SC 627

Assignment 4 - Robotic Networks in Balancing

Nakul Randad | 180010047

Introduction

The assignment is based on robotic networks in balancing This is a multi-agent control and coordination problem as the action of one bot affects the others.

Algorithm

Setup

An implementation of motion balancing of a multiple robots is discussed for one-dimensional motion. A total of 8 robots are considered which are placed along the x-axis. The first at the eighth robots are placed at (0, 0) and (14, 0) respectively and the rest of them are placed in between. The objective is to balance all the robots such that they are all placed in an equidistant manner along the x-axis. The robot $R_i, \forall i \in \{2, 3, \cdots, 7\}$ at every instance of time knows the position of the robots R_{i-1} and R_{i+1}.

Control Law

Since all the bots are along the x-axis, we basically penalise a robot if it gets too close to its left or right neighbours. For this, we implement a simple proportional controller on the error or deviation. The control action for robot R_i is: $u_i = k \times (x_{i-1} - x_i) + k \times (x_{i+1} - x_i)$. This control is applied to the velocity. We see that, in the case of equilibrium where i is at the midpoint of its left and right neighbours $(x_{i-1} - x_i) = -(x_{i+1} - x_i)$, the commanded velocity (control) becomes zero and thus converges. We use k = 1 for faster convergence.

