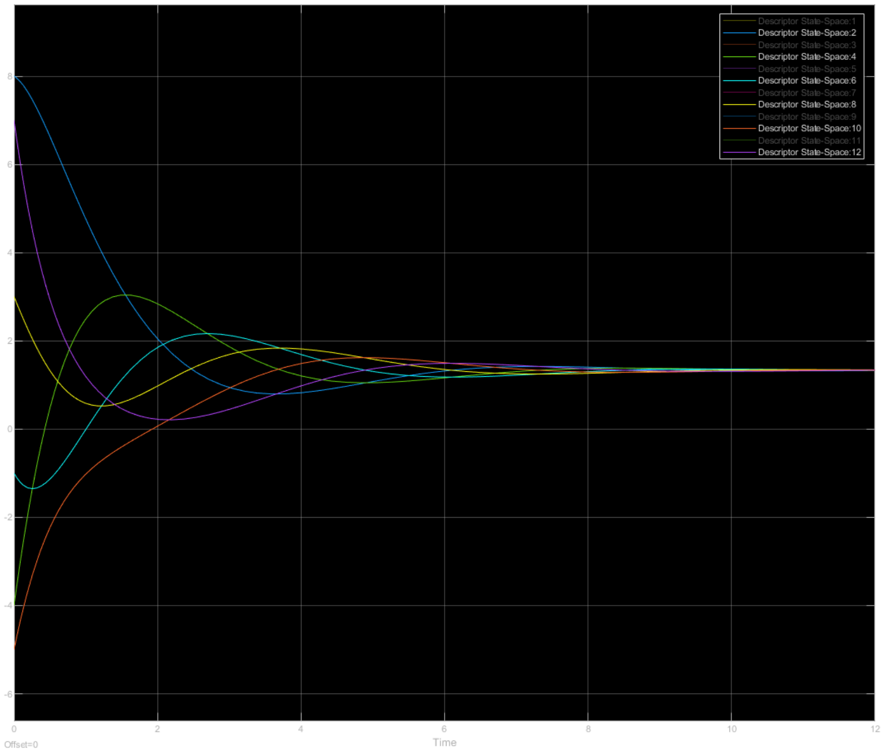


Figure 9.2 : time variation of y ($y_{h_i} \to 1.336$ $when\ t \simeq 15s$)

The time of convergence is very much higher than the case 1 because each robot converges towards its neighbor and not directly towards the meeting point. Like the previous case, we can study the evolution of the time of convergence (criterion of convergence : relative error for each x_{h_i} must be strictly inferior to 1%).

