

## **AI34 - TD2**

FISA

Alexis LEBEL, Justin FERDINAND

A24

Table des Matières

<b>1</b>	<b>Consensus-based control for mixing arts and sciences</b>	<b>2</b>
1.1	Pseudo code . . . . .	2
1.2	Student-made consensus . . . . .	2
1.2.1	Explanation of the topologies . . . . .	2
1.2.2	Identification of topologies . . . . .	3
1.3	Artistic motion . . . . .	3
1.4	Using Robotarium . . . . .	8

# 1 Consensus-based control for mixing arts and sciences

## 1.1 Pseudo code

1. Import the necessary libraries and initialize the required variables:
  - Define the number of robots.
  - Create an instance of Robotarium.
  - Define the Laplacian matrix.
2. Main loop:
  - For each iteration (1000):
    - For each agent:
      - \* Retrieve the topological neighbors.
    - For each topological neighbor:
      - \* Calculate the consensus term and add it to the total velocity.
3. Transform the simple-integrator command into a unicycle kinematic command.
4. Update the velocity commands for each robot.
5. Apply the velocity commands.

## 1.2 Student-made consensus

### 1.2.1 Explanation of the topologies

#### First topology :

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

#### Second topology :

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix} \quad L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix}$$

### 1.2.2 Identification of topologies

#### Topology of Fig. 3 :

This topology does not resemble either a cyclic case or a fully connected case. This topology does not seem suitable for our configurations. Furthermore, the figure shows a target offset. However, no offset is represented in the figures above.

#### Topology of Fig. 4 :

We observe spiral movements, which indicate slower convergence and therefore reduced communication between the mobile units. This resembles the cyclic topology as shown in figure 2b.

#### Topology of Fig. 5 :

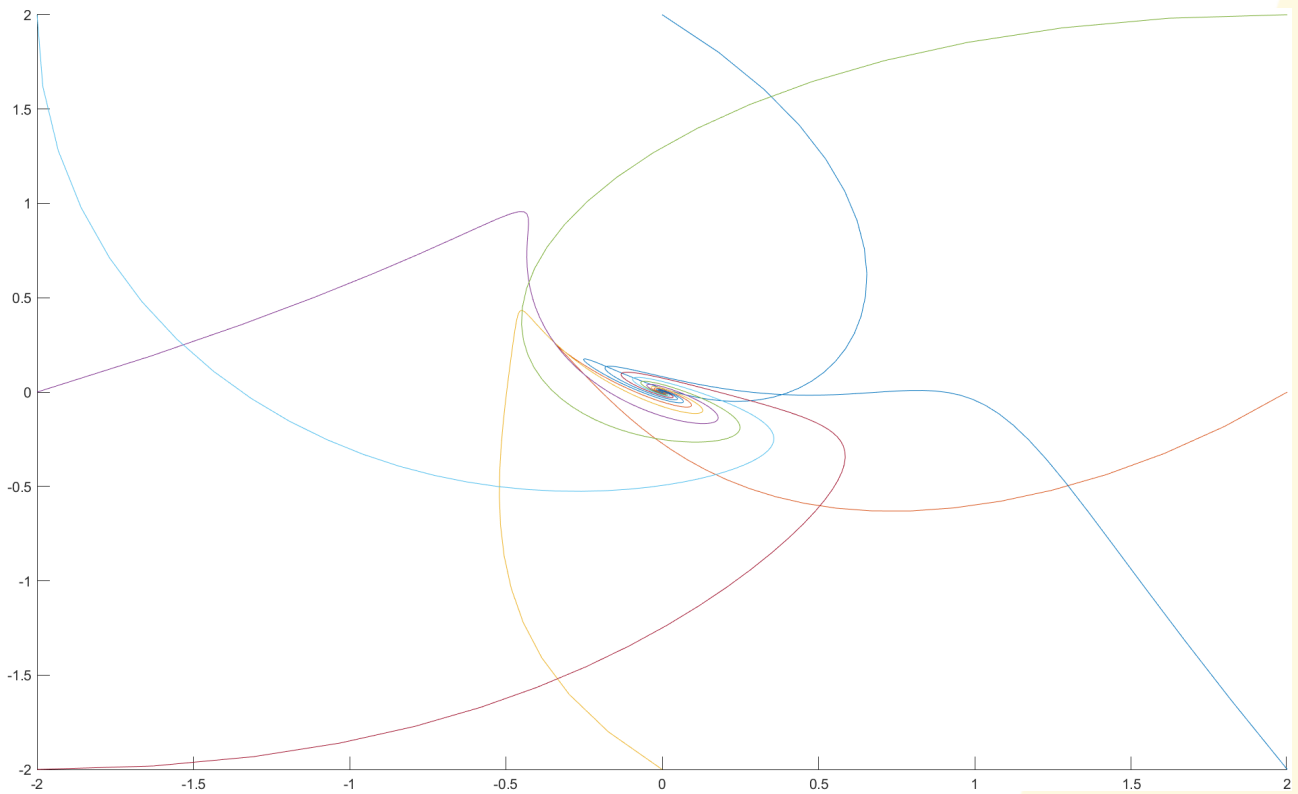
All robots converge directly to a central position, without complex movements, indicating that all robots communicate with each other. The trajectory is therefore adjusted towards the central point. This is the topology of figure 2a.

