Control of Multi-Agent Systems
EEL-3040: Control Systems

A Lab Report submitted by

Kharane Anand Madhukar
B23EE1035

Indian Institute of Technology Jodhpur
Department of Electrical Engineering

Control Multi-Agent Systems

Content

1) Abstract :-

This report presents a simulation-based study on the consensus problem in multi-agent systems. The experiment involves a group of six autonomous robots, each with a distinct initial position, velocity, and direction. The core objective is to achieve consensus in the velocity directions of all robots using a distributed control protocol. The study investigates the impact of network topology and controller gain on the system's convergence to a consensus state.
Two primary communication topologies, a fully connected network and a sparser, decentralized network are simulated and analyzed. Furthermore, the effect of varying the gain K is examined, demonstrating its influence on the speed of convergence. The final part of the experiment explores a modified control law with individual agent gains, showing that the final consensus value can be manipulated by appropriately tuning these gains. The results confirm that network connectivity is crucial for reaching consensus and that the controller gain directly affects the rate of agreement among the agents.

2) Objective :-

a) Aims :-

- To understand and implement a distributed consensus protocol for a multi-agent system of mobile robots.
- To simulate the system's behavior using MATLAB and analyze the trajectories, velocity directions, and control inputs of the agents.
- To investigate the effect of different communication topologies on the system's ability to reach consensus.
- To study the impact of the controller gain K on the rate of convergence to the consensus state.
- To explore how individual controller gains can be used to achieve consensus at a pre-specified value

Introduction :-

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents.These agents are autonomous entities that can perceive their environment and act upon it to achieve their goals. In many applications, from robotic swarms to sensor networks, it is desirable for the agents to reach an agreement on a certain quantity of interest. This problem is known as the consensus problem. This experiment focuses on achieving consensus on the velocity direction for a group of six mobile robots moving in a 2D plane .

Theory :-

The motion of each robot i is described by the following kinematic model : -

$$\begin{aligned} \dot{x}_i(t) &= v_i \cos\theta_i(t) \\ \dot{y}_i(t) &= v_i \sin\theta_i(t) \\ \dot{\theta}_i(t) &= u_i, \qquad i = 1, \ldots, N, \end{aligned}$$

where (xi,yi) are the robot's positional coordinates, viis its constant speed, θiis its velocity direction, and uiis the control input.

To achieve consensus on the velocity directions θi, a distributed control law, often called an agreement protocol, is used. The control input for each robot is based on information from its neighbors in the communication network

$$u_i = K \sum_{j \in N(i)} (\theta_j(t) - \theta_i(t)), \quad i = 1, \ldots, N,$$

Here, N(i) is the set of neighbors of robot i, and K>0 is a positive controller gain. This control law drives the velocity direction of each robot towards the average direction of its neighbors. The structure of the communication network, or topology, is fundamental to the system's behavior. The connectivity of the network graph determines whether a consensus can be reached. For a connected graph, all agents will eventually converge to a common velocity direction, which is the average of their initial directions. The rate of this convergence is influenced by both the controller gain K and the algebraic connectivity of the graph .

Experimental Procedure :-

The experiment was conducted entirely through simulation in MATLAB. A system of six robots was modeled with the initial conditions specified in the lab manual. The simulation involved implementing the kinematic equations and the distributed control law for each robot. The procedure was divided into tasks as outlined below :-

1. System Initialization:- The initial positions (xi(0),yi(0)), initial velocity directions θi(0), and constant speeds vifor the six robots were defined as per the provided table.

2. Q1(a) - Fully Connected Topology: - The control protocol was simulated for the fully connected network topology shown in Figure 1(a) of the lab manual. In this topology, every robot is a neighbor to every other robot. The simulation was run for three different controller gains: K = 0.01, K = 0.1, and K = 1.

3. Q1(b) - Sparse Topology:- The simulation was repeated for the sparse network topology in Figure 1(b). This was performed for two controller gains, K = 0.01 and K = 1, to compare its performance against the fully connected topology.

4. Q1(d) - Modified Sparse Topology: - The topology from Q1(b) was modified by removing the communication link between robot 2 and robot 5. The simulation was run again for K = 0.01 and K = 1 to observe the impact of reduced connectivity.

5. Q2 - Individual Controller Gains:- The control law was modified to allow for individual gains K_i for each robot. The system was simulated using the topology from Figure 1(b) to analyze the effect on the consensus value and demonstrate the ability to steer the consensus to a desired direction.

For each simulation, the trajectories of the robots, the evolution of their velocity directions (θi) over time, and their control inputs (ui) over time were recorded for subsequent analysis.

Data & Calculations :-

This section presents the codes and results from the MATLAB simulations

Q1(a): Simulation with Fully Connected Topology (Figure 1a) :-

The system was simulated using the below script for three different values of the controller gain K.

Code :-

```
close all;
clear all;
clc;

tic

%% number of robots

N=6;

%% Initialize

% first robot
x(1)     =  -10;           % x-position
y(1)     =  -2;            % y-position
theta(1) =   0*pi/180;     % velocity direction (converted from degrees to radian)
v(1) = 1.0;                % speed


% second robot
x(2)     =  4;
y(2)     = -2;
theta(2) =  30*pi/180;
v(2) = 1.5;


% third robot
x(3)     =  -10;
y(3)     = 10;
theta(3) =  45*pi/180;
v(3) = 2.0;

% fourth robot
x(4)     =  2;
y(4)     = 5;
theta(4) =  60*pi/180;
v(4) = 2.5;

% fifth robot
x(5)     =  0;
y(5)     = -5;
theta(5) =  75*pi/180;
v(5) = 3.0;

% sizth robot
x(6)     =  5;
y(6)     = -7;
theta(6) =  90*pi/180;
v(6) = 3.5;

%% controller gain

K = 0.01;

%% simulation time

T  = 100;          % total time
dt = 0.01;         % step size


%% store variables for offline plot
iStep = 1;
% xStore  = zeros(round(T/dt),N);
% yStore  = zeros(round(T/dt),N);
% thetaStore = zeros(round(T/dt),N);
% uStore  = zeros(round(T/dt),N);
% time = zeros(round(T/dt),1);
```

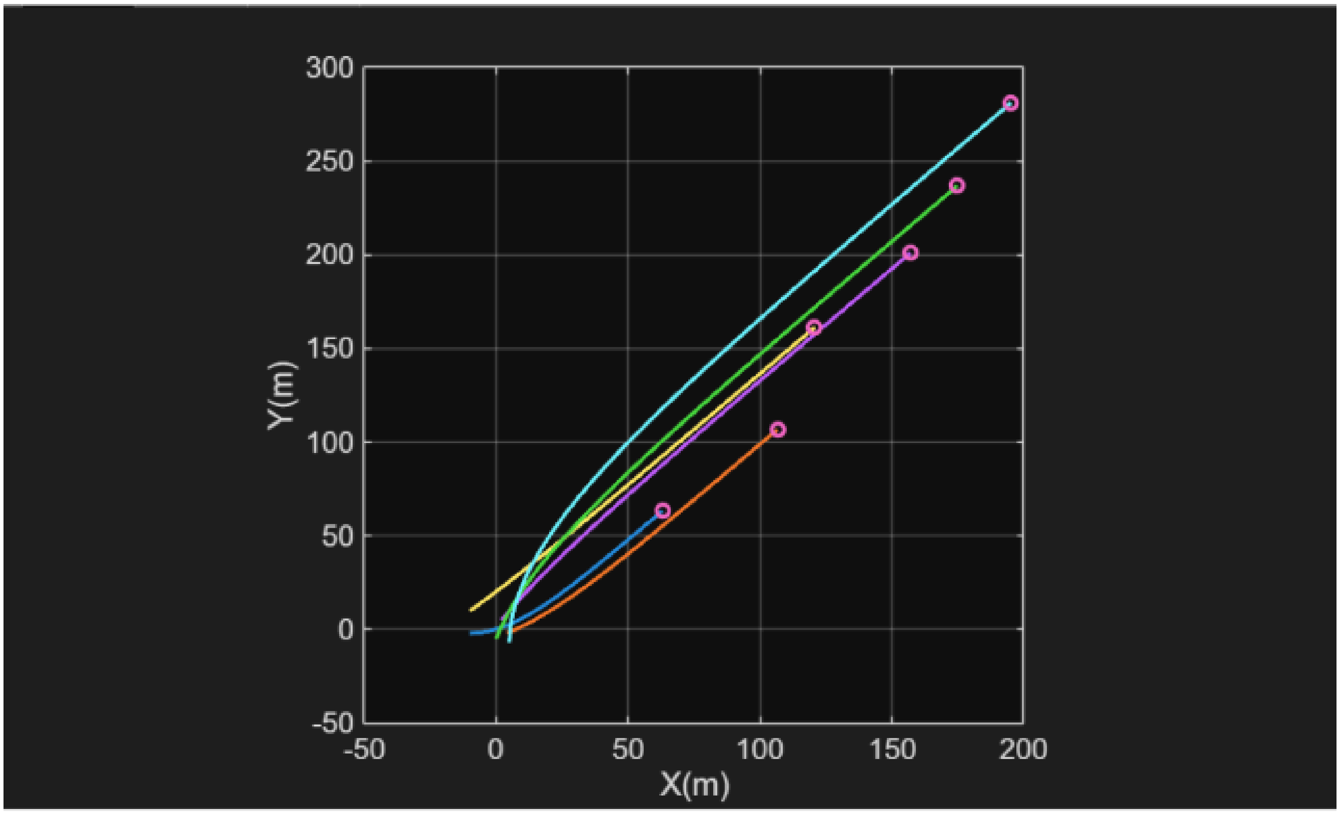
```
%% Control Loop

for t = 0:dt:T

%% control law

  u(1) = K*(theta(2) + theta(3) + theta(4) + theta(5) + theta(6) - 5*theta(1)); % control law for the first robot
  u(2) = K*(theta(1) + theta(3) + theta(4) + theta(5) + theta(6) - 5*theta(2)); % control law for the second robot
  u(3) = K*(theta(1) + theta(2) + theta(4) + theta(5) + theta(6) - 5*theta(3)); % control law for the third robot
  u(4) = K*(theta(1) + theta(2) + theta(3) + theta(5) + theta(6) - 5*theta(4)); % control law for the fourth robot
  u(5) = K*(theta(1) + theta(2) + theta(3) + theta(4) + theta(6) - 5*theta(5)); % control law for the fifth robot
  u(6) = K*(theta(1) + theta(2) + theta(3) + theta(4) + theta(5) - 5*theta(6)); % control law for the sixth robot

%% store for offline plotting
xStore(iStep,:) = x;
yStore(iStep,:) = y;
thetaStore(iStep,:) = theta;
uStore(iStep,:) = u;
time(iStep,:) = t;
iStep = iStep + 1;

```

```
%% update

x = x + v.*cos(theta)*dt;
y = y + v.*sin(theta)*dt;
theta = theta + u*dt;  % Implmentation using Euler's method in matrix notation

end

% plotting

%% trajectories of robots

figure(1)
plot(xStore,yStore,'LineWidth',2); hold on
plot(xStore(end,:),yStore(end,:),'o','LineWidth',2); hold on
set(gca,'fontsize',14,'Fontname','Helvetica');
xlabel('X(m)');
ylabel('Y(m)');
grid on
axis equal
axis square

```

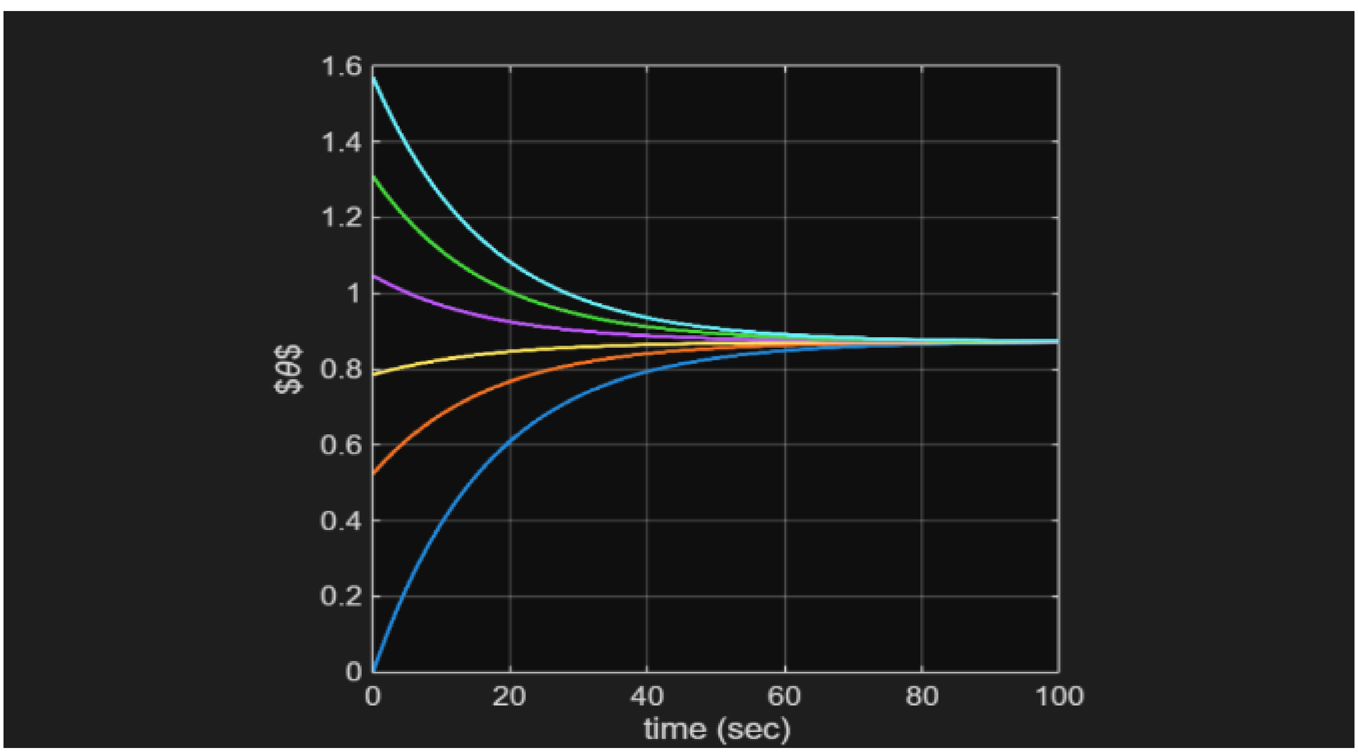
```
 %% velocity directions

figure(2)
plot(time(1:end,:), thetaStore(1:end,:), 'LineWidth',2); hold on
set(gca,'fontsize',14,'Fontname','Helvetica');
xlabel('time (sec)');
ylabel('$\theta$');
grid on
axis equal
axis square

 %% control inputs

figure(3)
plot(time(1:end,:), uStore(1:end,:), 'LineWidth',2); hold on
set(gca,'fontsize',14,'Fontname','Helvetica');
xlabel('time (sec)');
ylabel('u');
grid on
axis equal
axis square

```

Result for k = 0.01 :-

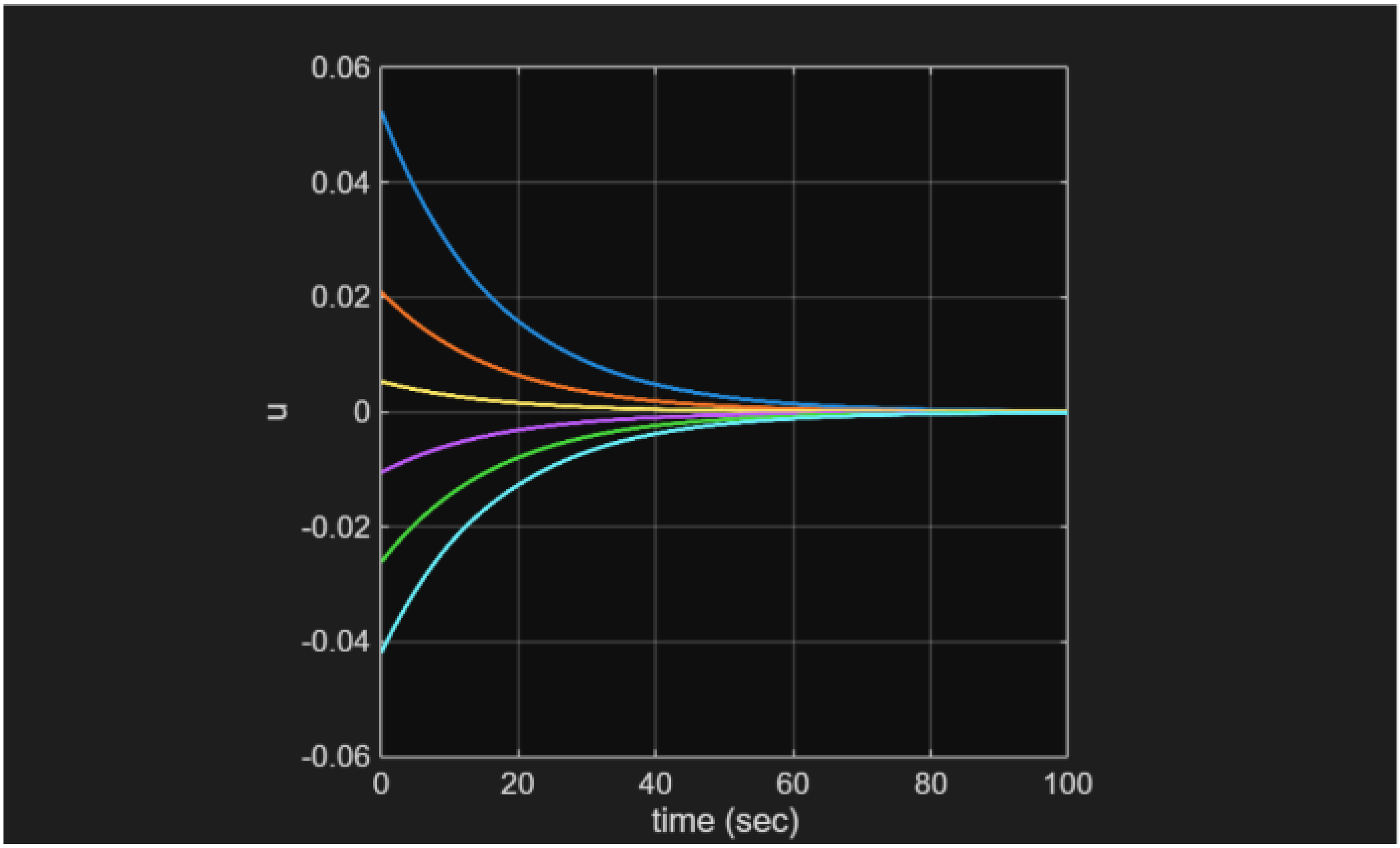
GRAPH 1:- Trajectories of Robots (K=0.01)