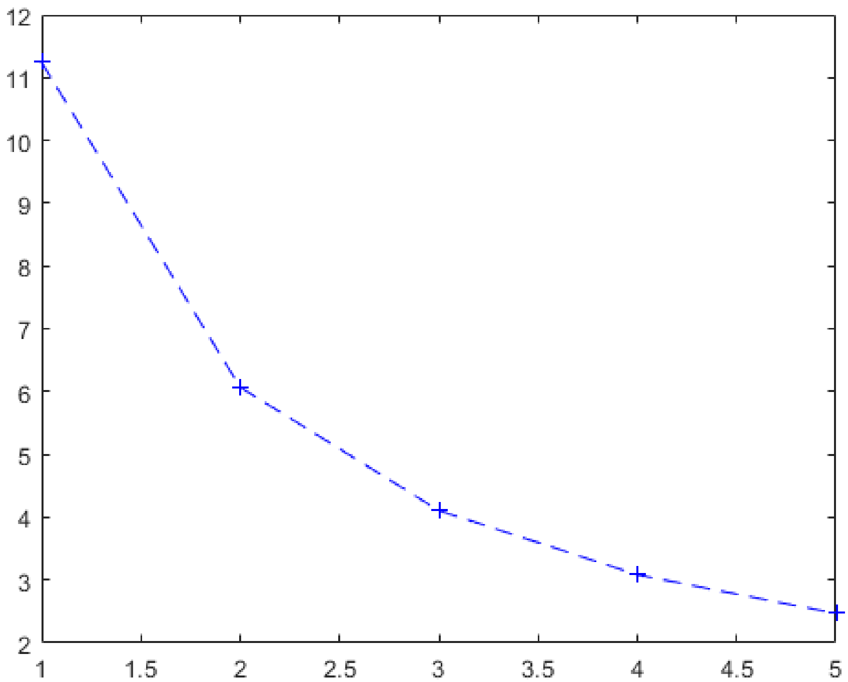


Figure 10 : Time of convergence for $k_{ij} = 1, \dots, 5$

- *<u>Propose a graph and a controller that let the robots meet at the first robot's position.</u>*

We want to force the robots to meet at the first robot's position. Therefore, we deduce that robot one must be the leader. We study the case in which the robot 1 is the leader and does not move (because it is the simplest case for this question). A possible graph is:

Figure 11: Graph of case 3 (each robot communicates with only one other robot, except the robot 1 which is the leader)

In this case, the matrix A is:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_{21} & 0 & -k_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{21} & 0 & -k_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{32} & 0 & -k_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{32} & 0 & -k_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{43} & 0 & -k_{43} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{43} & 0 & -k_{43} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{54} & 0 & -k_{54} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{54} & 0 & -k_{54} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{65} & 0 & -k_{65} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{65} & 0 & -k_{65} \end{bmatrix}$$

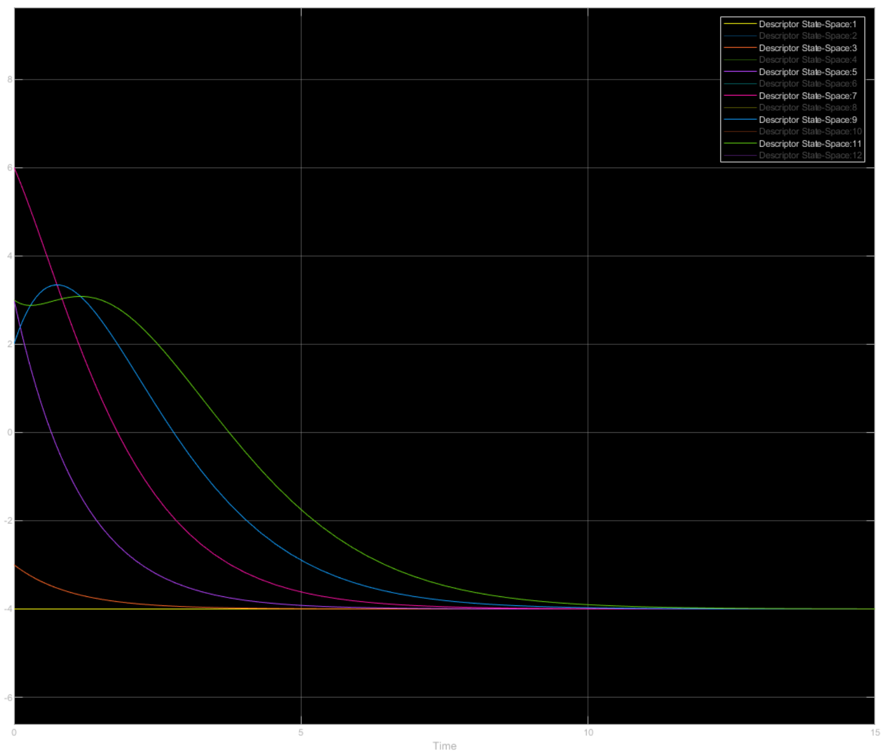


Figure 12.1 : time variation of x ($x_{h_i} \to -4$ when $t \simeq 15s$)