### 1.3 Artistic motion

Different artistic motion patterns can be obtained by tweaking the following parameters :

- Different MRS Graphs
- Different Convergence gains
- Different Starting positions

The following figures show the different artistic motion patterns obtained, and their associated parameters.



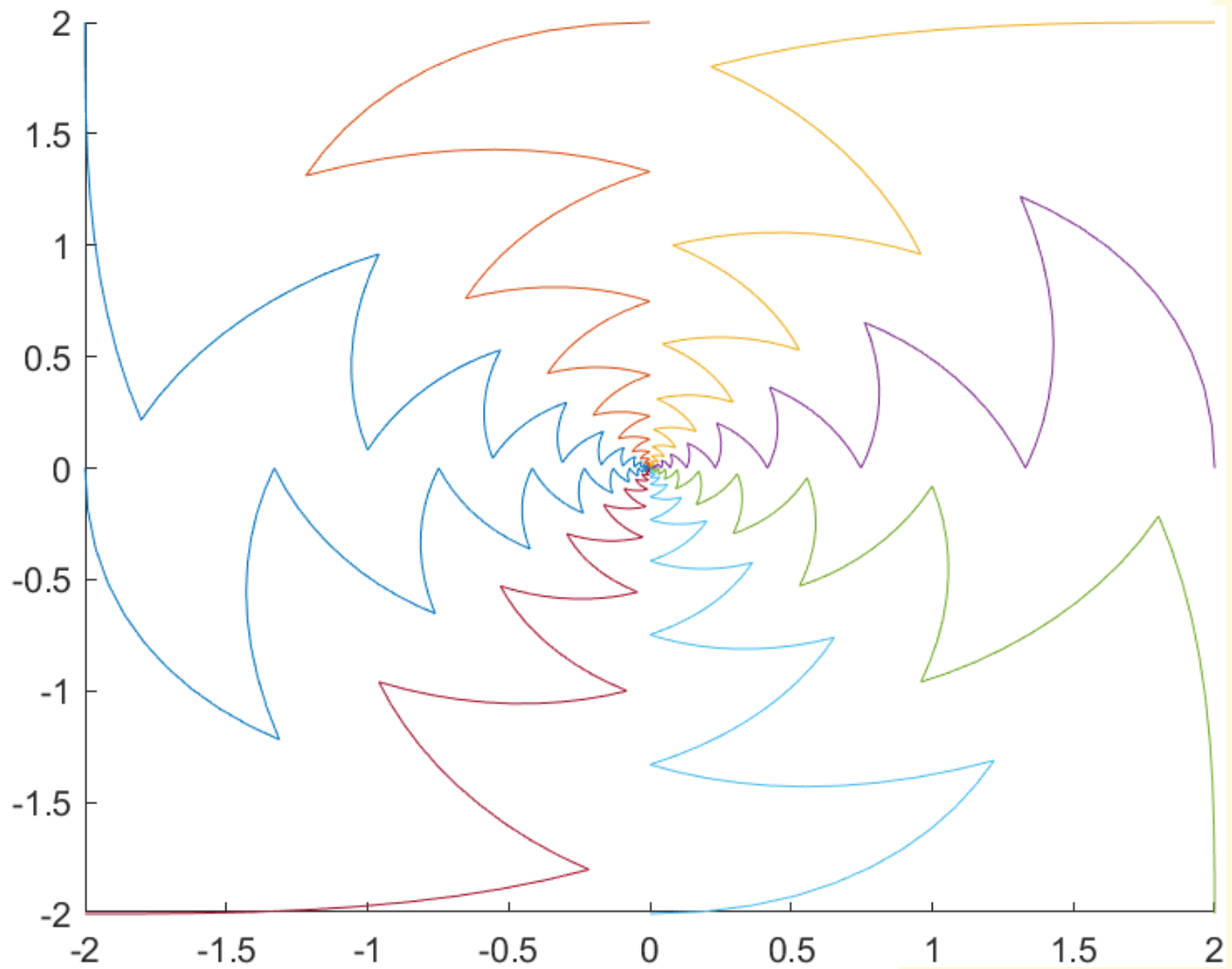
**Figure 1:** First artistic pattern

### Parameters :

Laplacian :

```
% Positions
X=[-2 0 2 2 2 0 -2 -2]';
Y=[2 2 2 0 -2 -2 -2 0]';
L2 = eye(N) - [
    0 1 0 0 0 0 0 0; % Sommet 1 pointe vers 2
    0 0 1 0 0 0 0 0; % Sommet 2 pointe vers 3
    0 0 0 1 0 0 0 0; % Sommet 3 pointe vers 8
    0 0 0 0 1 0 0 0; % Sommet 4 pointe vers 5
    0 0 0 0 0 1 0 0; % Sommet 5 pointe vers 6
    0 0 0 0 0 0 1 0; % Sommet 6 pointe vers 7
    0 0 0 0 0 0 0 1; % Sommet 7 pointe vers 8
    1 0 0 0 0 0 0 0; % Sommet 8 pointe vers 1
];
```

To obtain this pattern, the agents are placed at each corner and center of a square, but their position is switched 2 by 2. The convergence is achieved, but the interactions are different, giving this pattern.