Graph 2 :- Velocity Direction vs Time (k = 0.01)

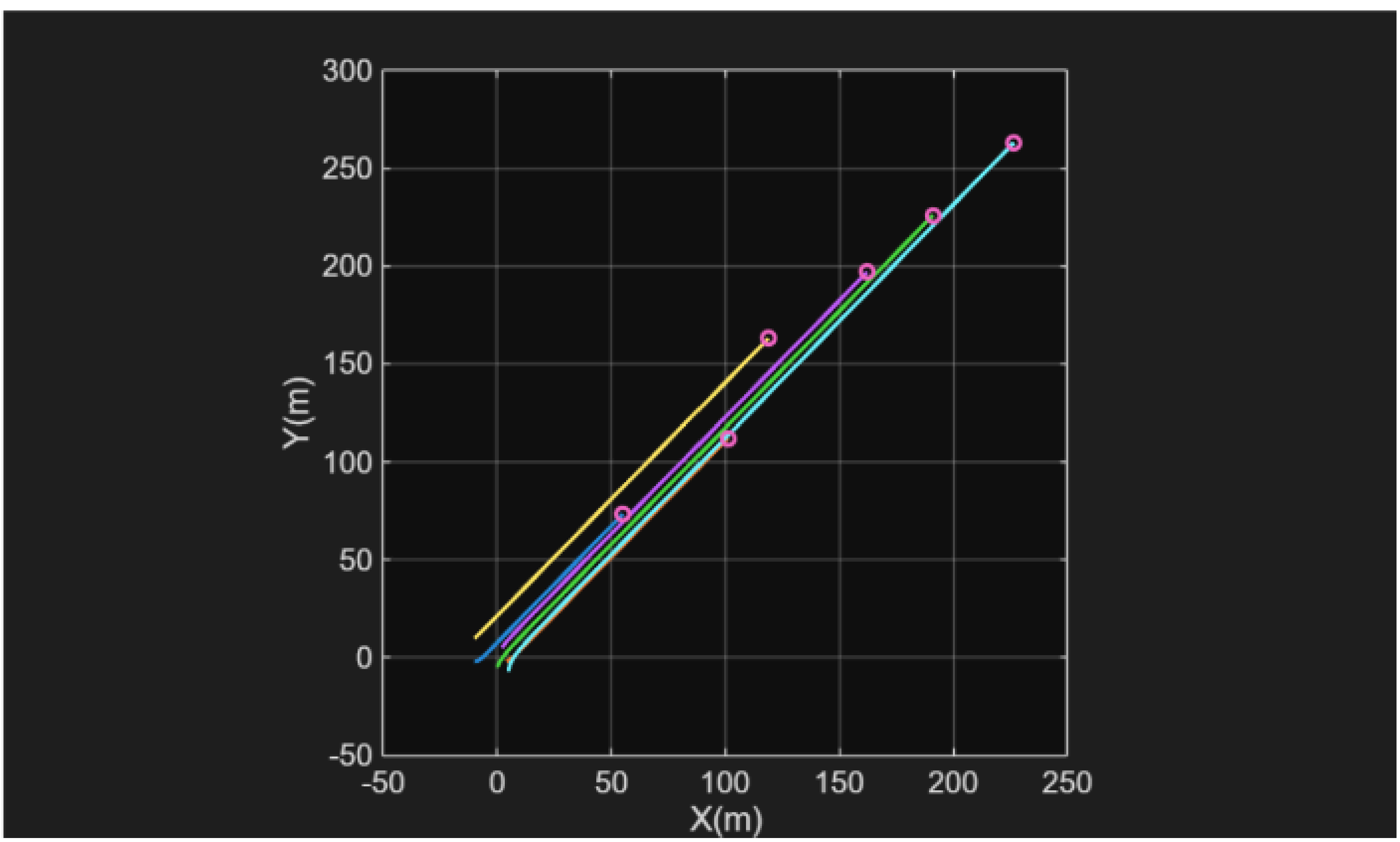


Graph 3 :- Control input vs Time (k=0.01)

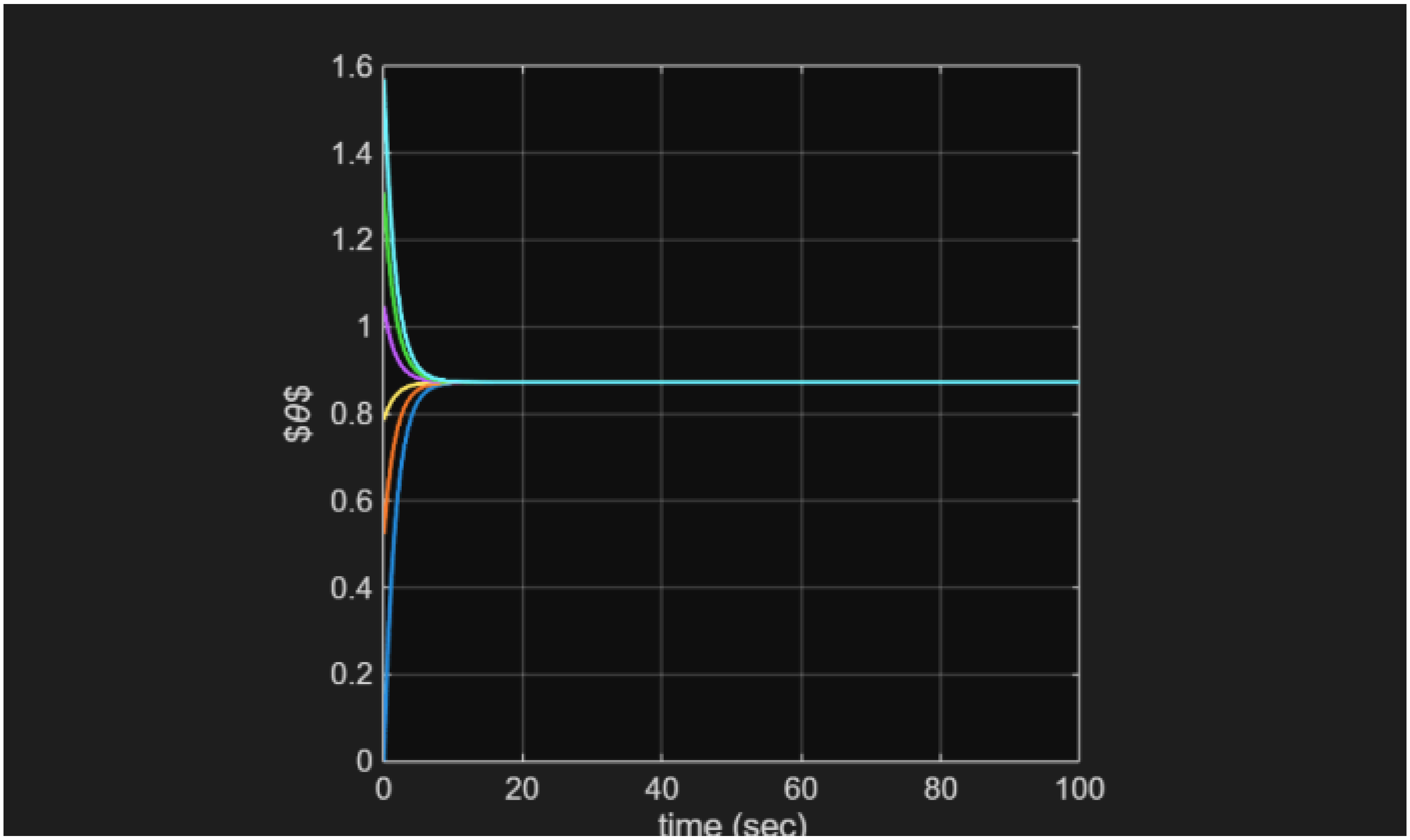


Result for k = 0.1

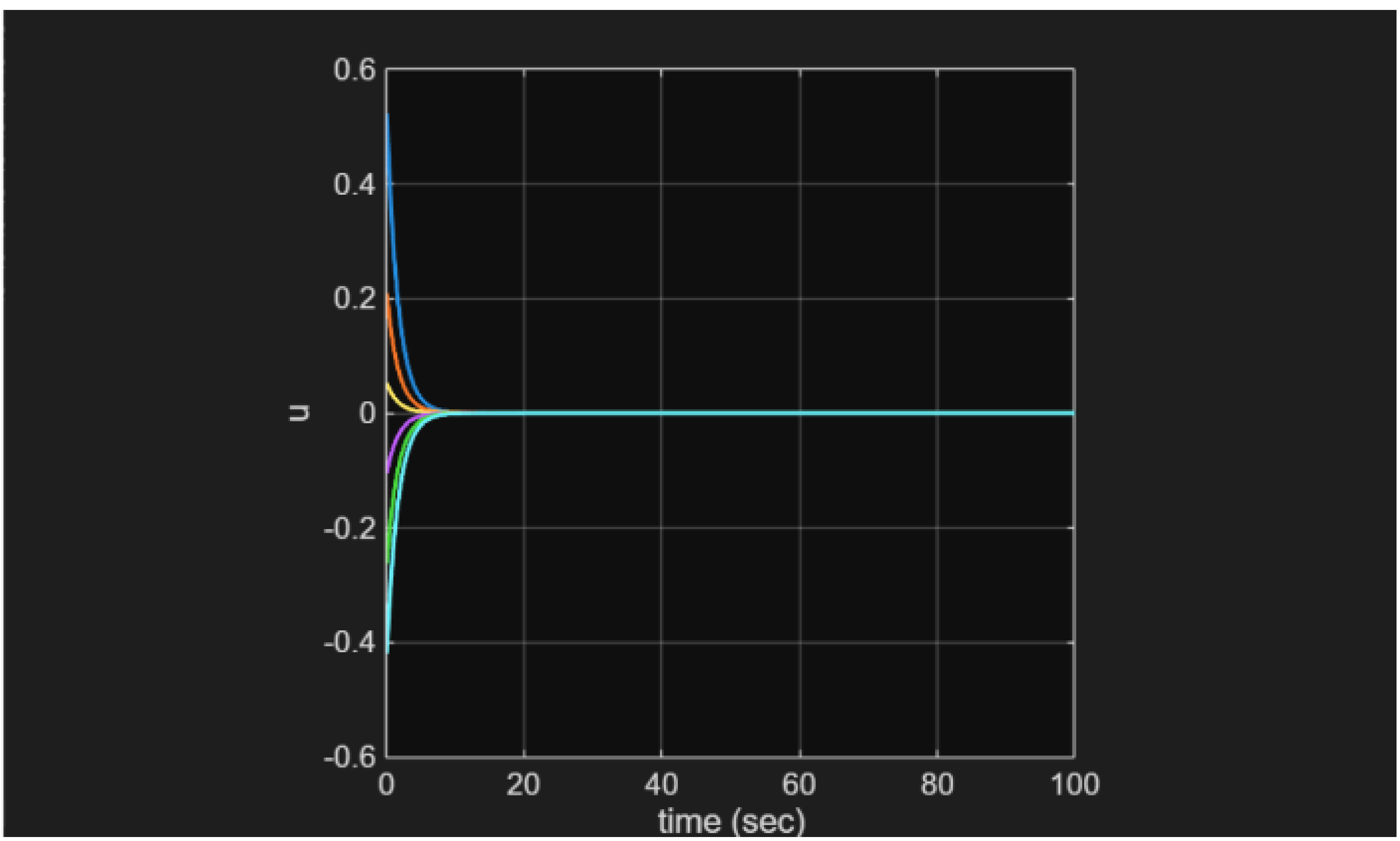
GRAPH 4 :- Trajectories of Robots (K=0.1)



GRAPH 5: Velocity Directions vs. Time (K=0.1)

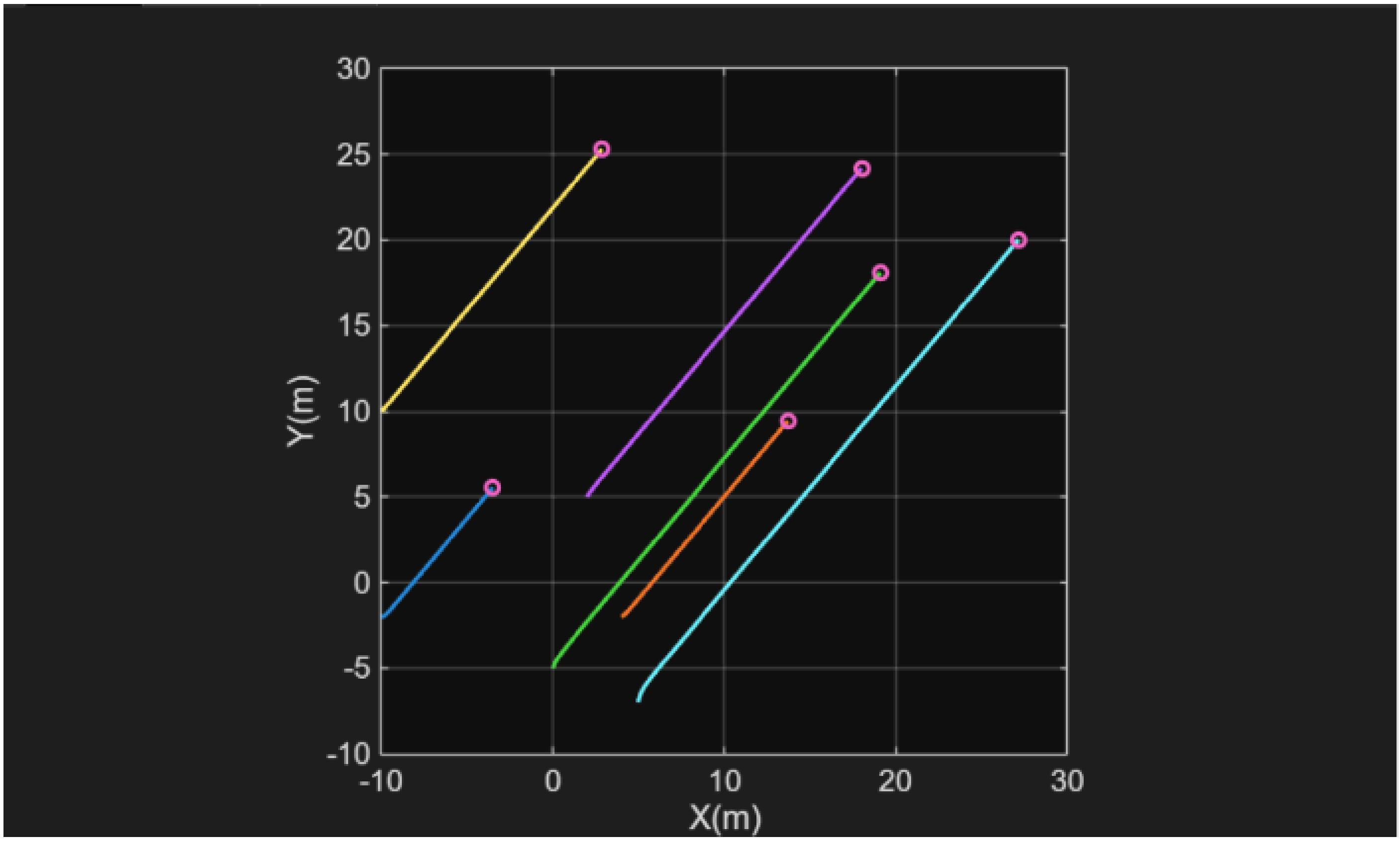


GRAPH 6: Control Inputs vs. Time (K=0.1)