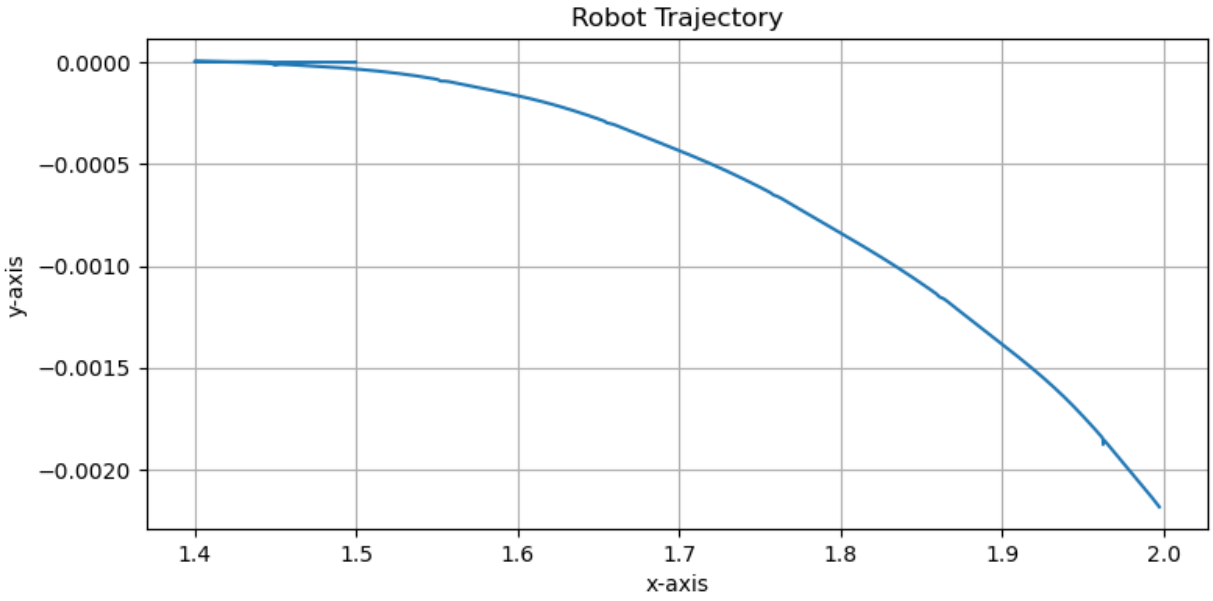
Termination

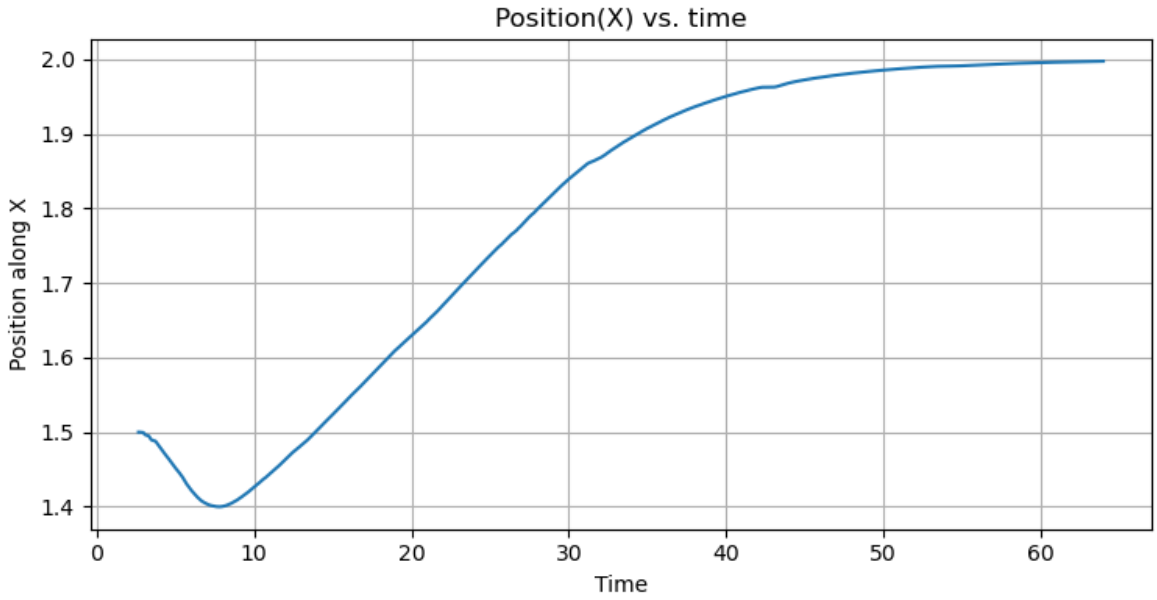
We terminate the algorithm once the velocities are close to zero and the robots begin to stabilize. To be precise, our exact stopping condition is when the