**Figure 2:** Second artistic pattern

**Parameters :**

```

M = 10000;
Xt = zeros(8, M);
Yt = zeros(8, M);
Temps = zeros(1, M);

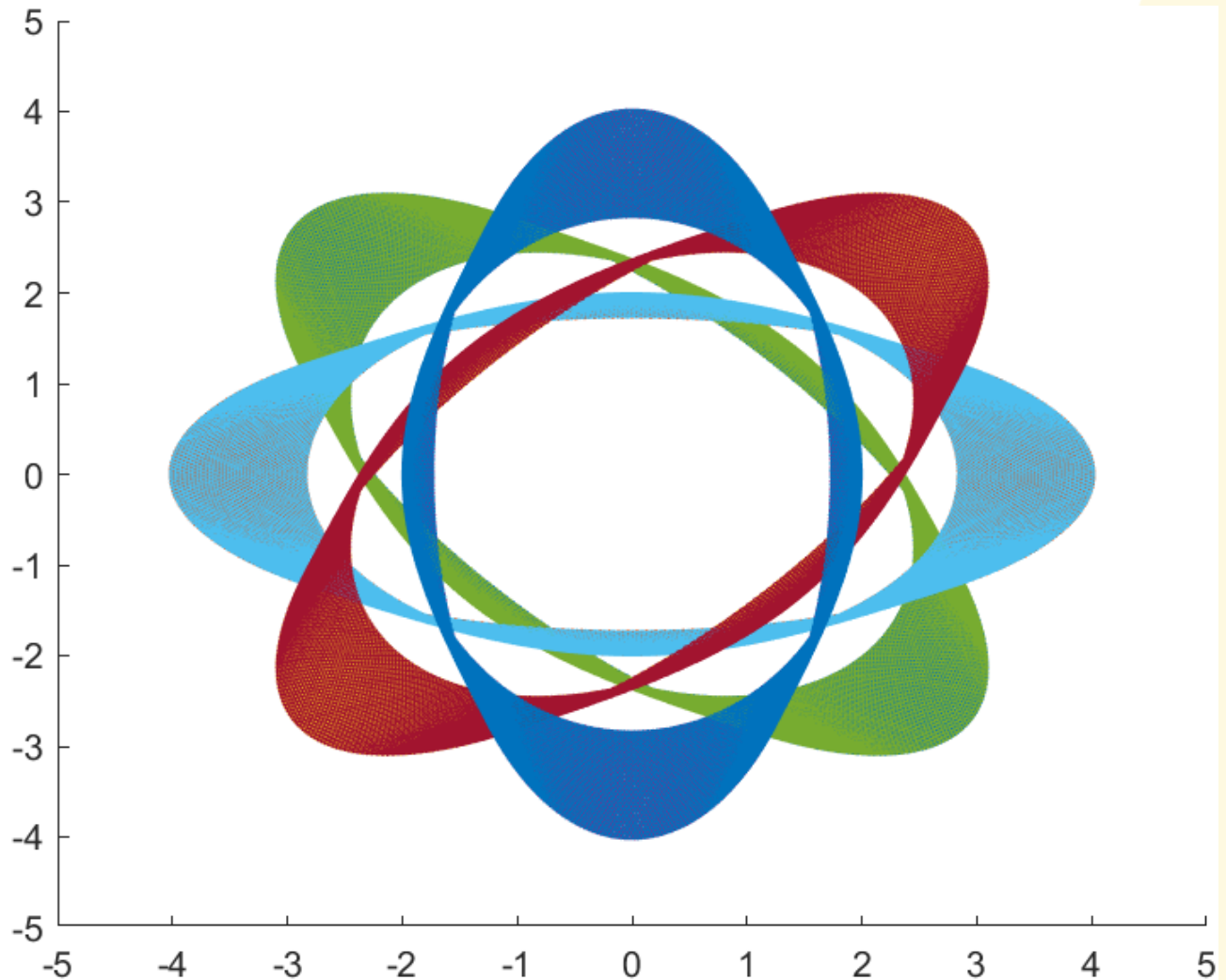
for k = 1:M
    Xt(:, k) = X;
    Yt(:, k) = Y;
    Temps(:, k) = k*Te;
    X = Ae*X;
    Y = Ae*Y;
    if mod(k, 15) == 0
        if A == -K * L2_anti_trigo
            A = -K * L2_trigo;
        else
            A = -K * L2_anti_trigo;
        end
        [Ae, Be] = c2d(A, B, Te);
    end
end

```

This pattern uses switched laplacian matrixes every 15 iterations. The agents are placed in a circle, in a cycle

topology. The convergence is achieved, alternating between trigonometric rotation and anti-trigonometric rotation.

By just changing the gain (divergent anti-trigonometric and convergent trigonometric), we can obtain a different pattern.