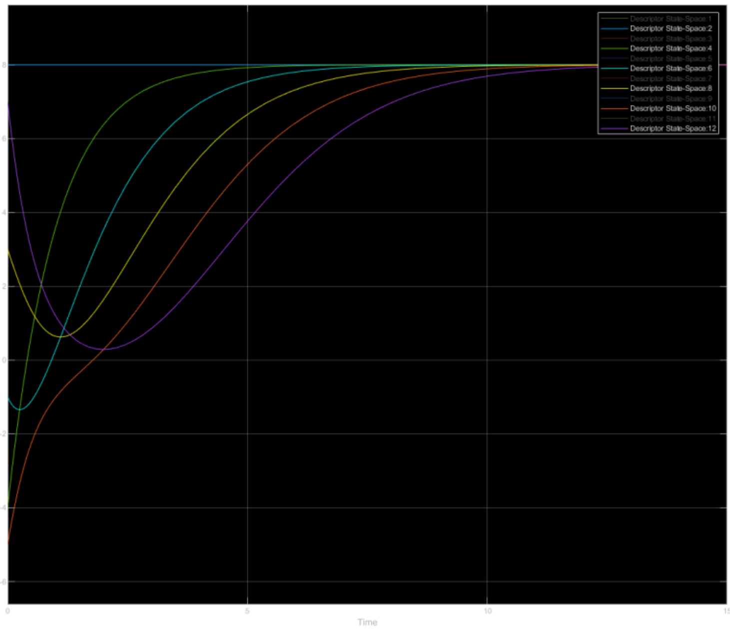


Figure 12.2 : time variation of y ($y_{h_i} \to 8 \text{ when } t \simeq 15s$)

We can study the evolution of the time of convergence (criterion of convergence : relative error for each x_{h_i} must be strictly inferior to 1%). The time of convergence is similar to the one of the case 2 because the graph of communication is approximately the same.

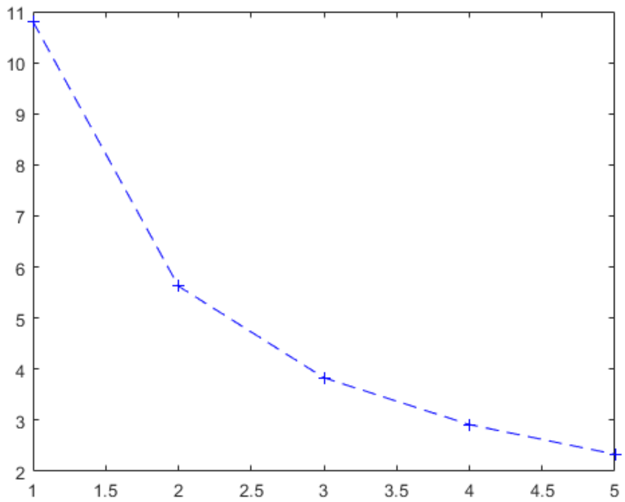
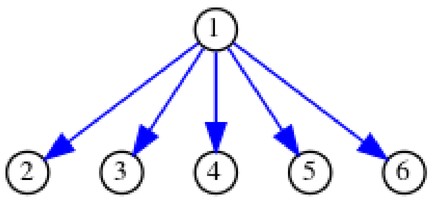


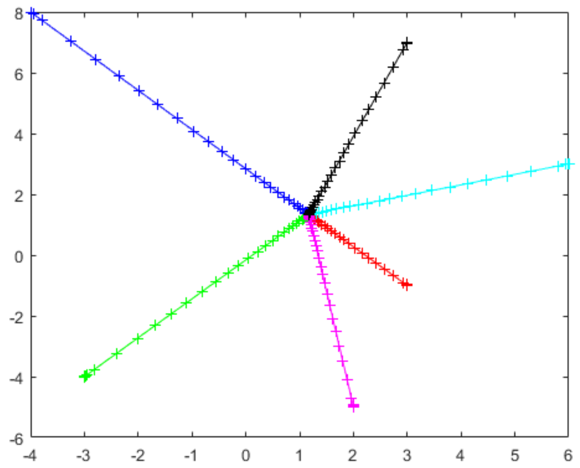
Figure 13 : Time of convergence for $k_{ij} = 1, \dots, 5$

Remark: the following graph (not studied here) allows us to have a less important time of convergence because the information of the leader robot is directly transmitted to the other robots (for example, the robot 3 receives information from the robot 1 without passing through the robot 2).