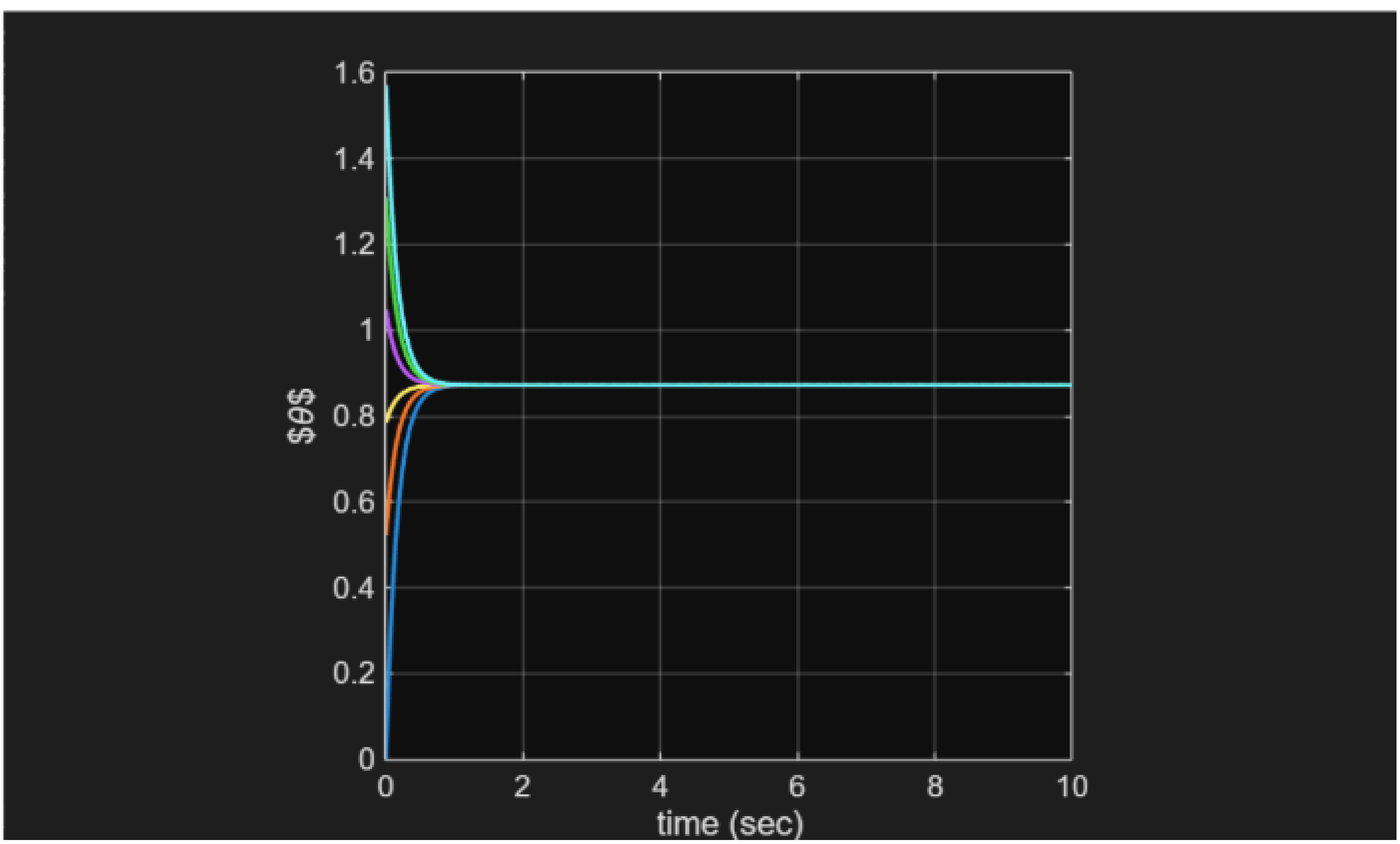


Results for K = 1

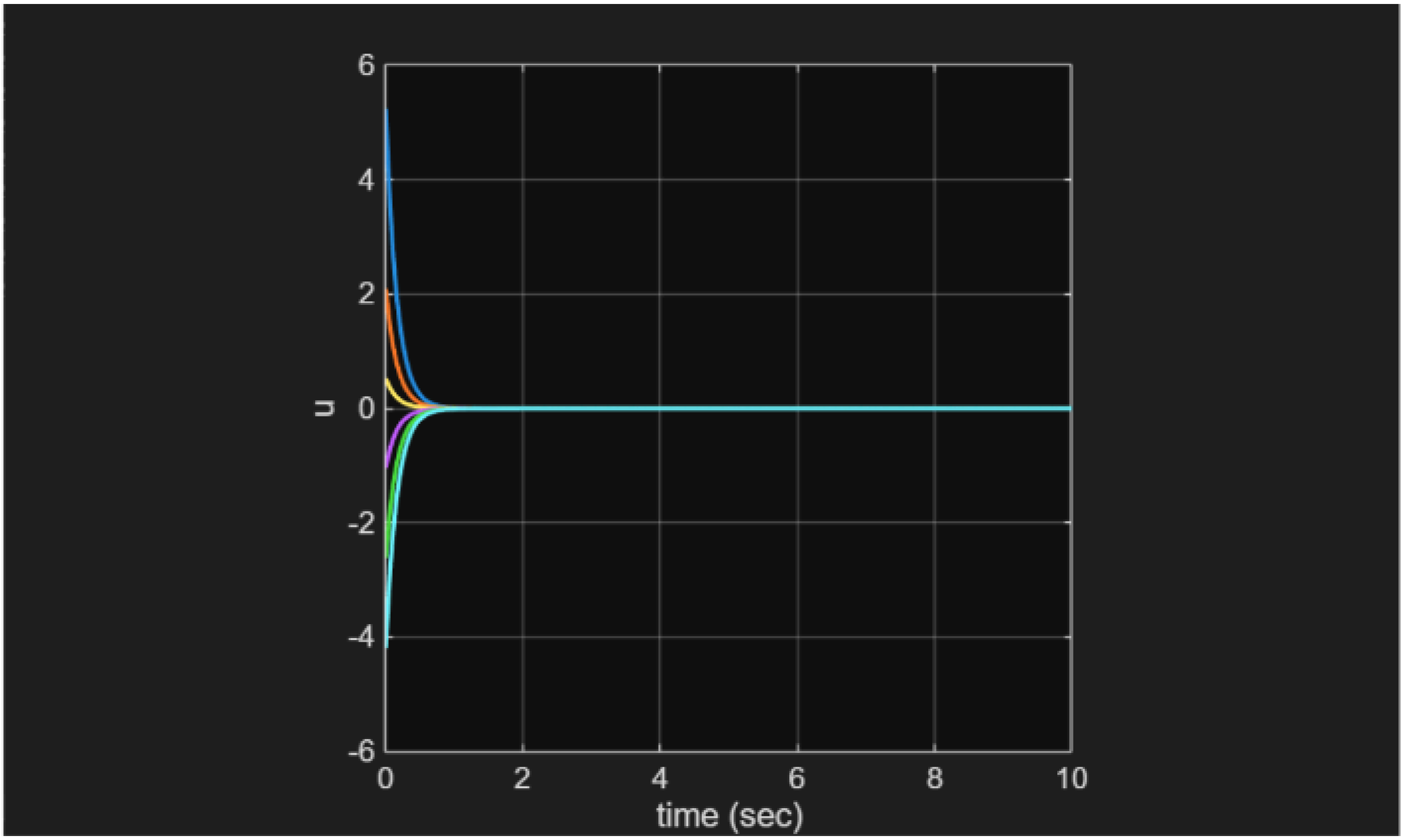
GRAPH 7 : Trajectories of Robots (K=1)



GRAPH 8: Velocity Directions vs. Time (K=1)



GRAPH 9: Control Inputs vs. Time (K=1)



Q1(b): Simulation with Sparse Topology (Figure 1b)
The system was simulated using the below script for K = 0.01 and K = 1.

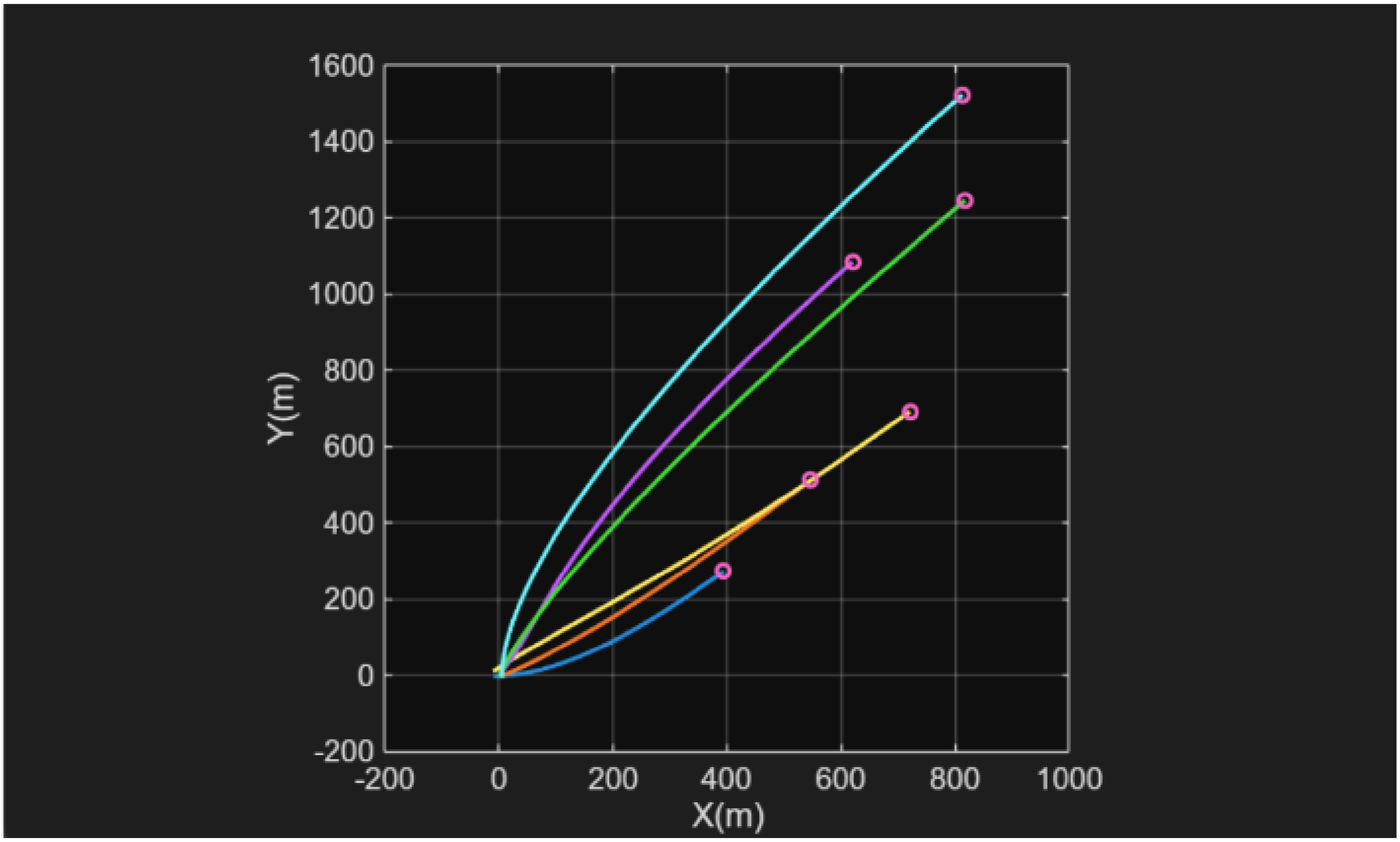
Code :-

The part of the code where changes are made in comparison with the previous code

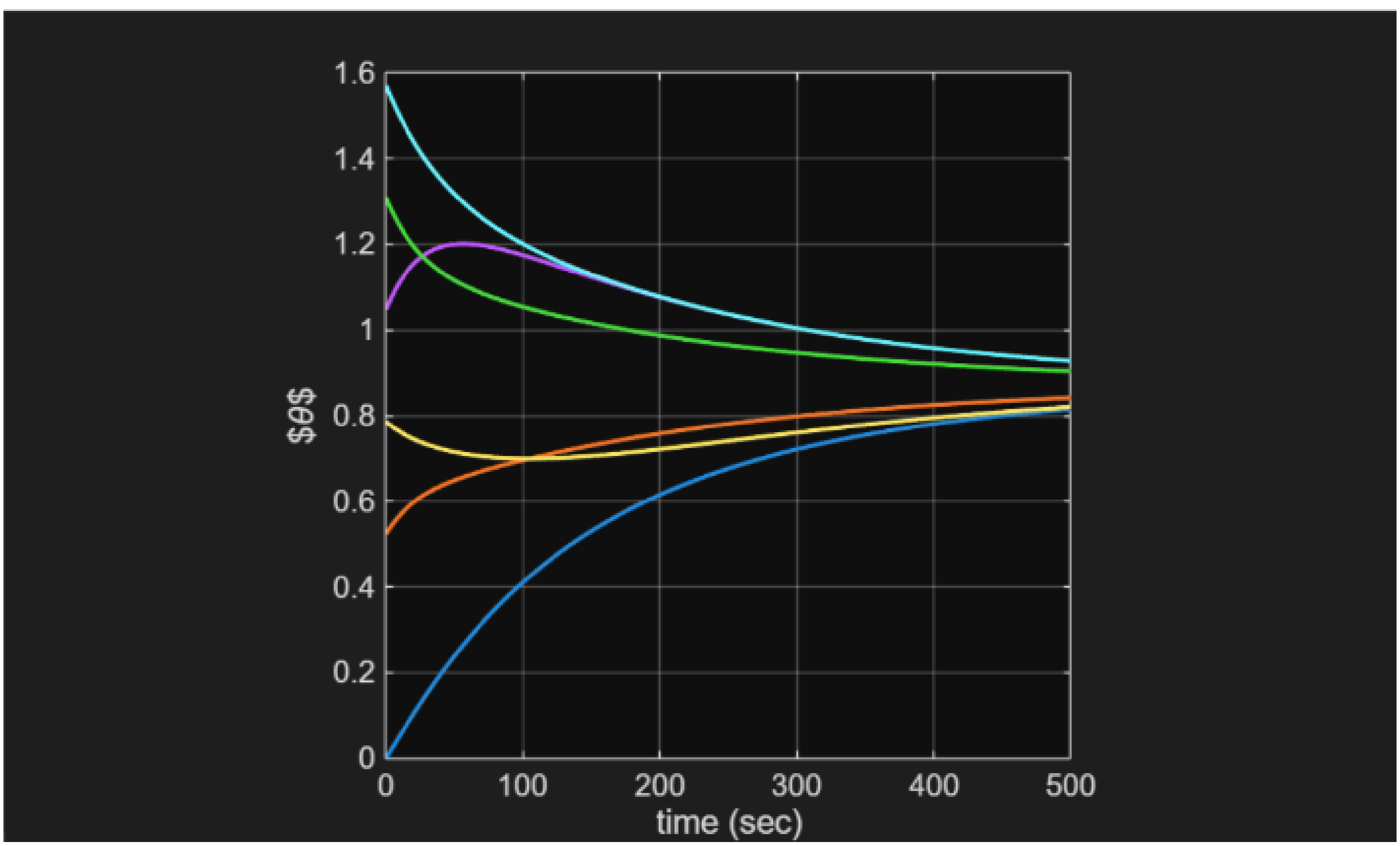
```
%% control law

  u(1) = K*(theta(2) - theta(1)); % control law for the first robot
  u(2) = K*(theta(1) + theta(3) + theta(5) - 3*theta(2)); % control law for the second robot
  u(3) = K*(theta(2) - theta(3)); % control law for the third robot
  u(4) = K*(theta(5) + theta(6) - 2*theta(4)); % control law for the fourth robot
  u(5) = K*(theta(2) + theta(4) + theta(6) - 3*theta(5)); % control law for the fifth robot
  u(6) = K*(theta(4) + theta(5) - 2*theta(6)); % control law for the sixth robot
```