absolute velocities of the robot and its immediate neighbors are within a δ bound.

Results and Observations

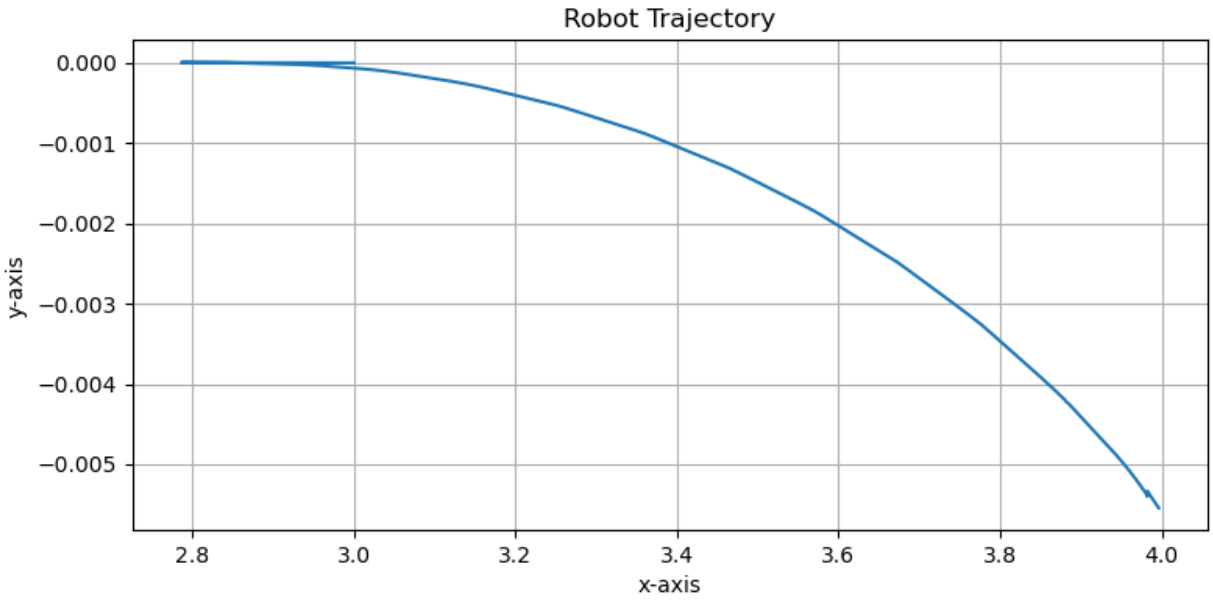
The above mentioned schemes can observed in the trajectory and movement along x-axis plots below:

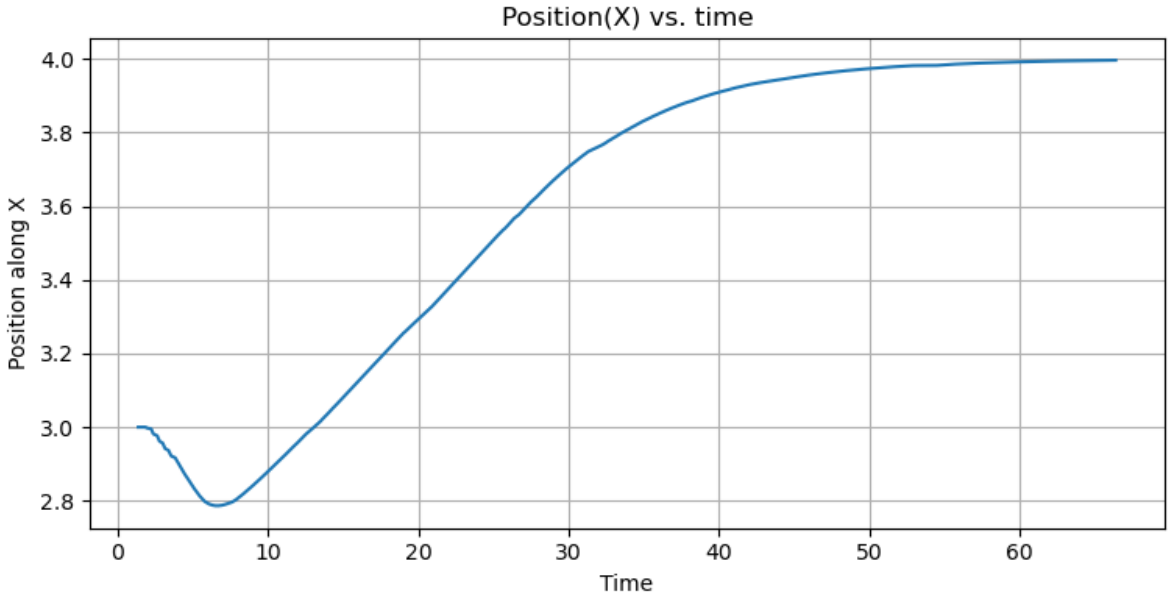
- Robot 2



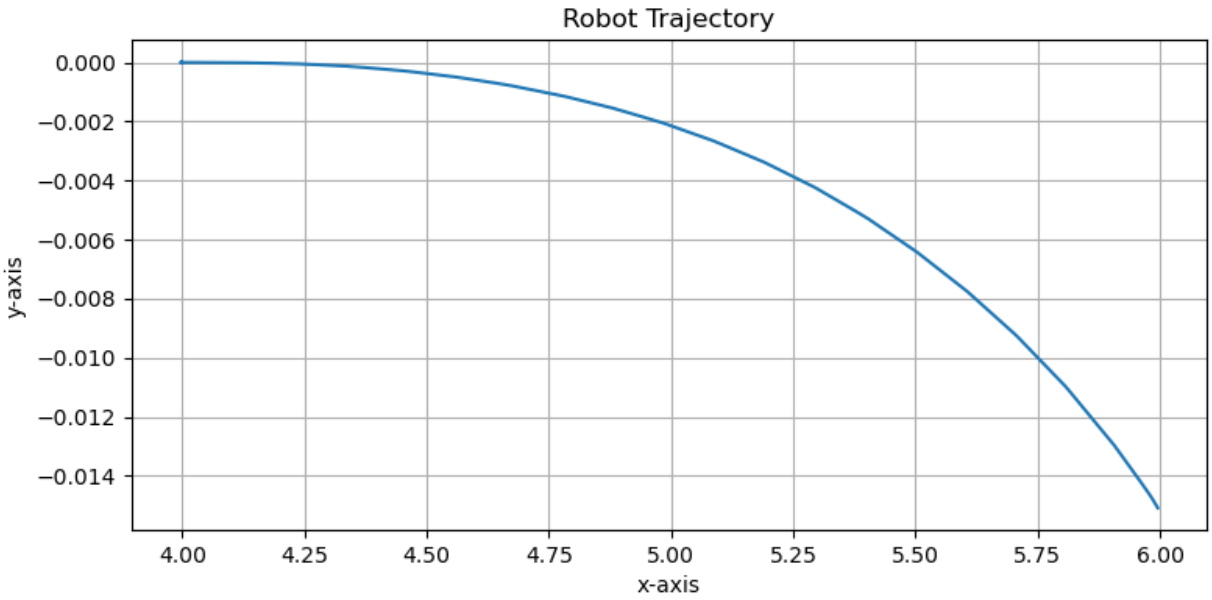


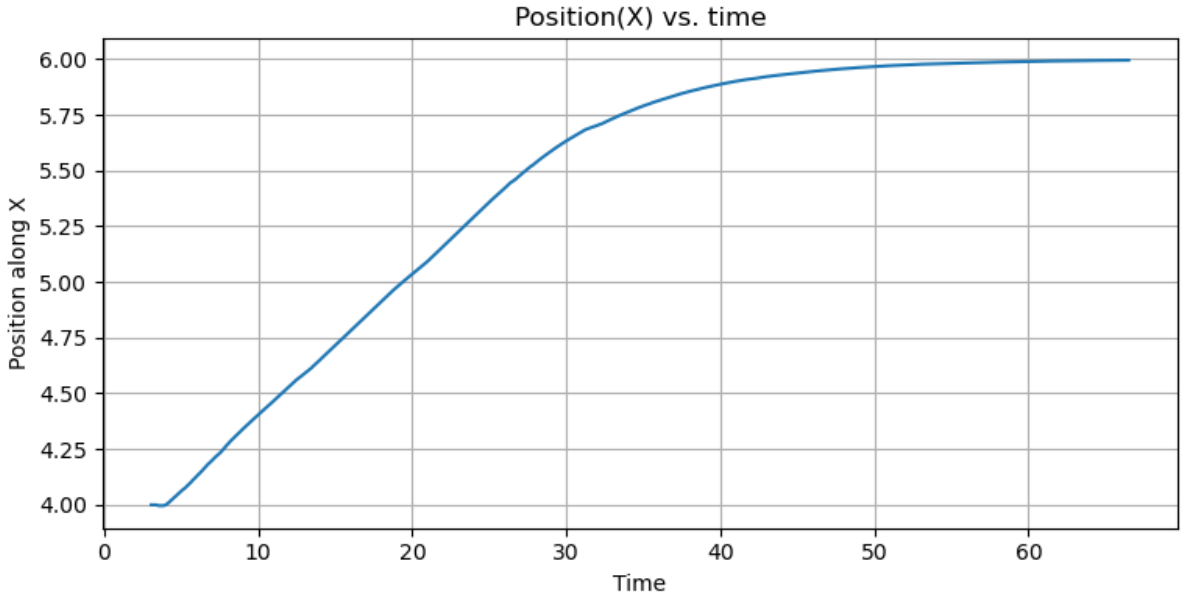
- Robot 3


Robot Trajectory
0.000
−0.001
−0.002
−0.003