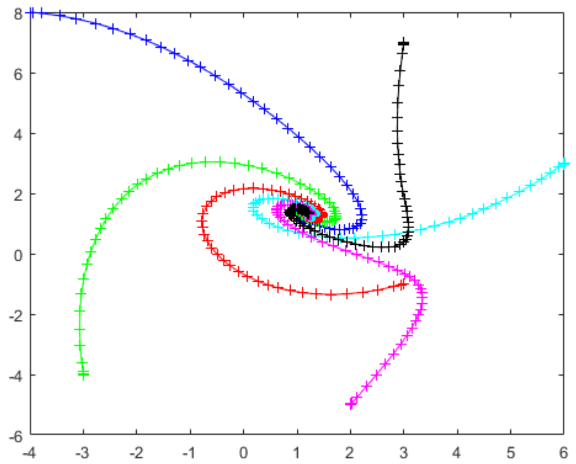


- *For each case, plot the trajectories of the different agents.*

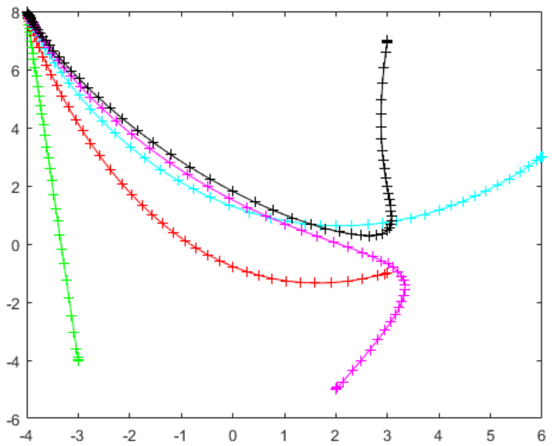
Case 1: all the robots communicate between themselves



Case 2: each robot receives information from only one other robot



Case 3: robot 1 is the leader (robot 1 doesn't receive information from any robot)



- *__Extra-question: Propose a controller that lets the robots meet, then continue to travel together parallel to the 'y-axis' at constant speed.__*

We can express the problem as following:

➢ On the one hand, the robots have to meet together. We can choose the graph of the case 2.

Model of the input:

$$\begin{cases} u_{x_i} = -\sum_{j=1}^{6} g_{ij}k_{ij}\left(x_{h_i} - x_{h_j}\right) \\ u_{y_i} = -\sum_{j=1}^{6} g_{ij}k_{ij}\left(y_{h_i} - y_{h_j}\right) \end{cases}$$

Graph of communication:

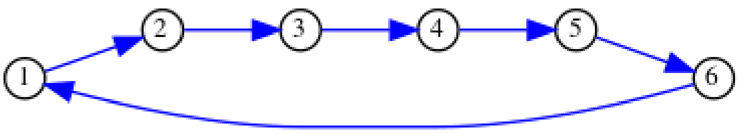


Figure 11: Graph of case 2 (each robot communicates with only one other robot)

In this case, the matrix A is:

$$A = \begin{bmatrix} -k_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{16} & 0 \\ 0 & -k_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{16} \\ k_{21} & 0 & -k_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{21} & 0 & -k_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{32} & 0 & -k_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{32} & 0 & -k_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{43} & 0 & -k_{43} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{43} & 0 & -k_{43} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{54} & 0 & -k_{54} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{54} & 0 & -k_{54} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{65} & 0 & -k_{65} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{65} & 0 & -k_{65} \end{bmatrix}$$

➢ On the other hand, the robots have to travel together in a defined direction, parallel to the y –axis at constant speed, so the robots must follow a trajectory.