Results for K = 0.01

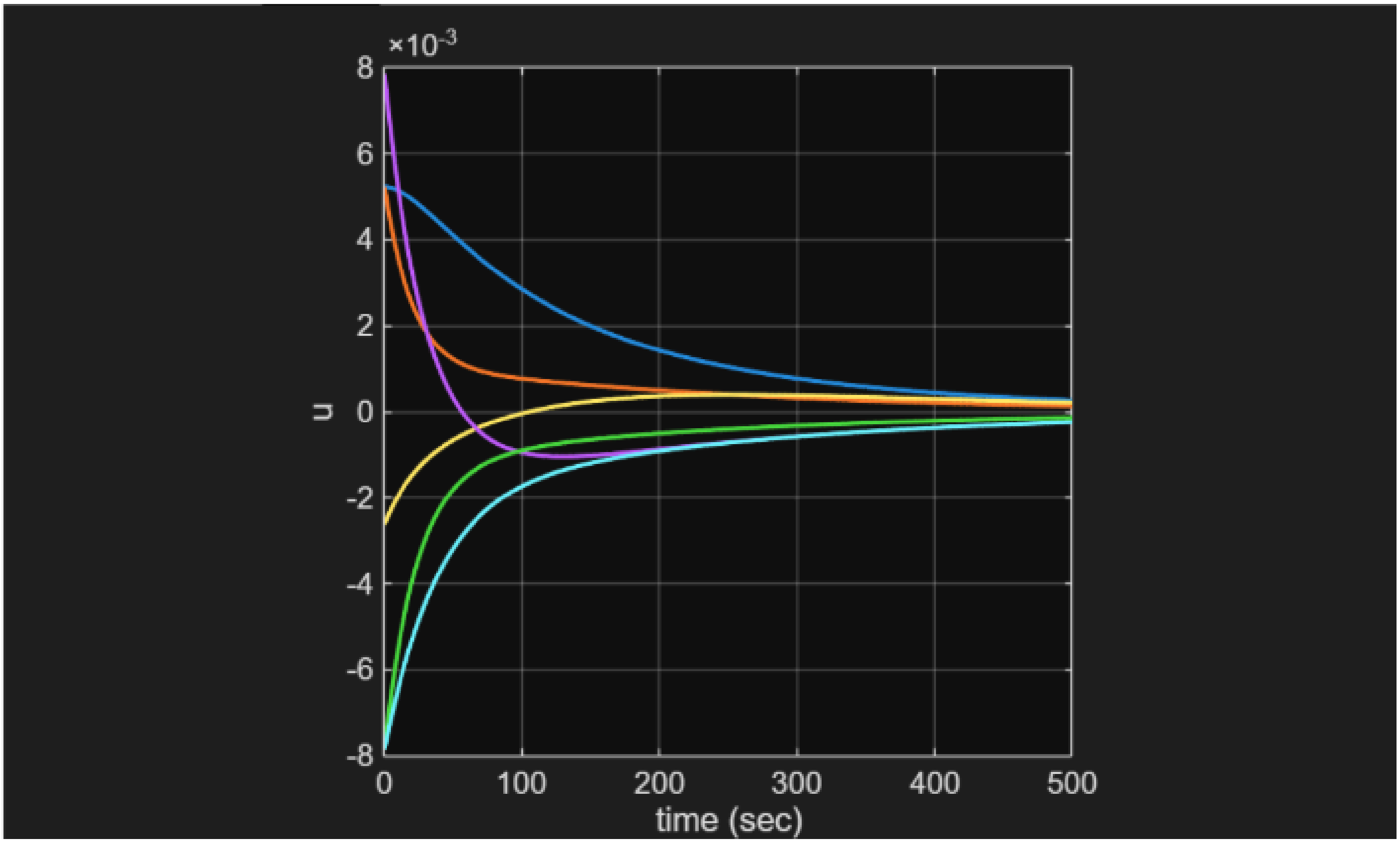
GRAPH 10: Trajectories of Robots (K=0.01)



GRAPH 11: Velocity Directions vs. Time (K=0.01)

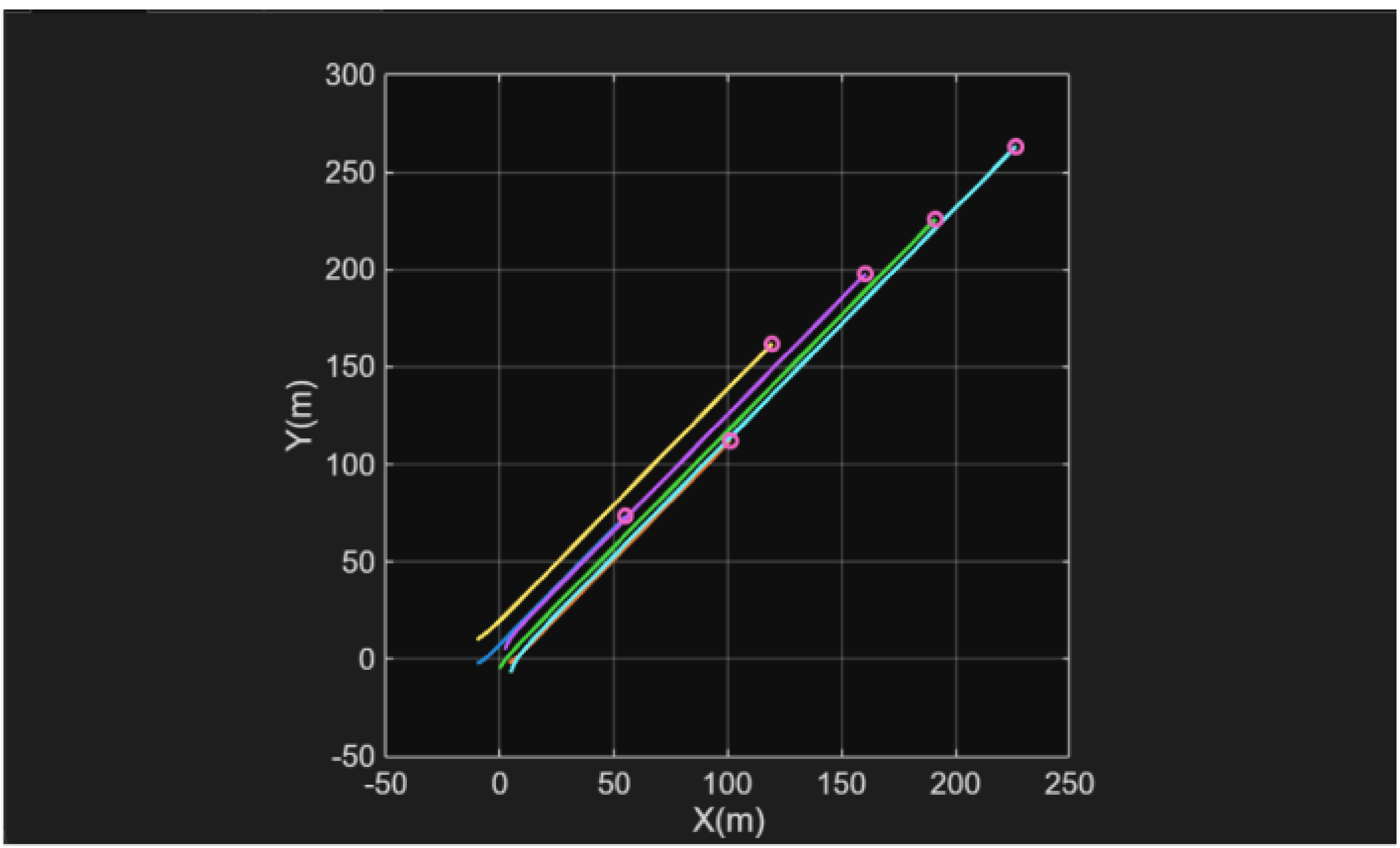


GRAPH 12: Control Inputs vs. Time (K=0.01)

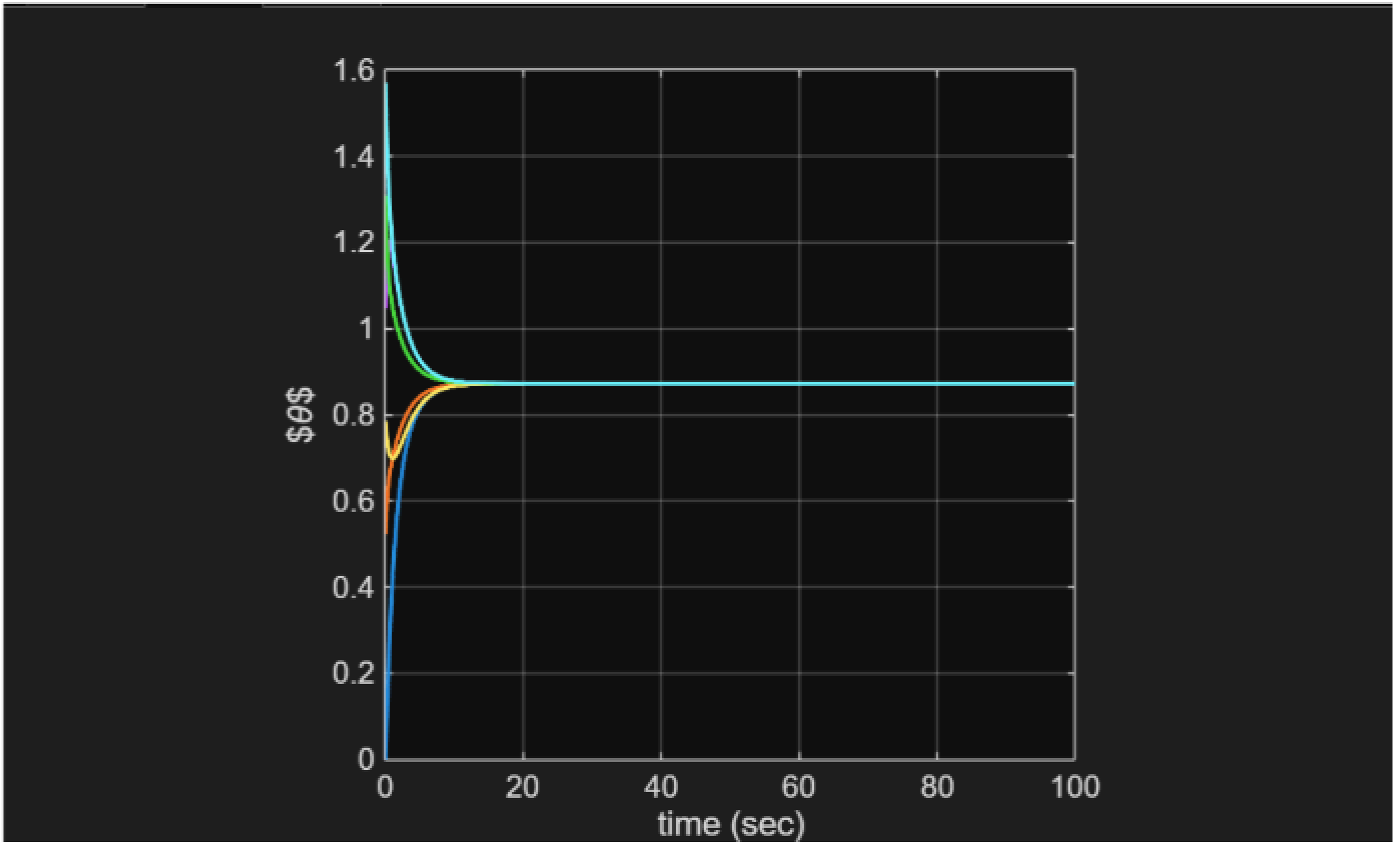


Results for K = 1

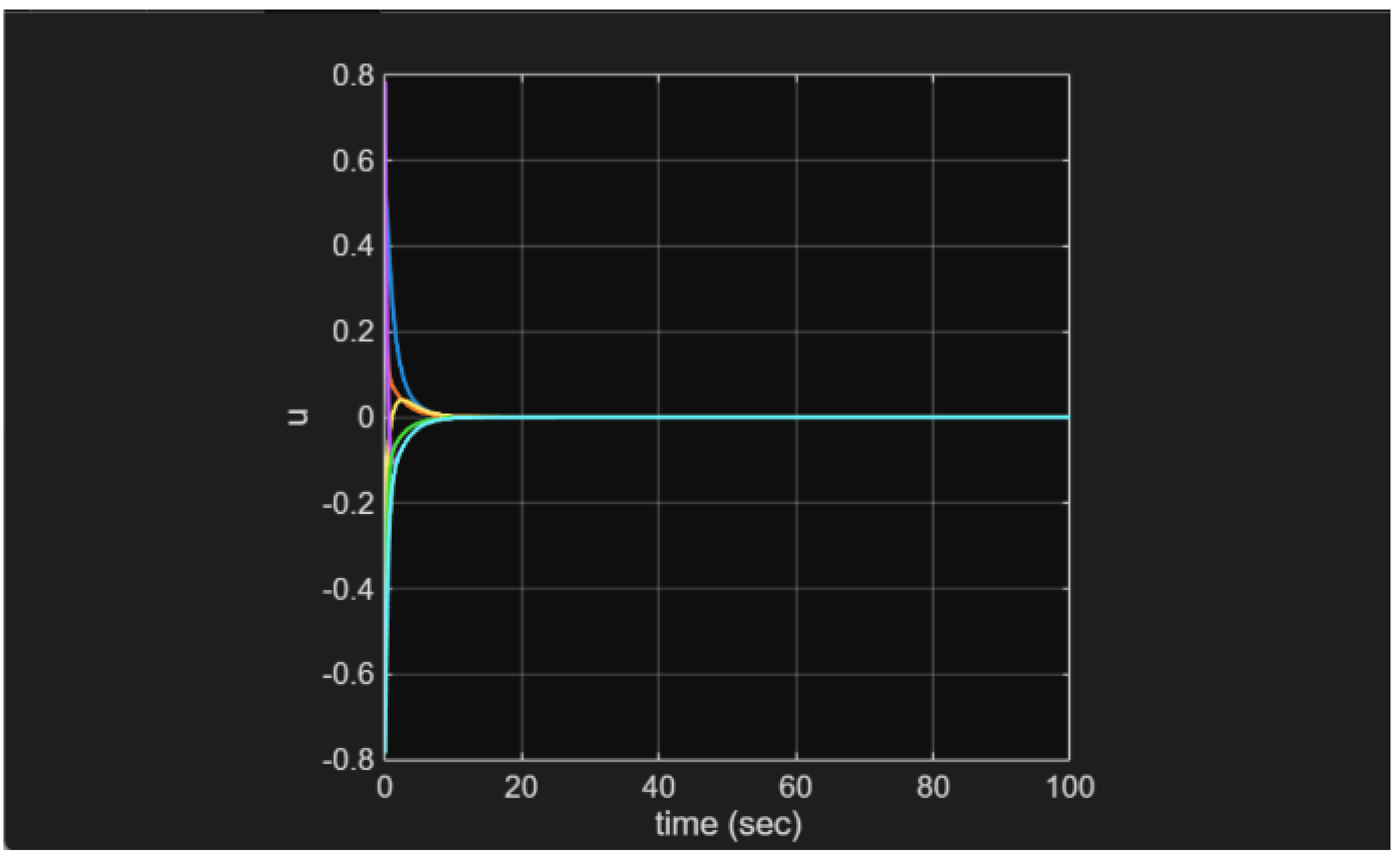
GRAPH 13: Trajectories of Robots (K=1)



GRAPH 14: Velocity Directions vs. Time (K=1)



GRAPH 15: Control Inputs vs. Time (K=1)



Q 1 c) Consensus Value Analysis :-

The consensus in velocity directions occurs at the average of the initial velocity directions of all robots. Initial Directions: $0^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ$.

Q.1 c) : Consensus Value Calculations

By using the control laws of Q1 (a) & Q.1(b) we can see that

$$\sum_{i=1}^{N} u_i = 0 .$$

& since $\dot{\theta}_i(t) = u_i$ (eqn 1c)

$$\rightarrow \sum_{i=1}^{N} \dot{\theta}_i(t) = 0$$

$$\sum_{i=1}^{N} \theta_i(t) = \text{Constant.}$$

assuming $\theta_i(t) = \theta_0$ at steady state.

$$\sum_{i=1}^{N} \theta_i(t) = \sum_{i=1}^{N} \theta_0 = N\theta_0$$

$$\theta_0 = \frac{\sum_{i=1}^{N} \theta_i(t)}{N} \longrightarrow$$ Does not depends on K.