−0.004
−0.005
y-axis
2.8
3.0
3.2
3.4
3.6
3.8
4.0
x-axis

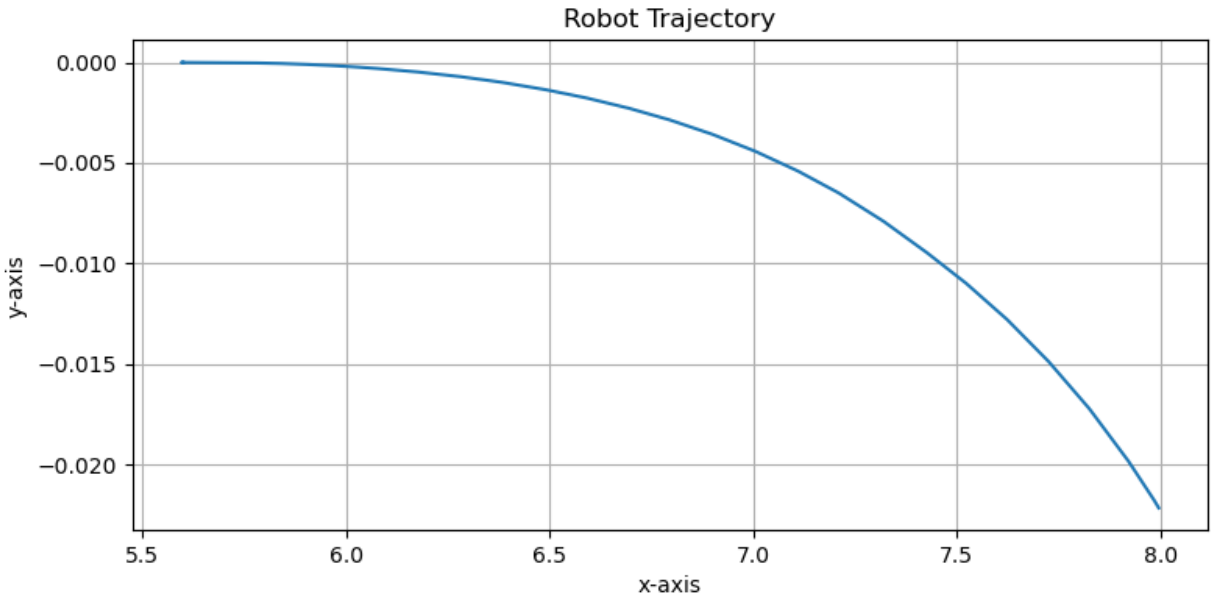

Position(X) vs. time
4.0
3.8
3.6
3.4
3.2
3.0
2.8
Position along X
0
10
20
30
40
50
60
Time