Model of the input:

$$\begin{cases} u_{x_i} = \dot{x_r} - (x_{hi} - x_r) \\ u_{y_i} = \dot{y_r} - (y_{hi} - y_r) \end{cases}$$

$$\Rightarrow \begin{cases} \dot{x_{hi}} = -x_{hi} + x_r + \dot{x_r} \\ \dot{y_{hi}} = -y_{hi} + y_r + \dot{y_r} \end{cases}$$

$$\begin{pmatrix} \dot{x_{h1}} \\ \dot{y_{h1}} \end{pmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{pmatrix} x_{h1} \\ y_{h1} \end{pmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} u_{xi} \\ u_{yi} \end{pmatrix}$$

$$\begin{pmatrix} u_{xi} \\ u_{yi} \end{pmatrix} = \begin{pmatrix} x_r(t) + \dot{x_r}(t) \\ y_r(t) + \dot{y_r}(t) \end{pmatrix}$$

The trajectory to be followed must be parallel to the y-axis so $\dot{x_r}(t) = 0\ \forall t$, and the speed must be constant so $\dot{y_r}(t) = cste\ \forall t$. For example, we choose $x_r(t) = 0\ \forall t$ and $y_r(t) = t$ for $t \geq 5s$ and $y_r(t) = 0$ for $0 \leq t < 5s$. We use a Matlab script to generate $x_r(t)$ and $y_r(t)$ (see Annexes Case 4) and we use the signals *'x_r.mat'* and *'y_r.mat'* in the Simulink model.

In conclusion, the global state space representation is: $\dot{X} = AX + Bu$

$$A = \begin{bmatrix} -k_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{16} & 0 \\ 0 & -k_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{16} \\ k_{21} & 0 & -k_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{21} & 0 & -k_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{32} & 0 & -k_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{32} & 0 & -k_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{43} & 0 & -k_{43} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{43} & 0 & -k_{43} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{54} & 0 & -k_{54} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{54} & 0 & -k_{54} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{65} & 0 & -k_{65} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{65} & 0 & -k_{65} \end{bmatrix} \quad u = \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \\ u_{x4} \\ u_{y4} \\ u_{x5} \\ u_{y5} \\ u_{x6} \\ u_{y6} \end{bmatrix}$$

We use the Simulink model below to simulate the behavior of the system.

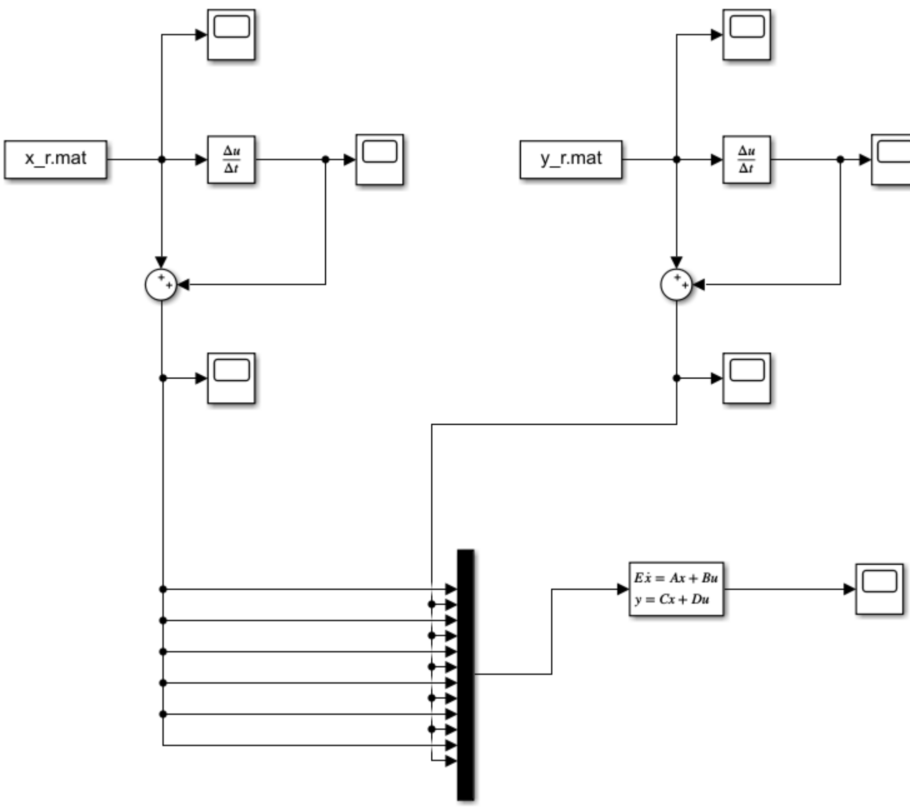


Figure 12: Simulink model of case 4 (case 2 + trajectory parallel to the y-axis at constant speed)