for Q.1

$$\theta_0 = \frac{0+30+45+60+75+90}{6}$$

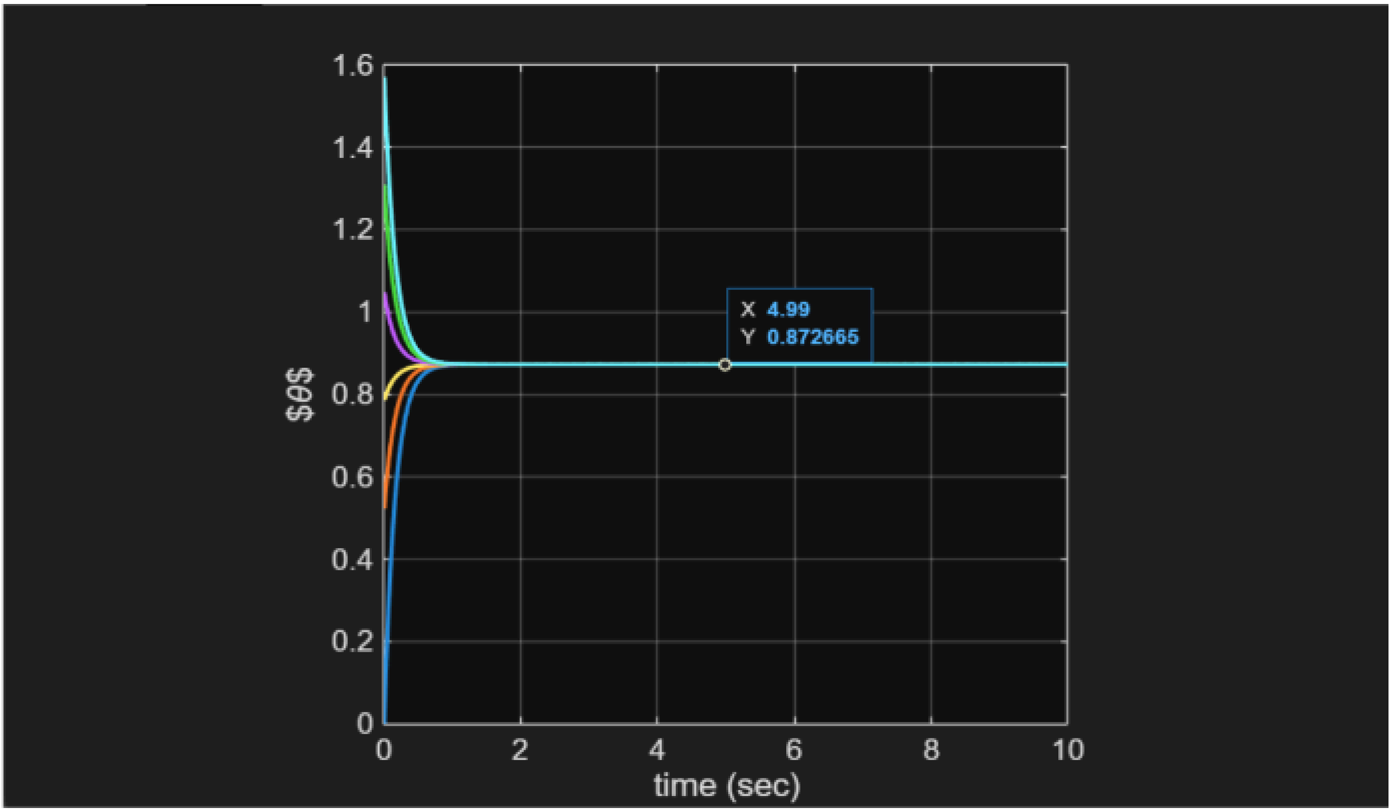
$$\boxed{\theta_0 = 50^\circ}$$

Practical Values :→ for both Q.1(a) & Q.1(b) as seen in graph 2 →

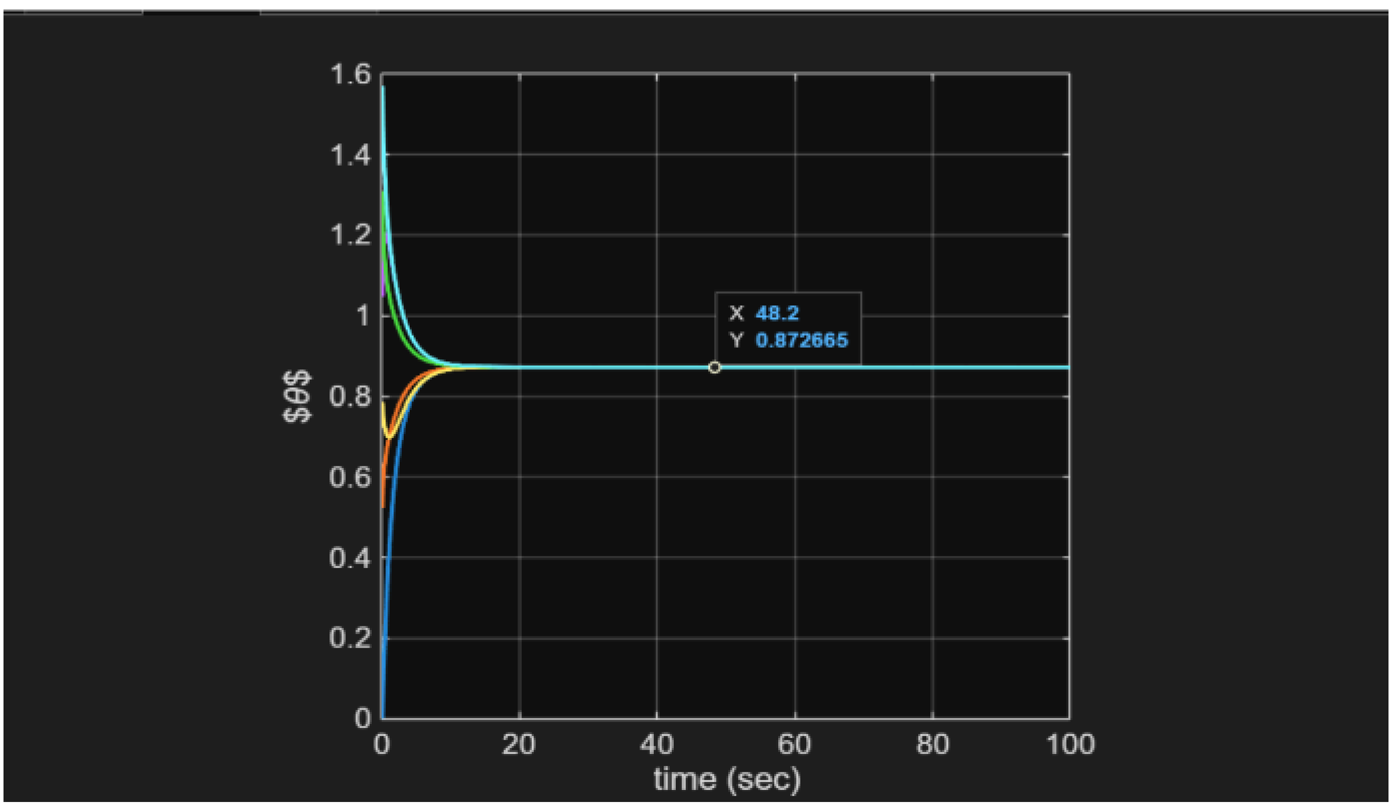
$$\theta_0 = 0.873 \times \frac{180}{\pi}$$

$$\theta_0 \approx 49.94^\circ$$

GRAPH 16: Velocity Directions vs. Time (K=1)(Topology 1)



GRAPH 17: Velocity Directions vs. Time (K=1)(Topology 2)



This theoretical value holds for any connected network topology where the coupling is symmetric.

Q1(d): Simulation with Removed Link (2-5)

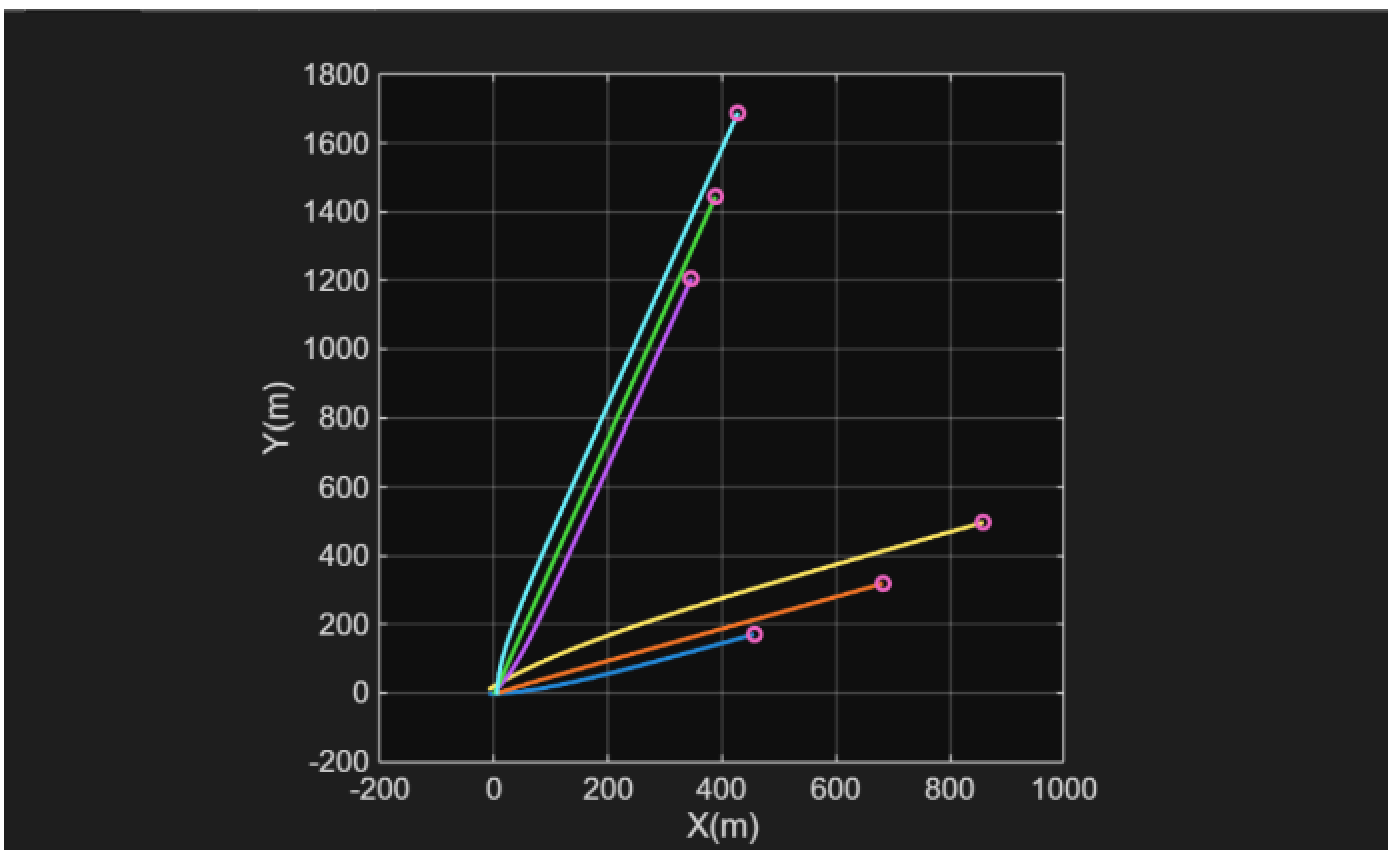
The system was simulated using the below script for K = 0.01 and K = 1.

Code:

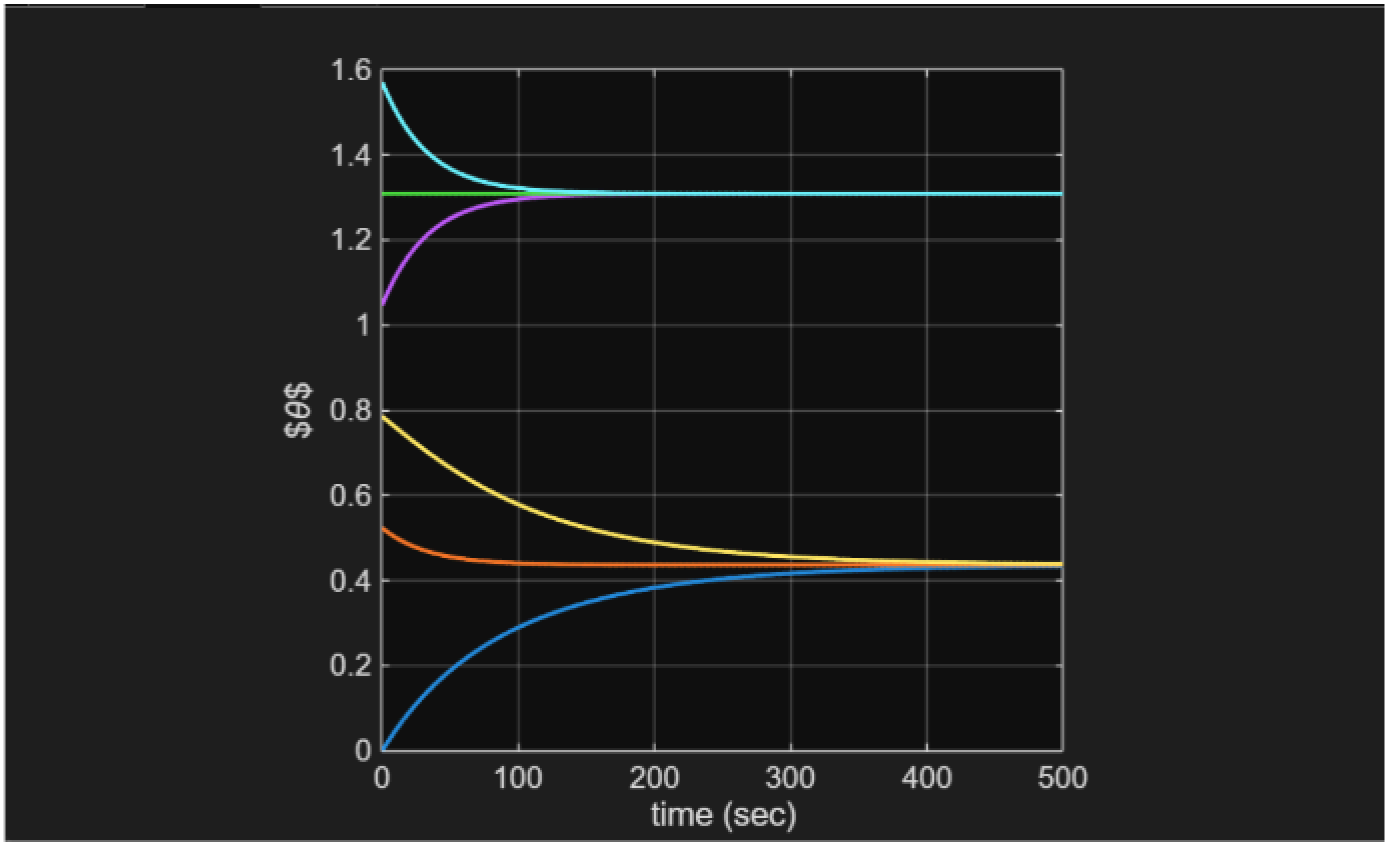
```
%% control law

  u(1) = K*(theta(2) - theta(1)); % control law for the first robot
  u(2) = K*(theta(1) + theta(3) - 2*theta(2)); % control law for the second robot
  u(3) = K*(theta(2) - theta(3)); % control law for the third robot
  u(4) = K*(theta(5) + theta(6) - 2*theta(4)); % control law for the fourth robot
  u(5) = K*(theta(4) + theta(6) - 2*theta(5)); % control law for the fifth robot
  u(6) = K*(theta(4) + theta(5) - 2*theta(6)); % control law for the sixth robot
```

Results for K = 0.01

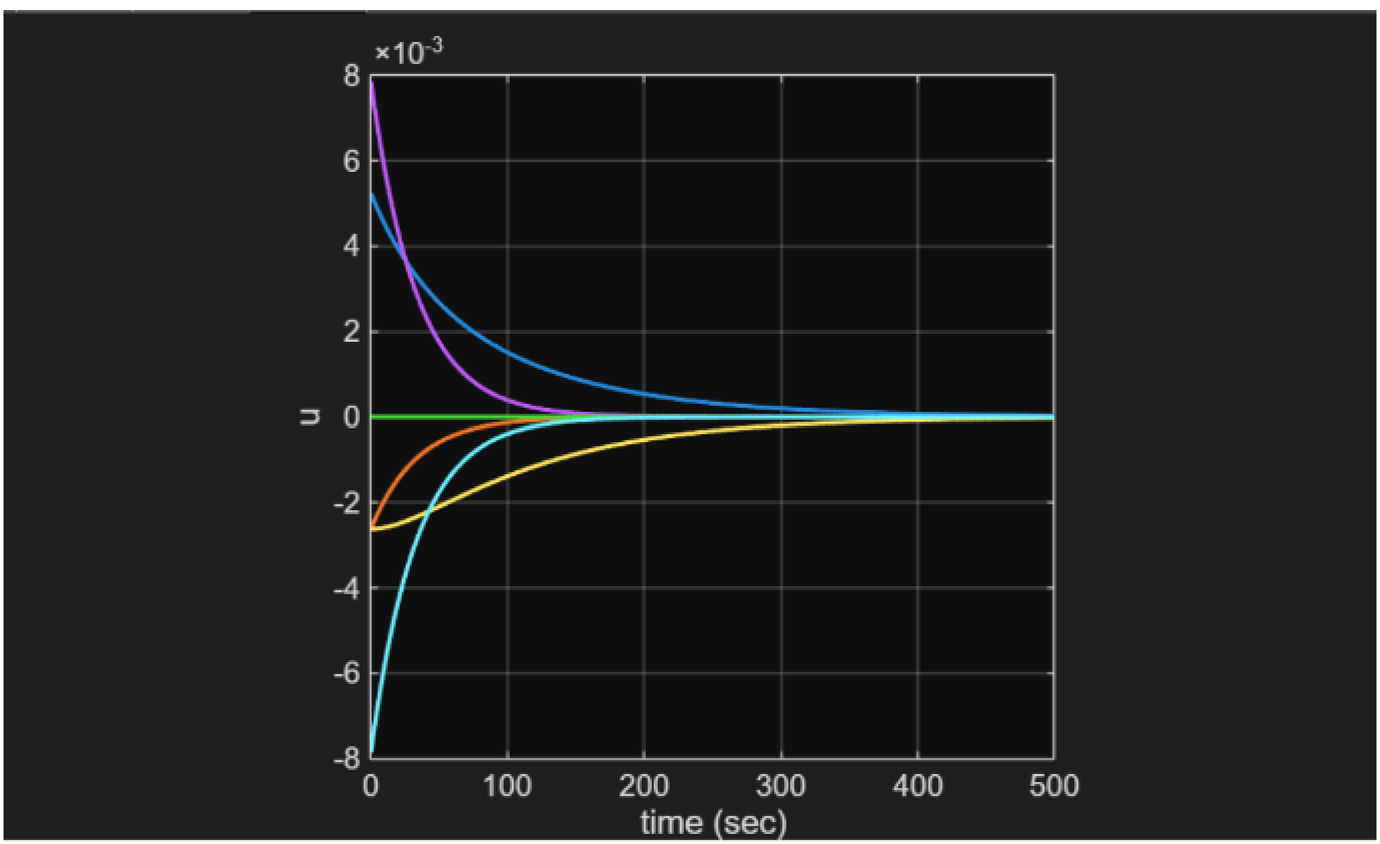
GRAPH 18: Trajectories of Robots (K=0.01)



GRAPH 19: Velocity Directions vs. Time (K=0.01)