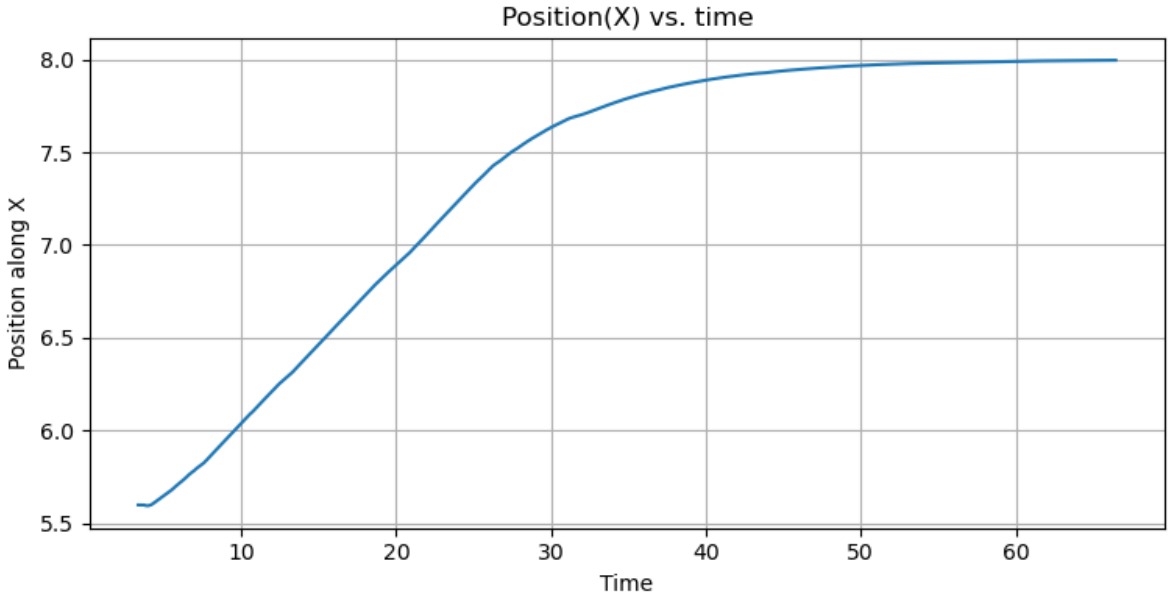
- Robot 4


Robot Trajectory
y-axis
x-axis
0.000
−0.002
−0.004
−0.006
−0.008
−0.010
−0.012
−0.014
4.00
4.25
4.50
4.75
5.00
5.25
5.50
5.75
6.00