We obtain the following variations of x and y, and the following trajectory :

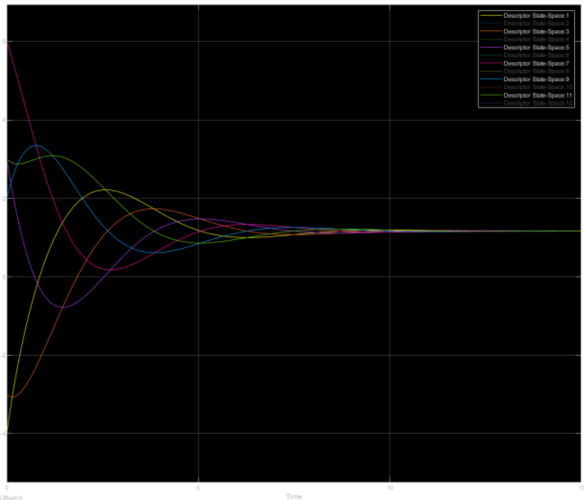


Figure 13.1: time variation of x ($x_{h_i} \to 1.667$ *when* $t \simeq 13s$)

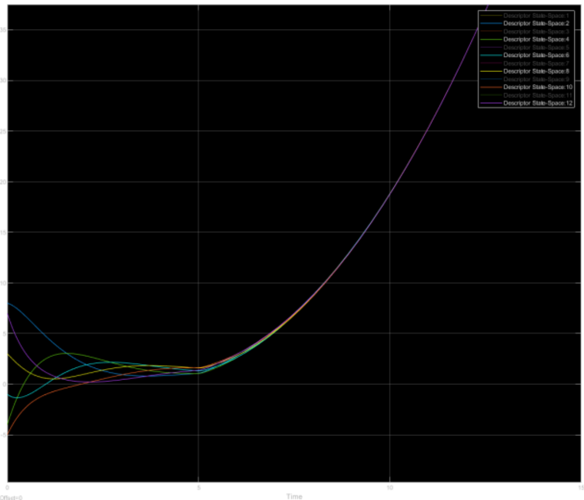


Figure 13.2: time variation of y

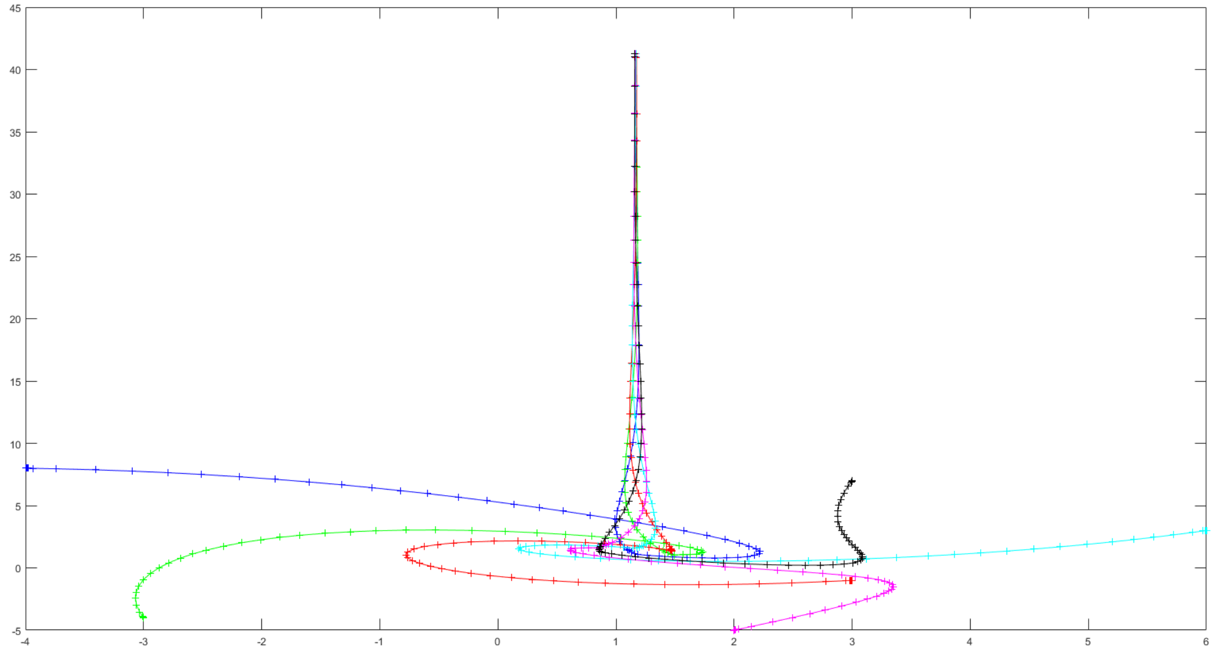


Figure 14: space trajectory of the robots

Remark: We must choose a graph of communication in which each robot receives information from at least one robot. A counterexample is the case 3: if the robot 1 (leader) follows the $(x_r; y_r)$ trajectory, the robots 2-6 will follow it but there will always be a delay between the robot 1 and the group of robots {2, 3, 4, 5,6}.

Annexes: Matlab Code

Case 1: all the robots communicate between themselves

```
%Conditions initiales
I=[-4;8;-3;-4;3;-1;6;3;2;-5;3;7];

%State space representation
B=zeros(12,1);
C=eye(12);
D=zeros(12,1);
E=eye(12);

%Test gain identique pour tous les contrôleurs
k0=1;
k1=16;
k2=1;
A=A_same_gains_2(k0);
%A=A_different_gains(k0,k1,k2);

%Simulation et évolutions des coordonnées au cours du temps
simu1 = sim('Project1_Simulink',2);
v1=simu1.ScopeData.signals.values;
x1=v1(:,1);
y1=v1(:,2);
x2=v1(:,3);
y2=v1(:,4);
x3=v1(:,5);
y3=v1(:,6);
x4=v1(:,7);
y4=v1(:,8);
x5=v1(:,9);
y5=v1(:,10);
x6=v1(:,11);
y6=v1(:,12);
figure(1)
plot(x1,y1,'b+-',x2,y2,'g+-',x3,y3,'r+-',x4,y4,'c+-',x5,y5,'m+-',x6,y6,'k+-
')

% Boucle évolution temps de convergence
eps = 0.01;
k_max = 5;
t = zeros(1,k_max);
k = 1:1:k_max;