**Figure 3:** Third artistic pattern

```
M = 10000;
Xt = zeros(8, M);
Yt = zeros(8, M);
Temps = zeros(1, M);

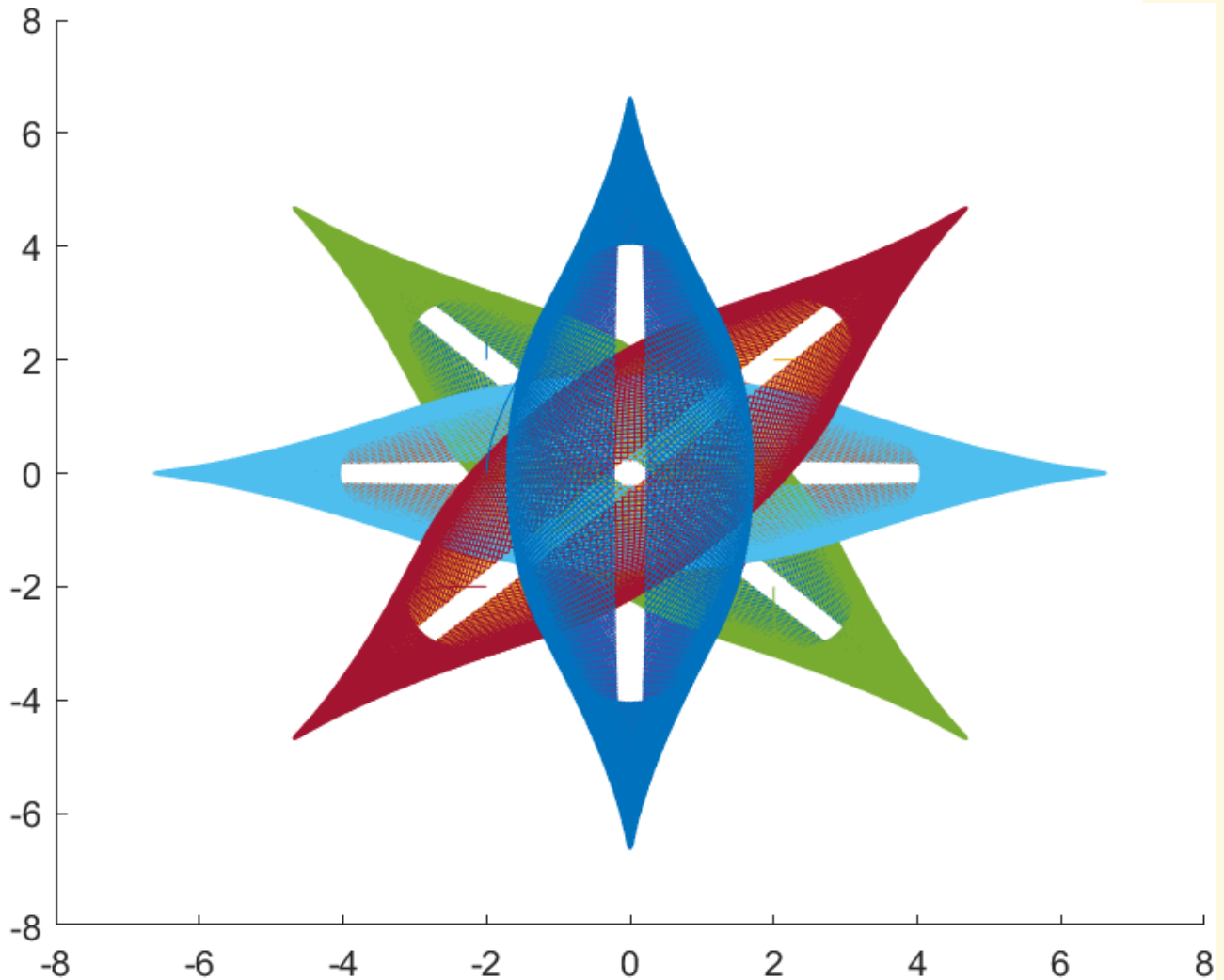
for k = 1:M
    Xt(:, k) = X;
    Yt(:, k) = Y;
    Temps(:, k) = k*Te;
    X = Ae*X;
    Y = Ae*Y;
    if mod(k, 15) == 0
        if A == -K * L2_anti_trigo
            A = K * L2_trigo;
        else
            A = -K * L2_anti_trigo;
        end
    end
end
```

```

    [Ae, Be] = c2d(A, B, Te);
end
end

```

And by changing the modulo of the iteration, we can obtain another a different pattern.