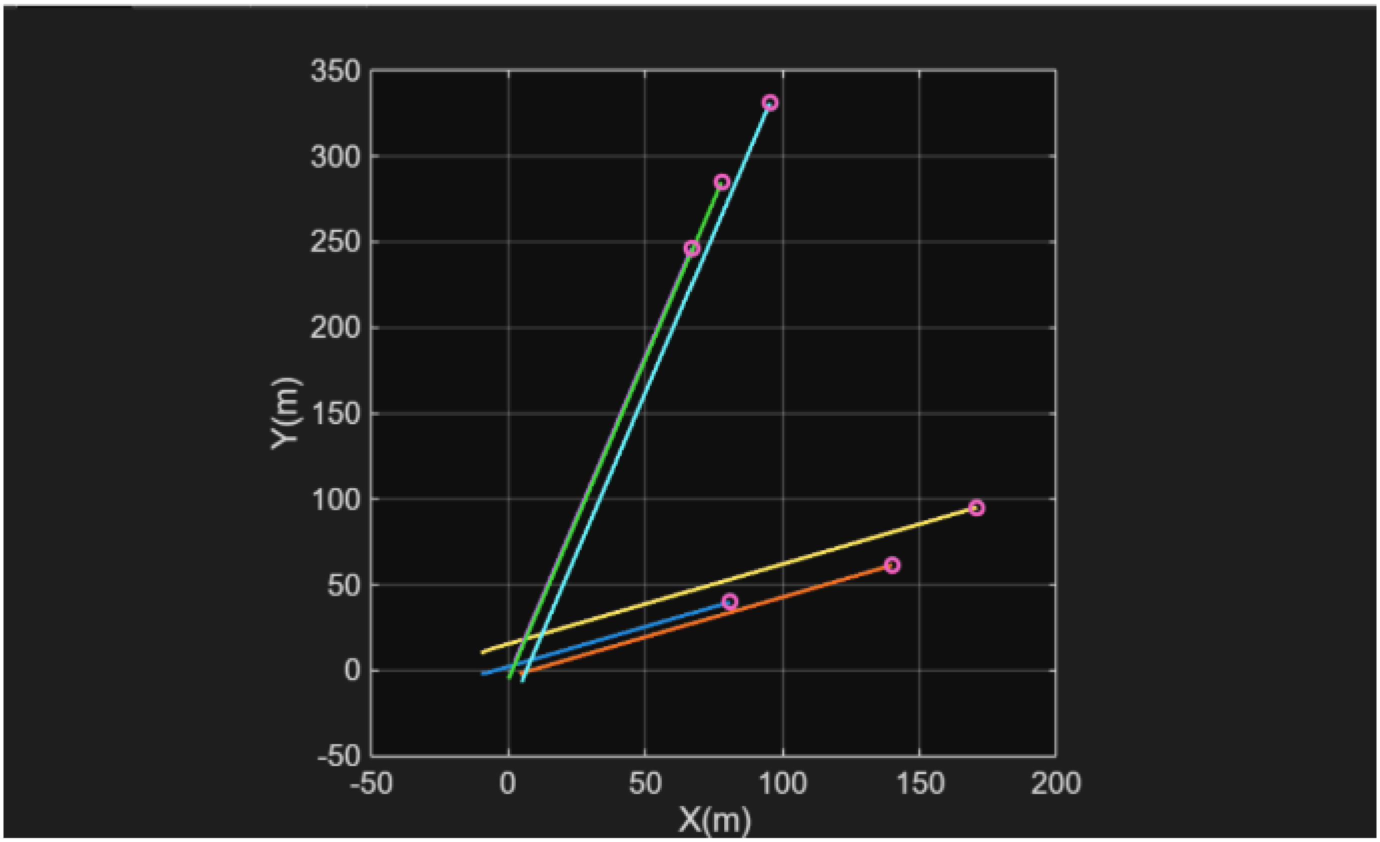


GRAPH 20: Control Inputs vs. Time (K=0.01)

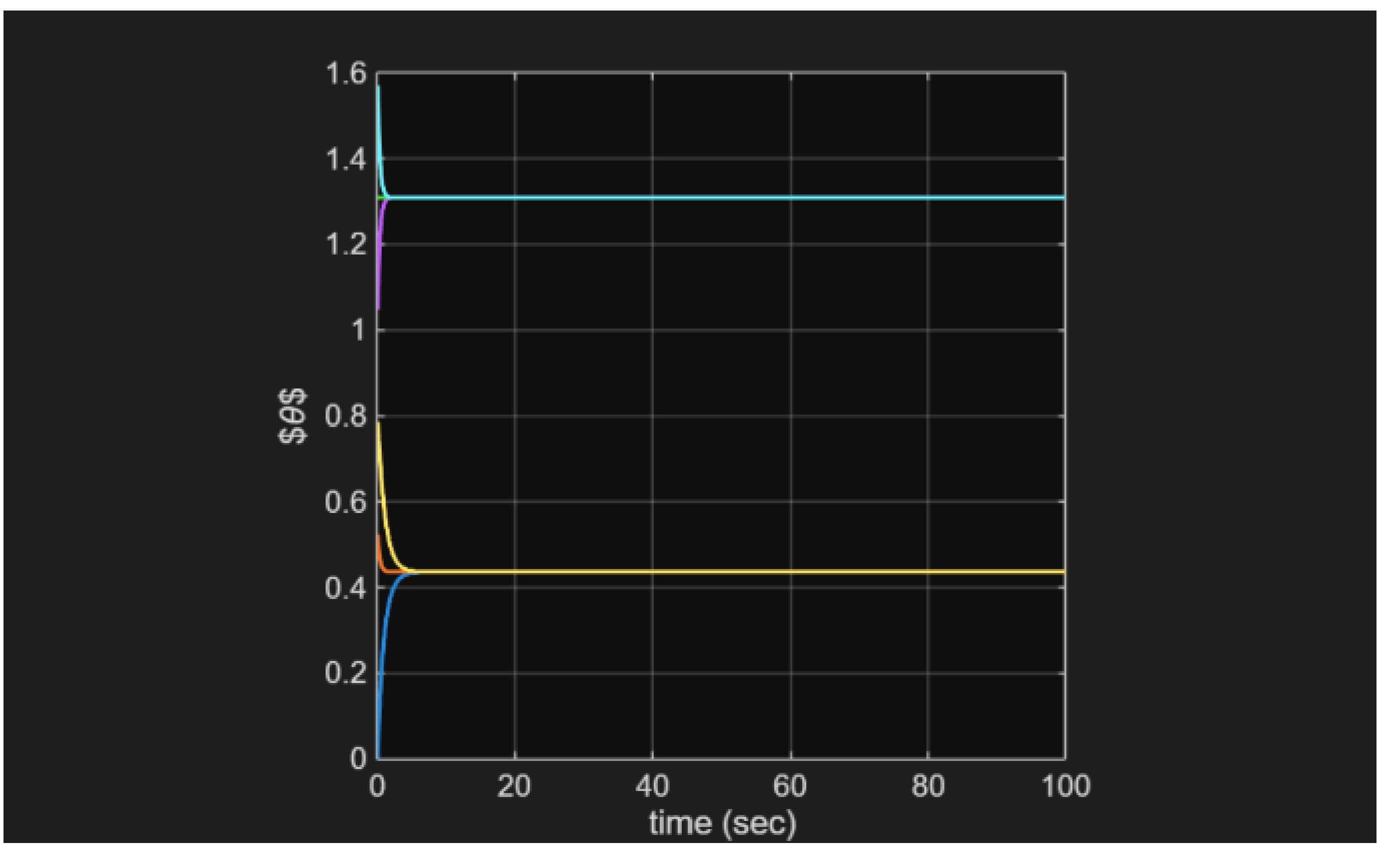


Results for K = 1

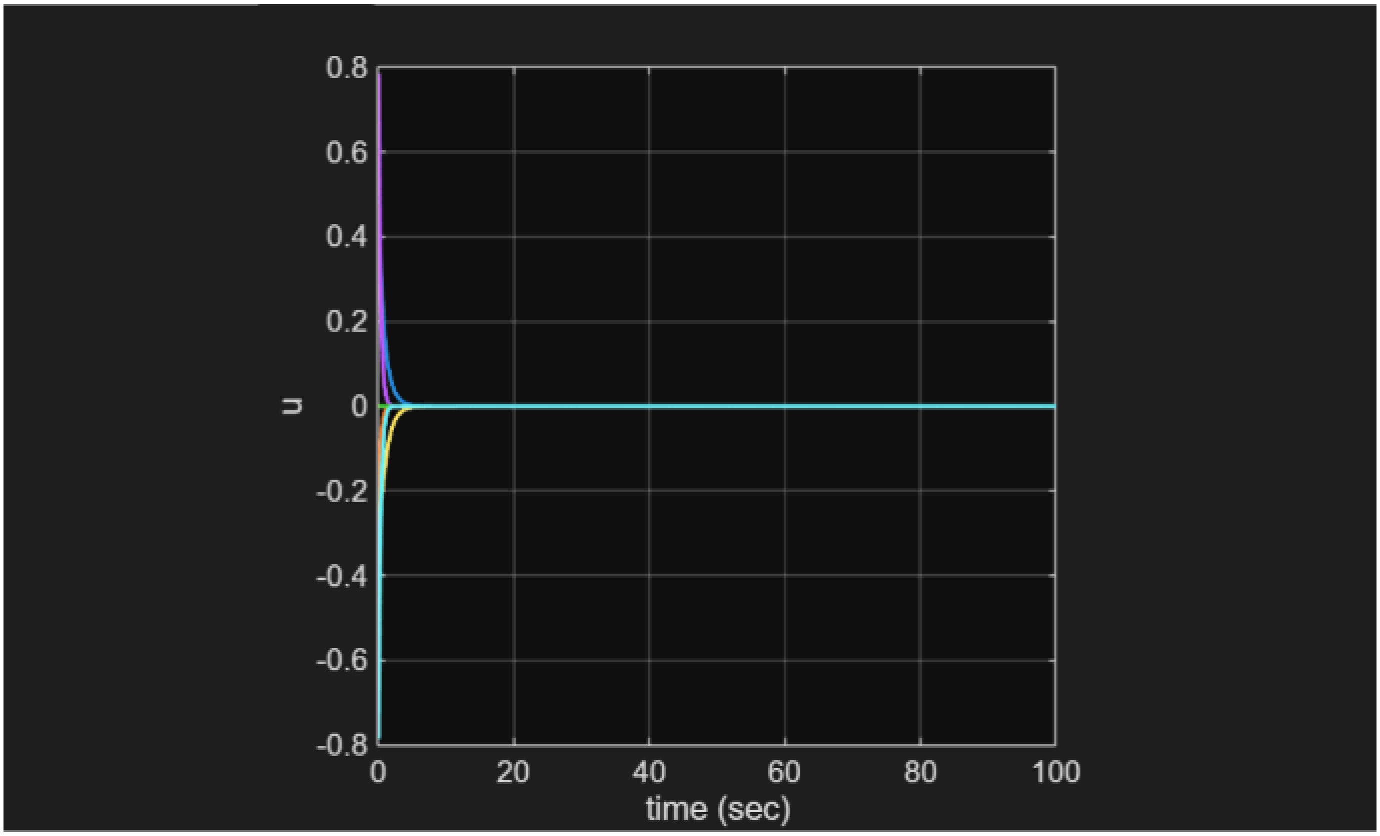
GRAPH 21: Trajectories of Robots (K=1)



GRAPH 22: Velocity Directions vs. Time (K=1)



GRAPH 23: Control Inputs vs. Time (K=1)



Q2: Simulation with Individual Gains :-

The system was simulated using the below script, which implements the control law with individual gains K_i for each agent.

Code:

The changes made from the topology 2 (In Q1(b)) in this case are -

```
%% control law

    u(1) = K1*(theta(2) - theta(1)); % control law for the first robot
    u(2) = K2*(theta(1) + theta(3) + theta(5) - 3*theta(2)); % control law for the second robot
    u(3) = K3*(theta(2) - theta(3)); % control law for the third robot
    u(4) = K4*(theta(5) + theta(6) - 2*theta(4)); % control law for the fourth robot
    u(5) = K5*(theta(2) + theta(4) + theta(6) - 3*theta(5)); % control law for the fifth robot
    u(6) = K6*(theta(4) + theta(5) - 2*theta(6)); % control law for the sixth robot
```

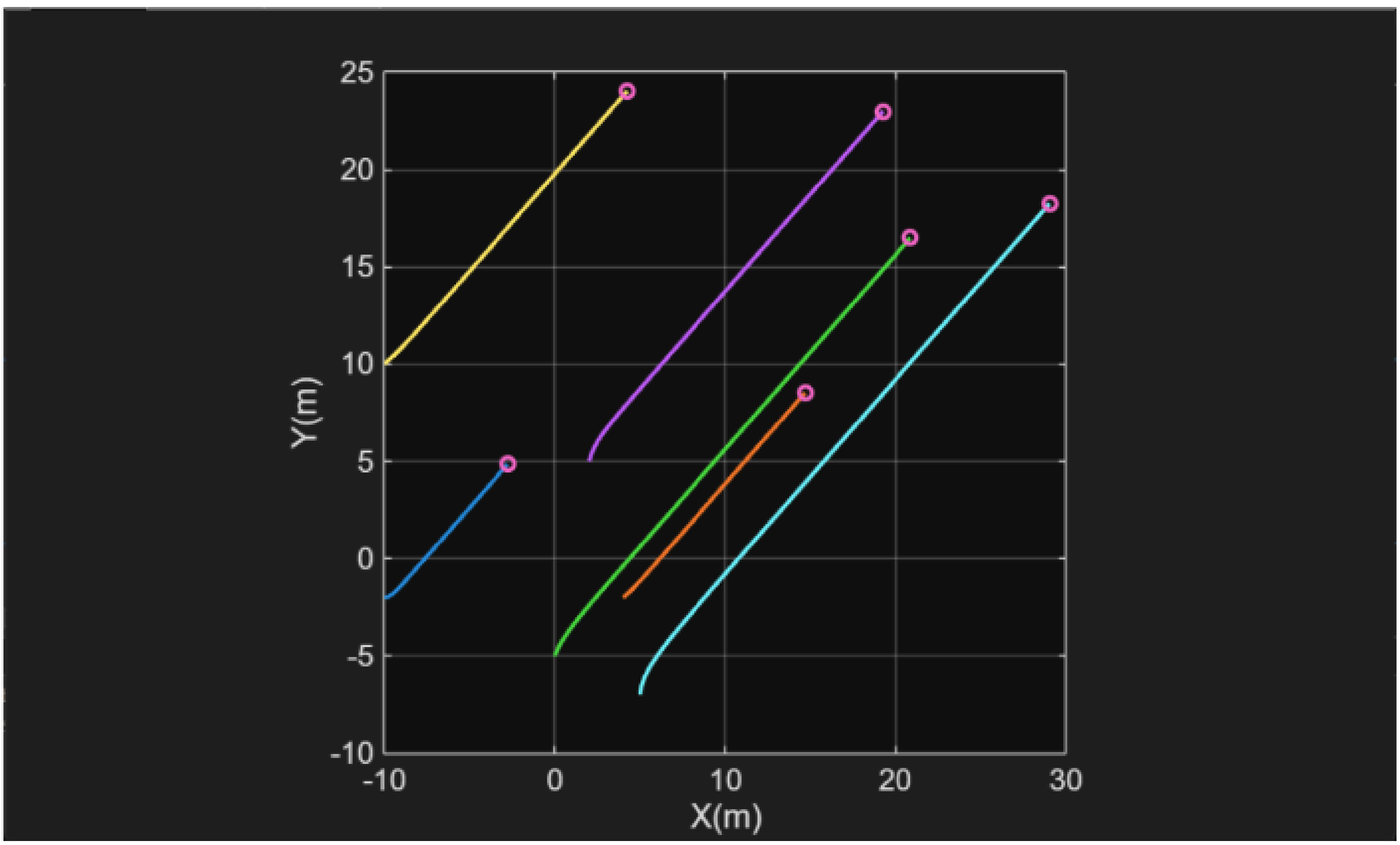
The values of K1-K6 are also different for this question, these are taken care of in their respective parts, Q2(a) and Q2(b). The code snippet is also attached in their respective parts.