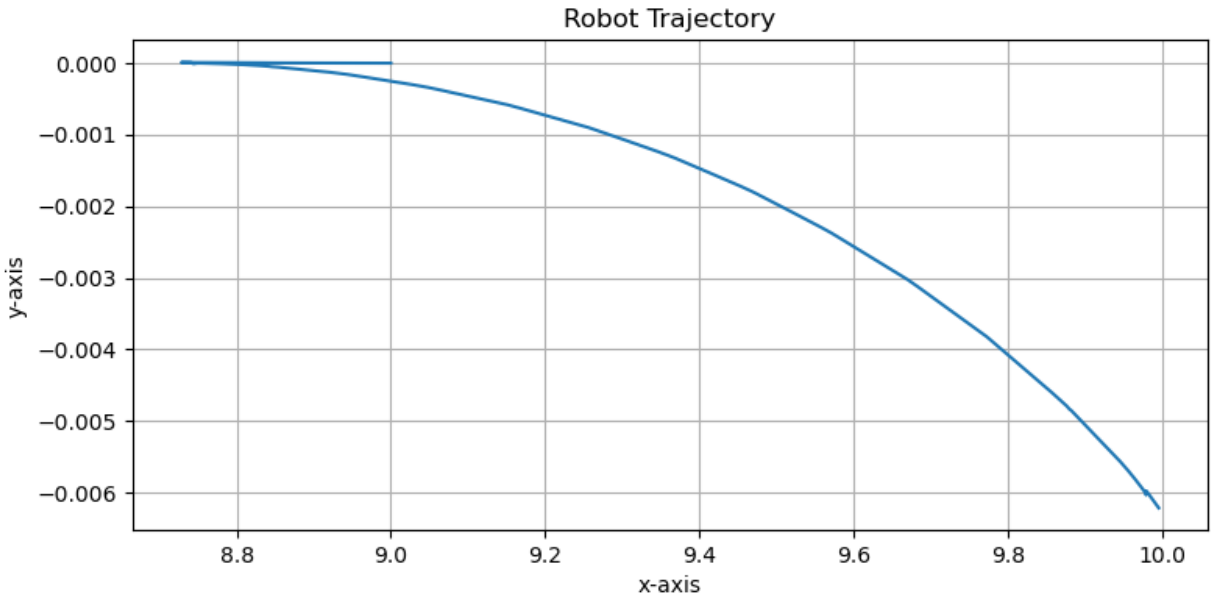

Position(X) vs. time
Position along X
Time
6.00
5.75
5.50
5.25
5.00
4.75
4.50
4.25
4.00
0
10
20
30
40
50
60

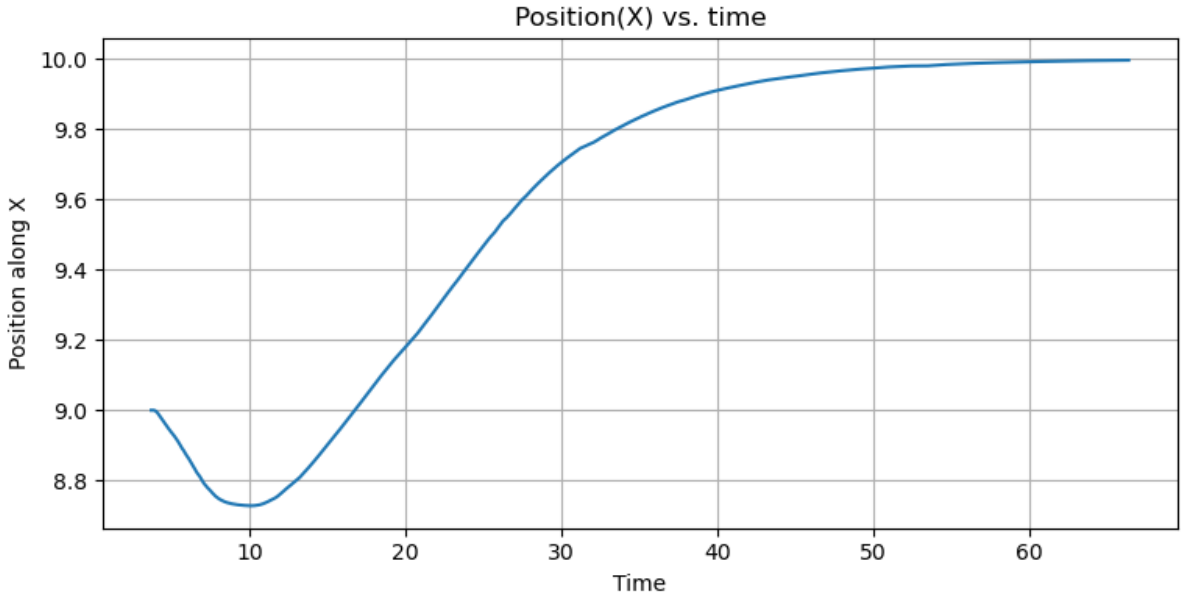
- Robot 5


Robot Trajectory
0.000
−0.005
−0.010
−0.015
−0.020
y-axis
5.5
6.0
6.5
7.0
7.5
8.0
x-axis