for n = k
    A_n = A_same_gains(n);
    simu_n = sim('Project1_Simulink',2);
    v_n = simu_n.ScopeData.signals.values;
    time_n = transpose(simu_n.ScopeData.time);
    L_n = length(time_n);
    for m = 1:1:L_n
        if  (v_n(L_n,1)-v_n(m,1))/v_n(L_n,1)<eps && (v_n(L_n,3)-
v_n(m,3))/v_n(L_n,3)<eps && (v_n(L_n,5)-v_n(m,5))/v_n(L_n,5)<eps &&
(v_n(L_n,7)-v_n(m,7))/v_n(L_n,7)<eps && (v_n(L_n,9)-
v_n(m,9))/v_n(L_n,9)<eps && (v_n(L_n,11)-v_n(m,11))/v_n(L_n,11)<eps
            break
        end
    end
    t(n) = time_n(m);
end
```

```
figure(2)
plot(k,t,'b+--')

function matrix = A_same_gains(k)
  matrix=[-5*k,0,k,0,k,0,k,0,k,0,k,0
     0,-5*k,0,k,0,k,0,k,0,k,0,k
     k,0,-5*k,0,k,0,k,0,k,0,k,0
     0,k,0,-5*k,0,k,0,k,0,k,0,k
     k,0,k,0,-5*k,0,k,0,k,0,k,0
     0,k,0,k,0,-5*k,0,k,0,k,0,k
     k,0,k,0,k,0,-5*k,0,k,0,k,0
     0,k,0,k,0,k,0,-5*k,0,k,0,k
     k,0,k,0,k,0,k,0,-5*k,0,k,0
     0,k,0,k,0,k,0,k,0,-5*k,0,k
     k,0,k,0,k,0,k,0,k,0,-5*k,0
     0,k,0,k,0,k,0,k,0,k,0,-5*k];
end

function matrix = A_different_gains(k,k1,k2)
  matrix=[-5*k1,0,k1,0,k1,0,k1,0,k1,0,k1,0
     0,-5*k1,0,k1,0,k1,0,k1,0,k1,0,k1
     k2,0,-4*k-k2,0,k,0,k,0,k,0,k,0
     0,k2,0,-4*k-k2,0,k,0,k,0,k,0,k
     k2,0,k,0,-4*k-k2,0,k,0,k,0,k,0
     0,k2,0,k,0,-4*k-k2,0,k,0,k,0,k
     k2,0,k,0,k,0,-4*k-k2,0,k,0,k,0
     0,k2,0,k,0,k,0,-4*k-k2,0,k,0,k
     k2,0,k,0,k,0,k,0,-4*k-k2,0,k,0
     0,k2,0,k,0,k,0,k,0,-4*k-k2,0,k
     k2,0,k,0,k,0,k,0,k,0,-4*k-k2,0
     0,k2,0,k,0,k,0,k,0,k,0,-4*k-k2];
end
```