Q2(a) :- Analysis of Individual Gains

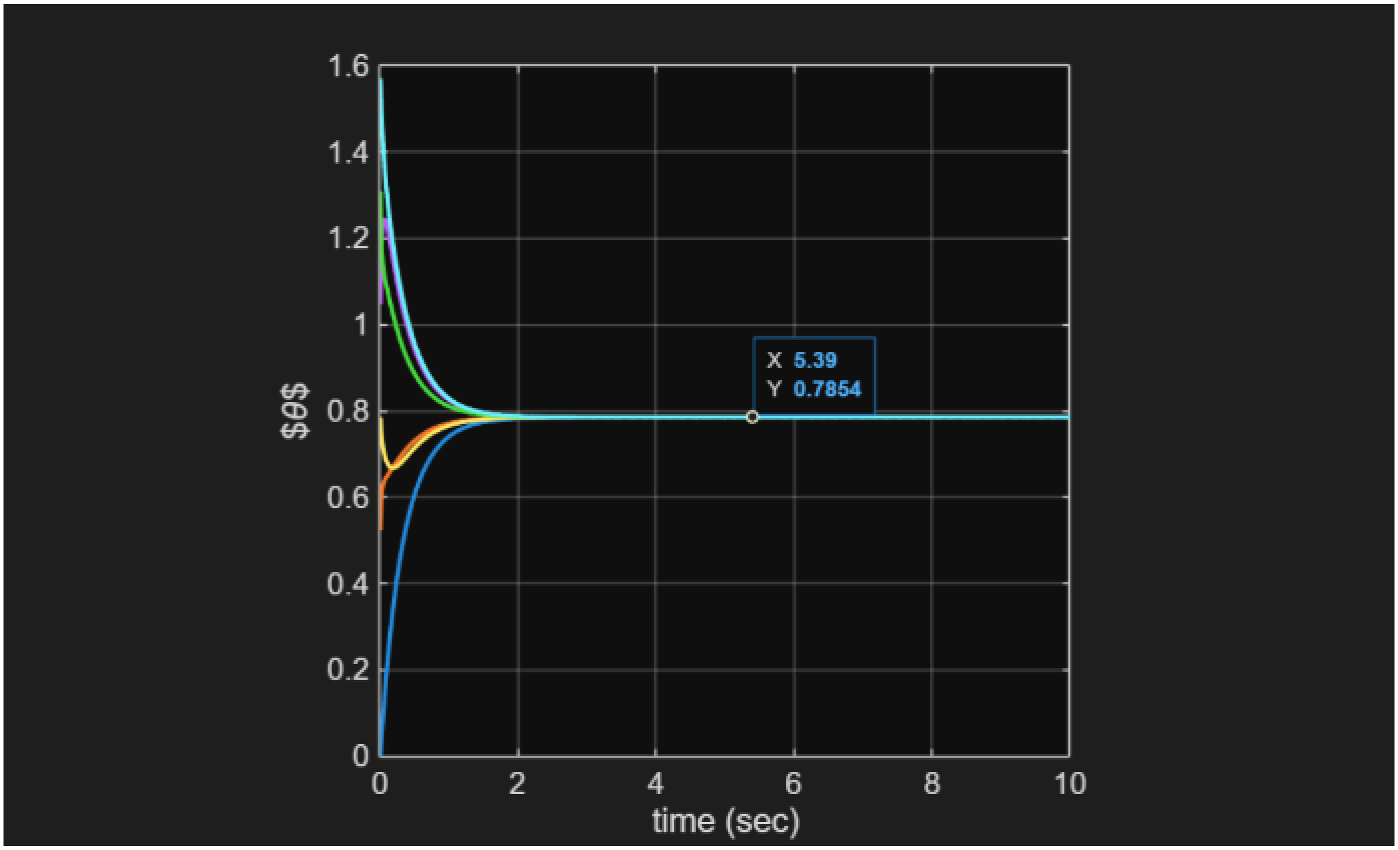
The value of K1-K6 taken for this part are -

```
%% controller gain

K1 = 4;
K2 = 10;
K3 = 10;
K4 = 10;
K5 = 10;
K6 = 5.4545;
```

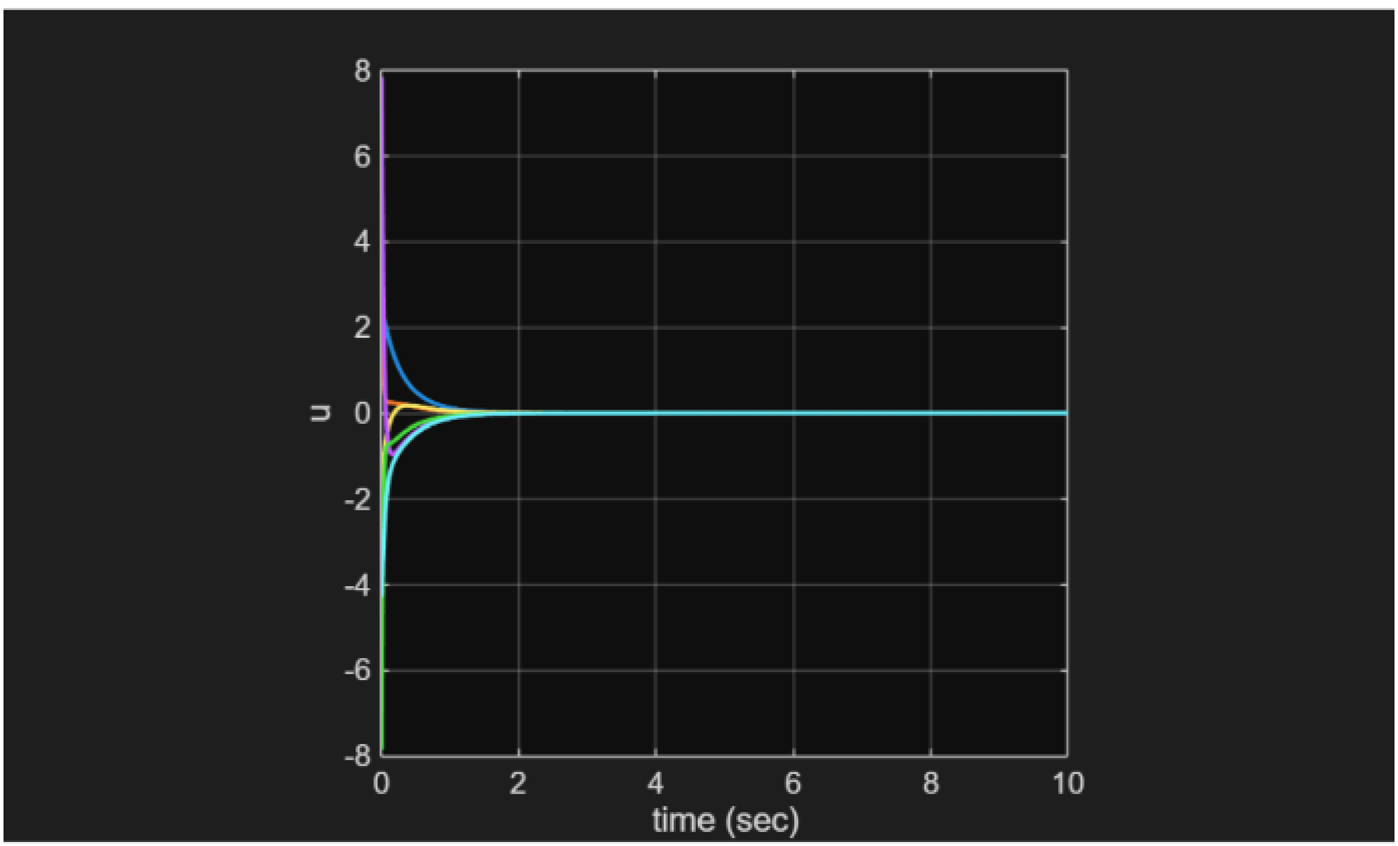
GRAPH 24: Trajectories of Robots (Individual Gains)



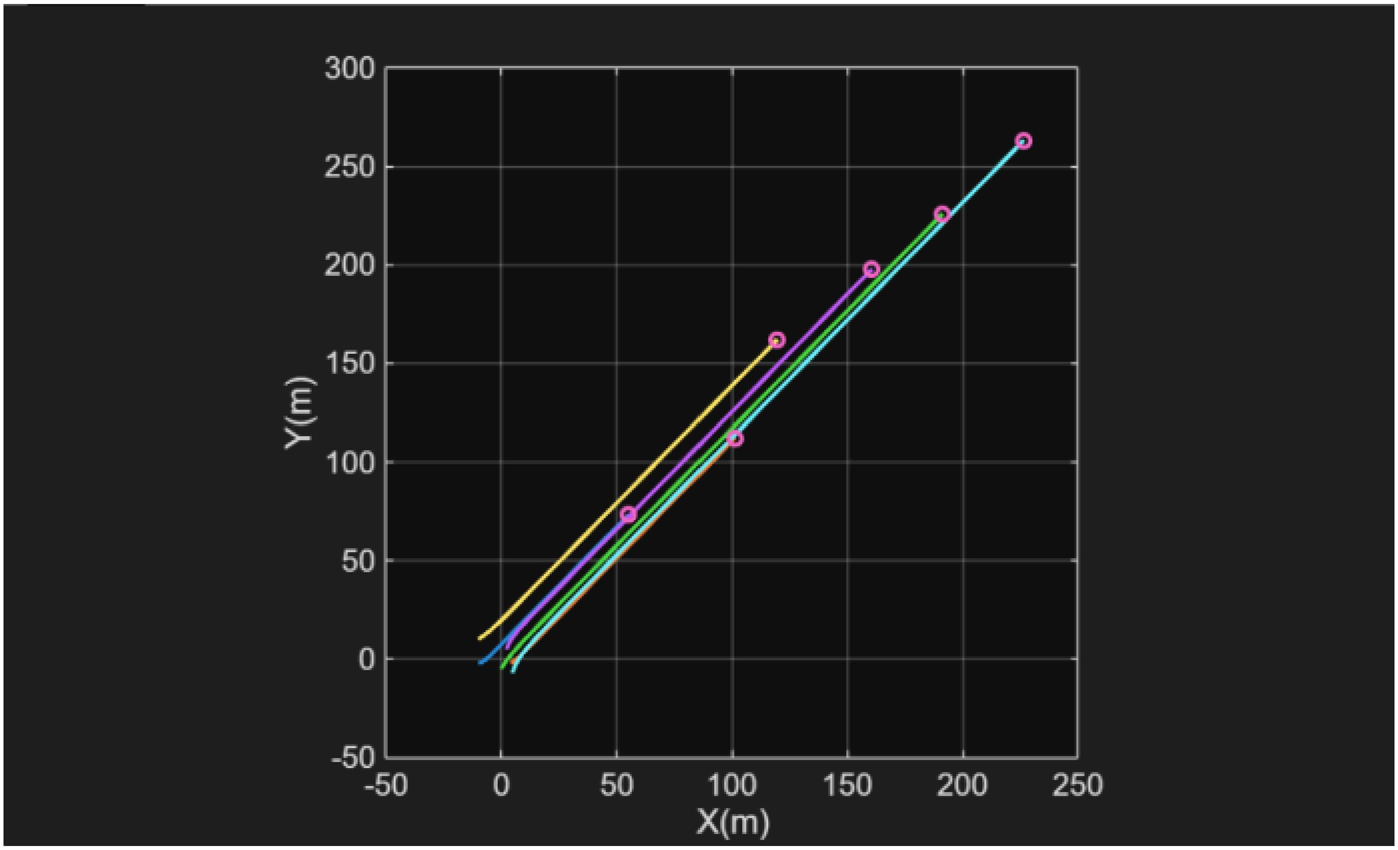
GRAPH 25: Velocity Directions vs. Time (Individual Gains)



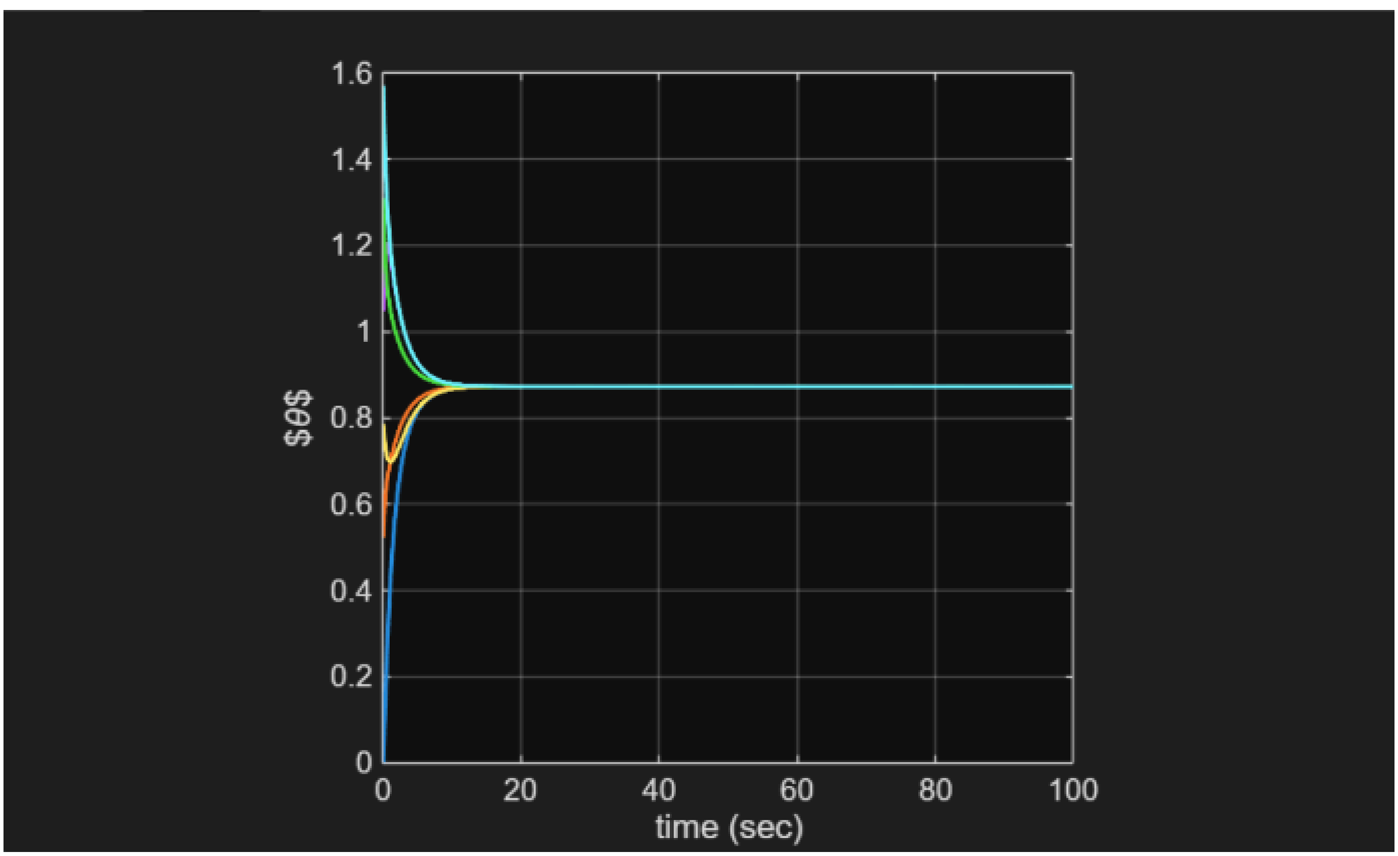
GRAPH 26: Control Inputs vs. Time (Individual Gains)



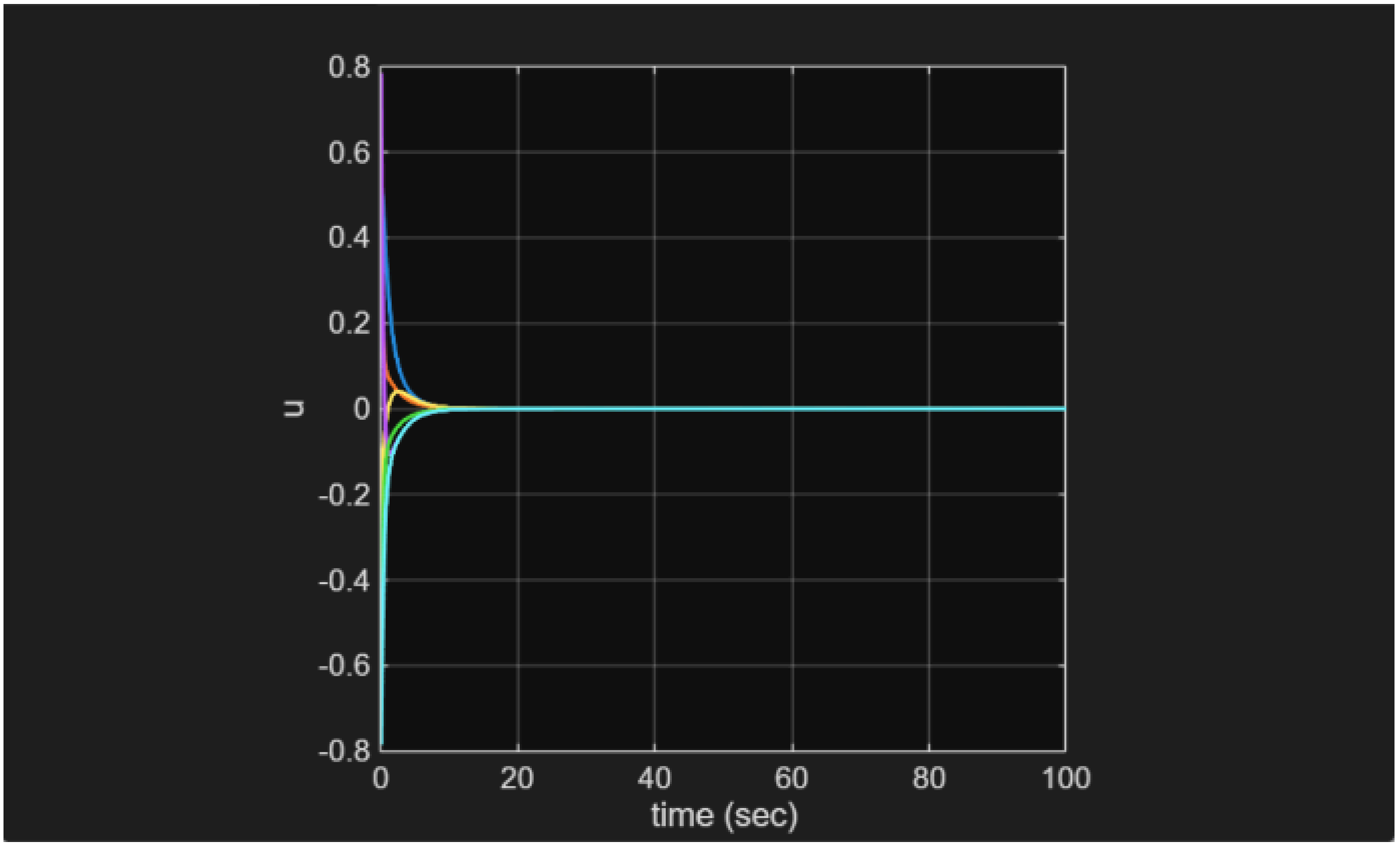
GRAPH 27: Trajectories of Robots (Uniform Gain K=1)