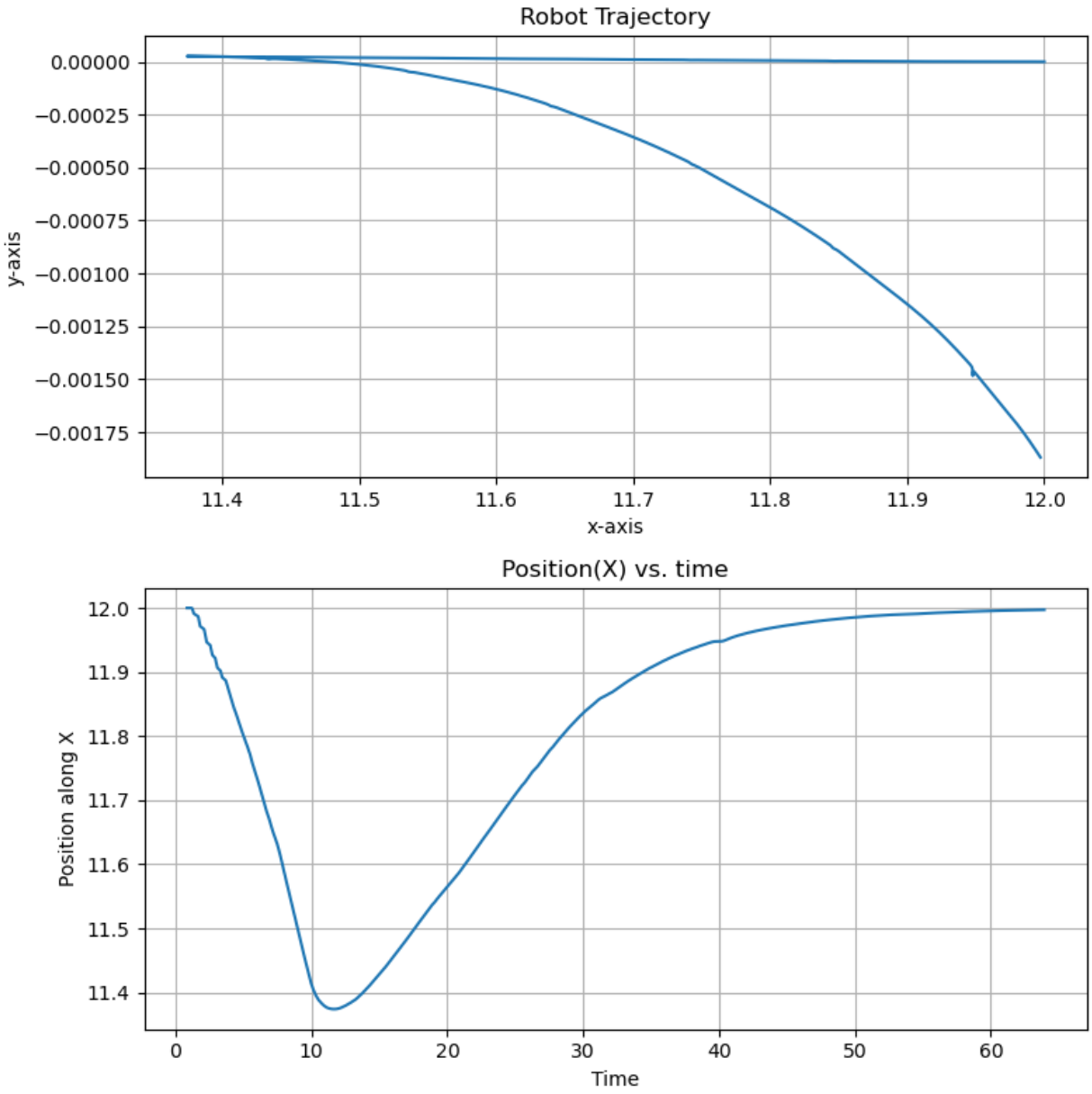

Position(X) vs. time
8.0
7.5
7.0
6.5
6.0
5.5
Position along X
10
20
30
40
50
60
Time

- Robot 6


Robot Trajectory
0.000
−0.001
−0.002
−0.003
−0.004
−0.005
−0.006
y-axis
8.8
9.0
9.2
9.4
9.6
9.8
10.0
x-axis


Position(X) vs. time
10.0
9.8
9.6
9.4
9.2
9.0
8.8
Position along X
10
20
30
40
50
60
Time

- Robot 7



The results show that the positions converge such that all the robots are placed in equidistant manner i.e. towards {2,4,6,8,10,12} for robots {2,3,4,5,6,7} respectively.