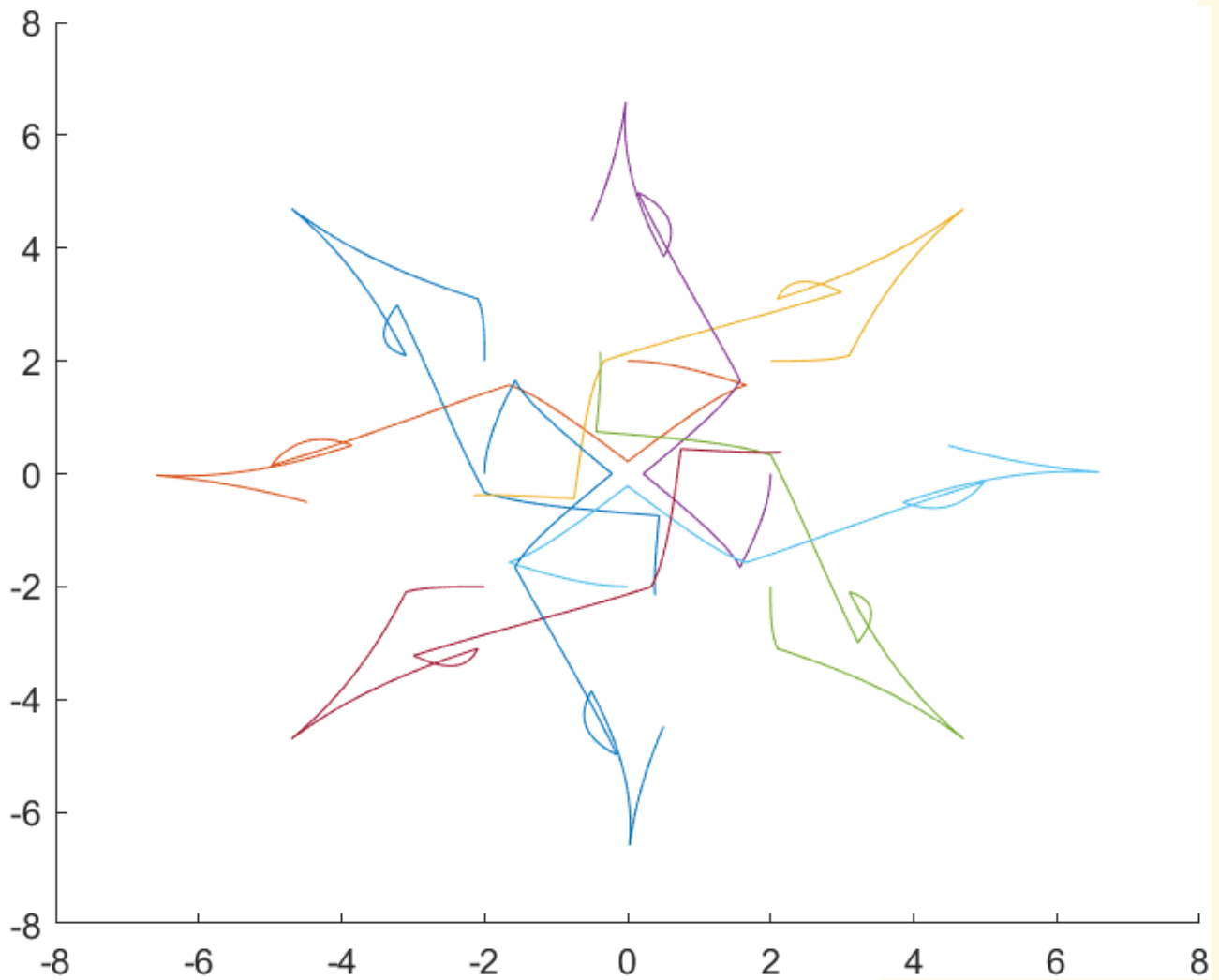
**Figure 4:** Fourth artistic pattern

```

for k = 1:M
    Xt(:, k) = X;
    Yt(:, k) = Y;
    Temps(:, k) = k*Te;
    X = Ae*X;
    Y = Ae*Y;
    if mod(k, 15) == 0
        if A == K * L2_anti_trigo
            A = -K * L2_trigo;
        else
            A = K * L2_anti_trigo;
        end
        [Ae, Be] = c2d(A, B, Te);
    end
end
end