Case 2: each robot receives information from only one other robot

```
function matrix = A_consensus(k)
  matrix=[-k,0,0,0,0,0,0,0,0,0,k,0
    0,-k,0,0,0,0,0,0,0,0,0,k
    k,0,-k,0,0,0,0,0,0,0,0,0
    0,k,0,-k,0,0,0,0,0,0,0,0
    0,0,k,0,-k,0,0,0,0,0,0,0
    0,0,0,k,0,-k,0,0,0,0,0,0
    0,0,0,0,k,0,-k,0,0,0,0,0
    0,0,0,0,0,k,0,-k,0,0,0,0
    0,0,0,0,0,0,k,0,-k,0,0,0
    0,0,0,0,0,0,0,k,0,-k,0,0
    0,0,0,0,0,0,0,0,k,0,-k,0
    0,0,0,0,0,0,0,0,0,k,0,-k];
end
```

Case 3: robot 1 is the leader (robot 1 doesn't receive information from any robot)

```
function matrix = A_leader(k)
  matrix=[0,0,0,0,0,0,0,0,0,0,0,0
    0,0,0,0,0,0,0,0,0,0,0,0
    k,0,-k,0,0,0,0,0,0,0,0,0
    0,k,0,-k,0,0,0,0,0,0,0,0
    0,0,k,0,-k,0,0,0,0,0,0,0
    0,0,0,k,0,-k,0,0,0,0,0,0
    0,0,0,0,k,0,-k,0,0,0,0,0
    0,0,0,0,0,k,0,-k,0,0,0,0
    0,0,0,0,0,0,k,0,-k,0,0,0
    0,0,0,0,0,0,0,k,0,-k,0,0
    0,0,0,0,0,0,0,0,k,0,-k,0
    0,0,0,0,0,0,0,0,0,k,0,-k];
end
```

Case 4: case 2 + trajectory parallel to the y-axis at constant speed

```
%Script generation des signaux x_r et y_r
time = linspace(0,99,100);
x_r = zeros(1,100);
y_r = [zeros(1,5) linspace(0,94,95)];
x_r_time = timeseries(x_r,time);
y_r_time = timeseries(y_r,time);
save('x_r.mat', 'x_r_time', '-v7.3')
save('y_r.mat', 'y_r_time', '-v7.3')


%Prise en compte des entées u1 à u12
B=eye(12);
```