GRAPH 28: Velocity Directions vs. Time (Uniform Gain K=1)



GRAPH 29: Control Inputs vs. Time (Uniform Gain K=1)



Q2(b): Achieving a Pre-specified Consensus

The results from Q2(a) demonstrate that consensus can be achieved at a value different from the simple average by adjusting individual gains.
Now, if we want to achieve consensus at 45 degrees by adjusting the values of K1-K6, then the results will be:

```
%% controller gain

K1 = 4;
K2 = 10;
K3 = 10;
K4 = 10;
K5 = 10;
K6 = 5.4545;
```

Calculations :-

Q.2 b) in this case

$$\sum_{i=1}^{N} \frac{u_i}{k_i} = 0$$

$$\theta_0 = \frac{\sum_{i=1}^{N} \frac{u_i}{k_i}}{\sum_{i=1}^{N} \frac{1}{k_i}}$$

If we want to achive consensus at $\underline{\theta_0 = 45^\circ}$ then,

$$\frac{\pi}{4} = \frac{0}{k_1} + \frac{\pi}{6k_2} + \frac{\pi}{4k_3} + \frac{\pi}{3k_4} + \frac{5\pi}{6k_5} + \frac{\pi}{2k_6}$$

assume $\boxed{k_2 = k_3 = k_4 = k_5 = 10}$

& finding $k_6 = ?$

$$\frac{\pi}{4} = \frac{\pi}{60} + \frac{\pi}{40} + \frac{\pi}{30} + \frac{5\pi}{60} + \frac{\pi}{2k_6}$$

$$\boxed{k_6 = 5.45}$$

& let $k_1 = 4$. → since it is not depending on k_1

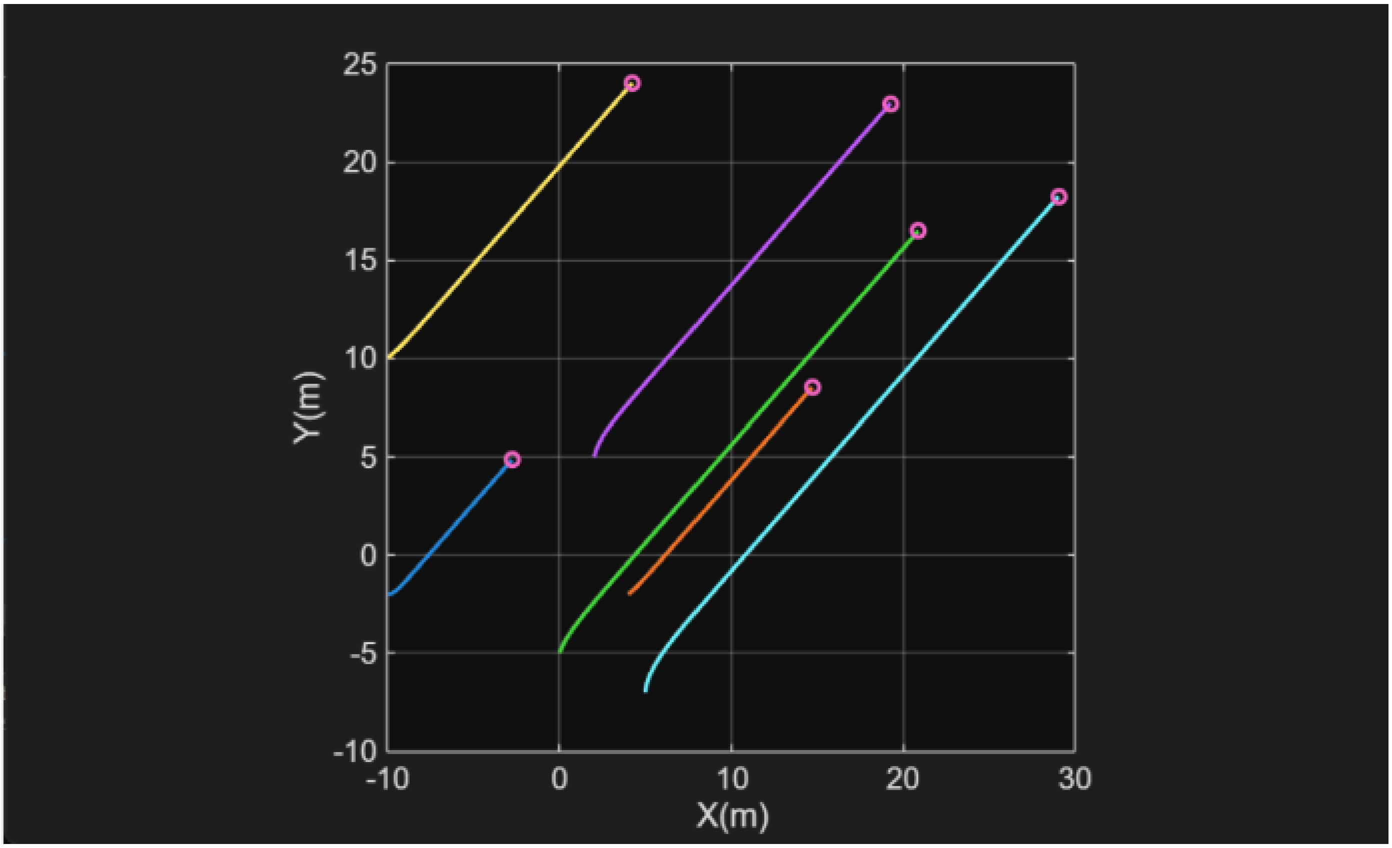
By Putting these values & Plotting the graph —

we get $\theta_0 = 0.79 \times \frac{180}{\pi}$

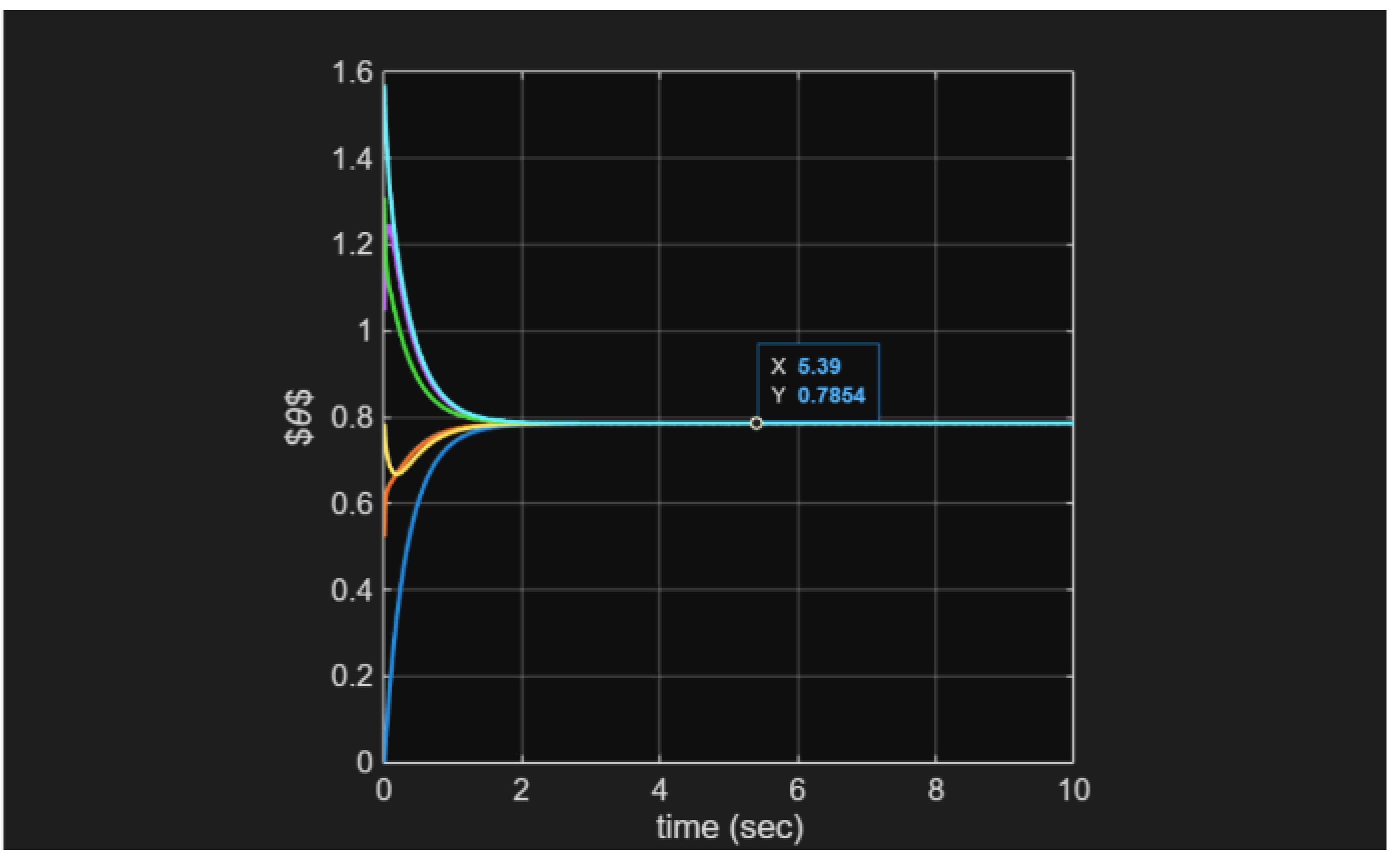
$= 45.28^\circ$

↳ so we get consensus at desired value.

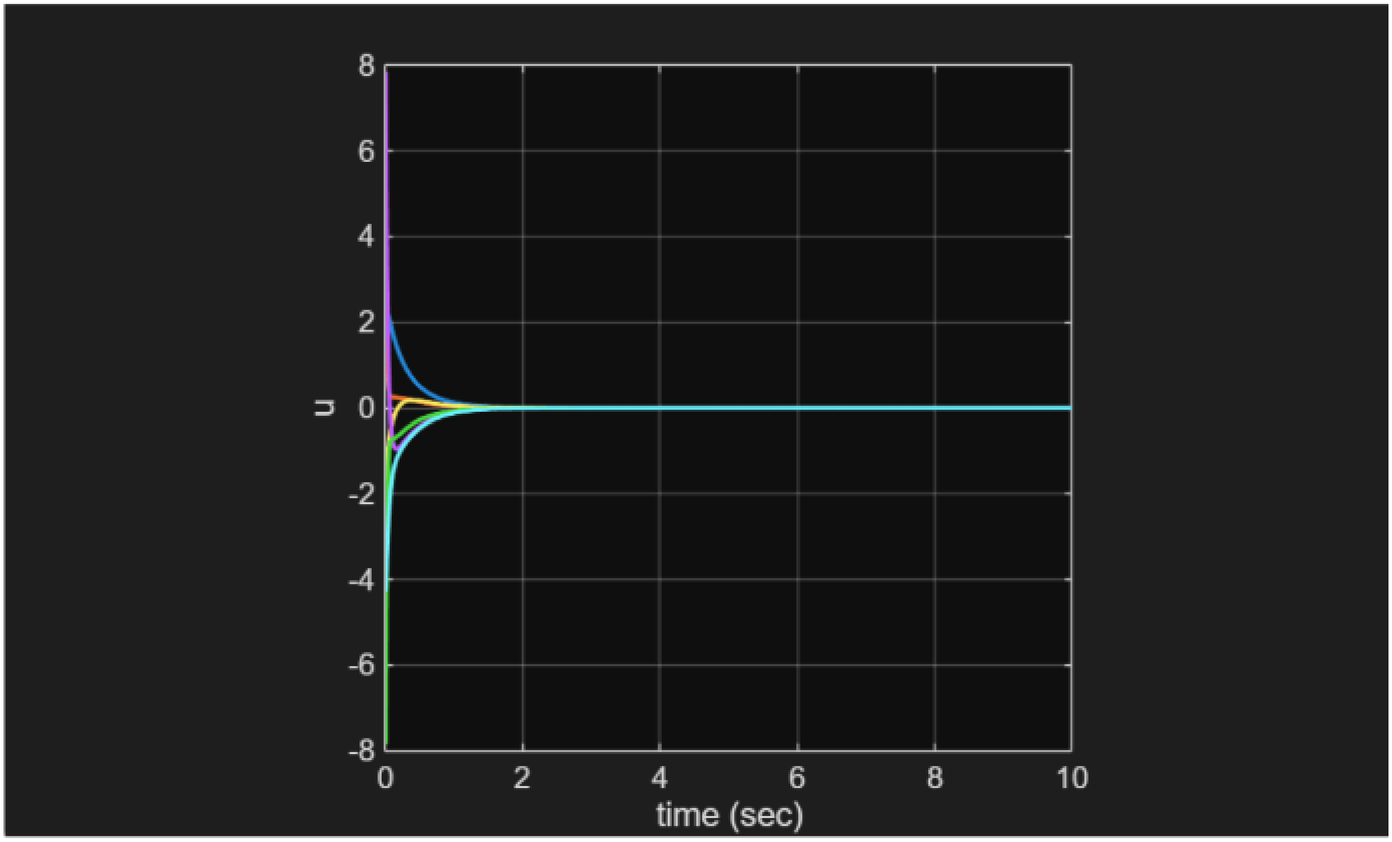
GRAPH 30: Trajectories of Robots (Targeted Consensus)



GRAPH 31: Velocity Directions vs. Time (Targeted Consensus)



GRAPH 32: Control Inputs vs. Time (Targeted Consensus)



Results and Analysis :-

Q1(a) & Q1(b): Effect of Topology and Gain K :-

In both the fully connected (1a) and sparse (1b) topologies, the robots eventually aligned their velocity directions to a common value. The θ versus time plots confirm this, as all six trajectories merge toward the same angle. The final consensus angle is close to 50° (about 0.87 radians), which matches the theoretical mean of the initial headings.

The key distinction between the two cases lies in how quickly consensus is reached. With the same controller gain (K=0.01), the fully connected network converges much faster than the sparse one. This happens because, in a fully connected setup, each agent can directly exchange information with every other agent, leading to quicker agreement.

The gain K itself plays a major role in the convergence rate. Larger values of K speed up the process significantly—when K=1, the system reaches consensus within a few seconds, whereas with K=0.01, the process can take several hundred seconds. The control signal starts out relatively high but gradually decreases to zero as the agents settle on the common direction, which makes sense since the difference (θj−θi) shrinks over time.

Q1(c): Consensus Value Justification :-

The consensus angle of about 50° appears across all connected topologies because the communication graph is both undirected and connected. Since the control strategy relies on the graph Laplacian, it guarantees that the final consensus will be the average of the agents' initial directions. The controller gain K affects the eigenvalues of the system matrix larger values of K produce greater non-zero eigenvalues, which in turn make the error terms decay more quickly. As a result, the system reaches consensus faster. However, the value of K only changes the rate of convergence; it does not affect the final consensus value itself.

Q1(d): Effect of Removing a Link :-

When the connection between robots 2 and 5 was removed, the network still stayed connected, so the agents were able to agree on the same consensus value of 50°. The difference was that the convergence took longer compared to the original sparse topology (1b) with the same gain K. This slowdown occurs because removing a link decreases the graph's overall connectivity (i.e., lowers its algebraic connectivity). With weaker connectivity, information takes more time to travel across the two subgroups of robots {1, 2, 3} and {4, 5, 6} which are now only linked through indirect paths.

Q2: Effect of Individual Gains :-

When each agent is assigned its own gain Ki, the group still manages to reach consensus, but the agreed-upon value is no longer just the simple average of the initial states. Instead, it becomes a weighted average, where the weights depend on both the chosen gains and the network structure. The simulation in Q2(a) illustrates this: the system settles at a value different from 50°. This outcome highlights the idea discussed in Q2(b) by carefully tuning the individual gains Ki, the consensus can be guided toward a desired direction. Agents with larger gains exert more influence, effectively pulling the final consensus closer to their initial states. This feature adds an extra level of flexibility in shaping the group's overall behavior.

Conclusion :-

This experiment, carried out through MATLAB simulations, clearly illustrated how consensus emerges in multi-agent systems. The main insights are as follows :-

- A distributed control strategy allows a team of autonomous agents to align their velocity directions and eventually agree on a common value.

- The structure of the communication network plays a vital role. Denser topologies, like a fully connected network, enable quicker convergence, whereas sparser ones converge more slowly. Still, as long as the graph remains connected, consensus will be reached.

- The controller gain K governs how quickly agents converge. Larger values of KKK accelerate convergence but may pose challenges in real-world scenarios due to actuator limits and increased sensitivity to noise.

- With a uniform gain K and an undirected graph, the final consensus is simply the average of the agents' initial conditions.

- Allowing each agent to have its own gain Ki provides additional flexibility. In this case, the consensus becomes a weighted average, enabling the final agreement point to be influenced by carefully tuning the gains.