```





**Figure 5:** With fewer iterations to see the movement

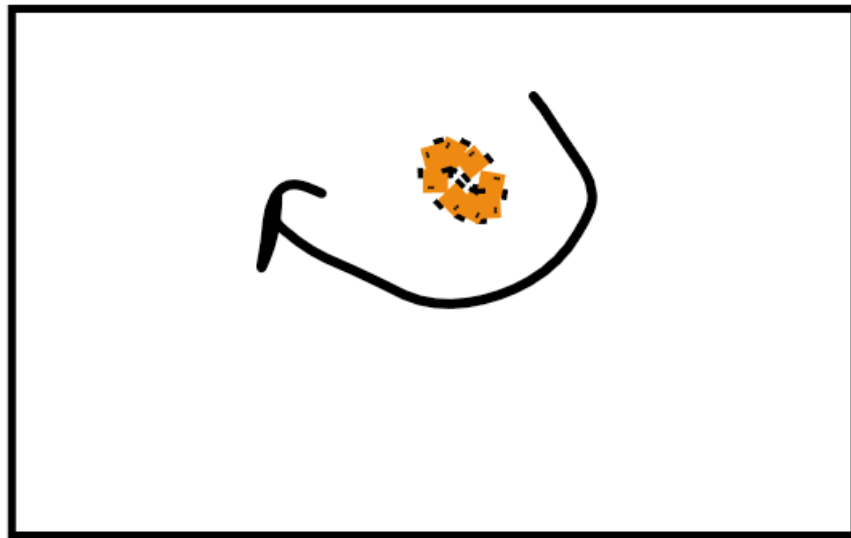
### 1.4 Using Robotarium

The important thing to note here, is to update the SI velocities :

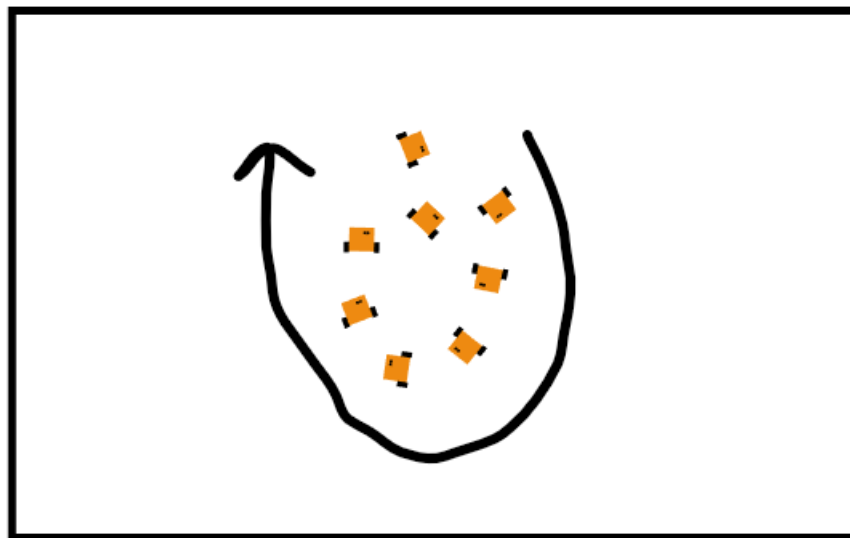
```
dx_i = [Ae*Xt(:, t) - Xt(:, t), Ae*Yt(:, t) - Yt(:, t)]';
```

And to use the `si_to_uni` function to convert the SI velocities to unicycle velocities :

```
dx_u = si_to_uni_dyn(dx_i, x);
```



**Figure 6:** Movement without barriers



**Figure 7:** Movement using barriers

To enhance the trajectory of the agents, we can use a more cognitive approach, as this approach can be considered reactive (as the agents are avoiding collisions and eventually following a “leader”) That cognitive approach can be achieved by using limit cycles where we consider the path of the agents as an obstacle, and move only a limited set of agents.

## Conclusion

In this lab, we have seen how to use consensus-based control to achieve different artistic patterns. We have seen how to use different topologies to achieve different patterns, and how to use the Robotarium to simulate the agents' movements.

Understanding how Robotarium works is essential for the upcoming project.