Assignment 4

Name: Ashay Wakode
Roll no: 170100033

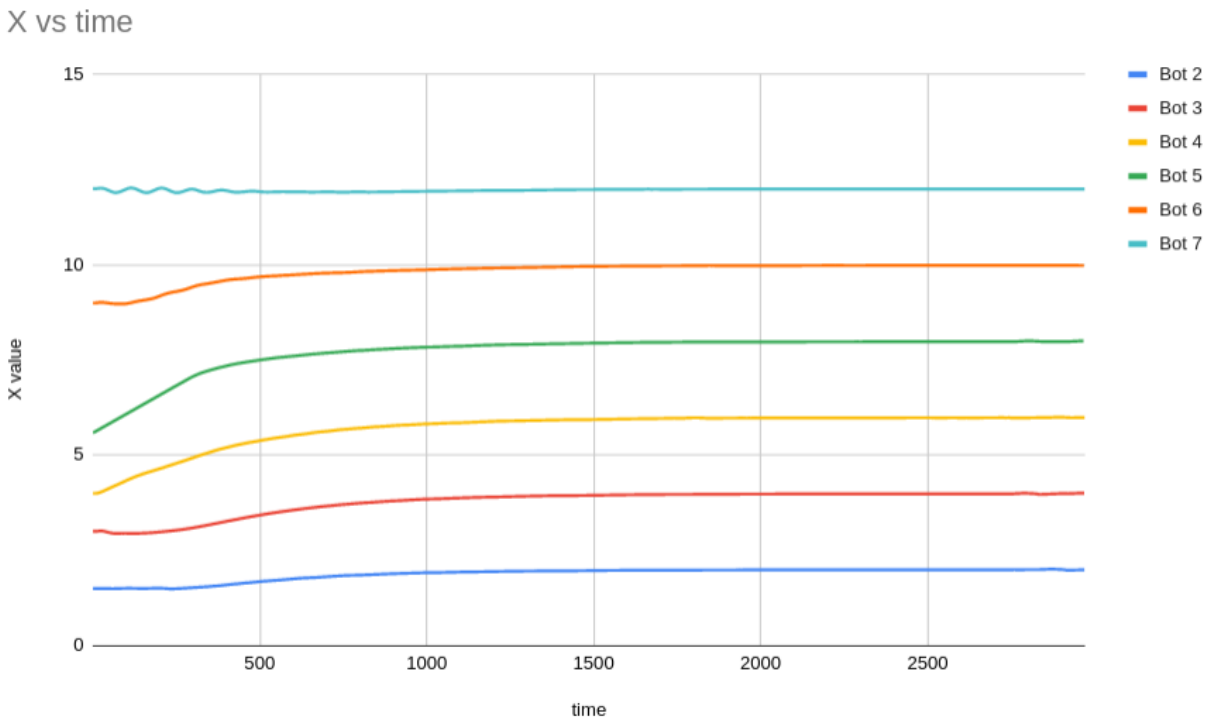
● Details of Implementation :

Given: Left robot x and y position, Right robot x and y position
The V_x was calculated as,

V_{xi} = (x_left+x_right)/2-xi , i is the robot number

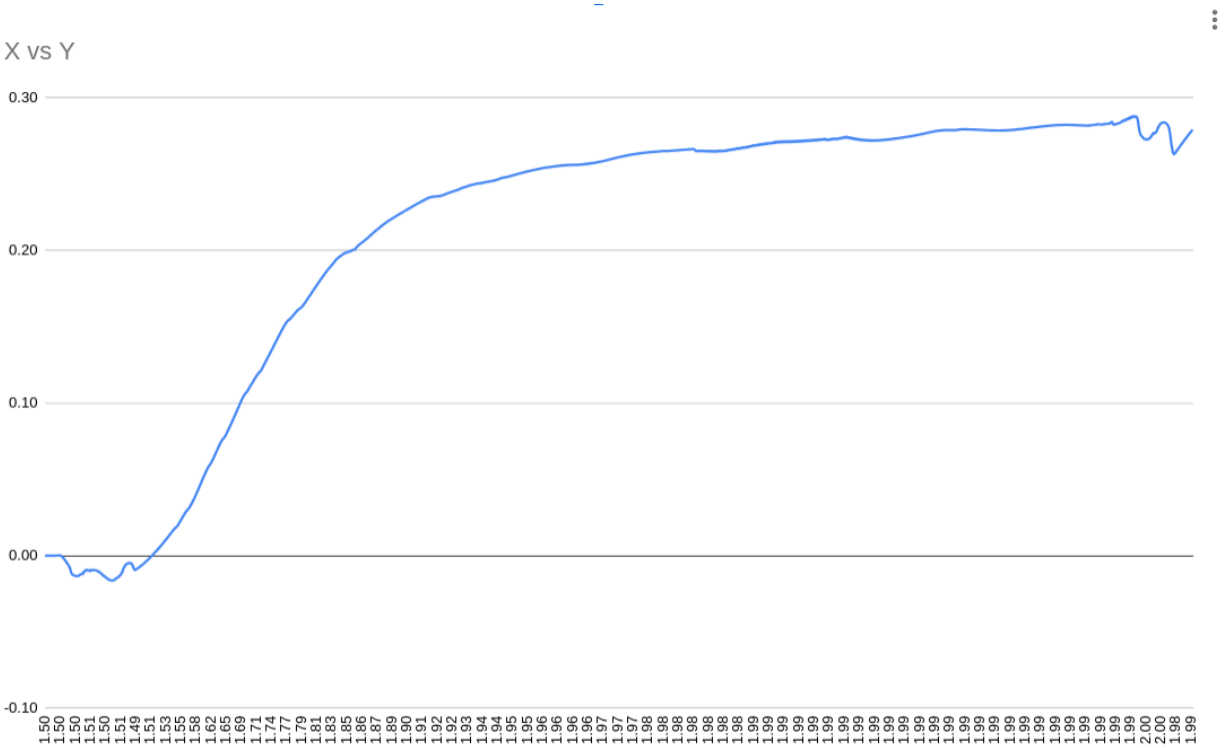
The V_y is taken to be zero since coordination in the y-direction is not intended.
This (V_x,V_y) is given to the velocity convert function, to be converted into linear and angular velocities. The limit of angular rotation of pi/12 was relaxed for faster implementation. Also, the gain_ang was changed to 2 for higher angular velocities of the robot. The output of the velocity convert function was published onto “Twist” topic, which acts like linear and angular velocity instructions to the individual robot. The values of x location, y location, and time stamp were saved in separate files, using rospy.get_namespace() as an identifier. The time stamps were different for different robots, so a unique time identifier array was generated for plotting the X vs time graph, the graph is attached below. The X vs Y graphs are also generated and are attached below. The while loop in the script is not altered because, the robot will simply cease to move once they become equidistant, due to the consensus algorithm used. Instead, the simulation was run for 12-15 mins for robots to settle and show very little moment. The data was saved for each bot in different text files which were fetched to generate the plots attached below.

- Plots
- X vs time

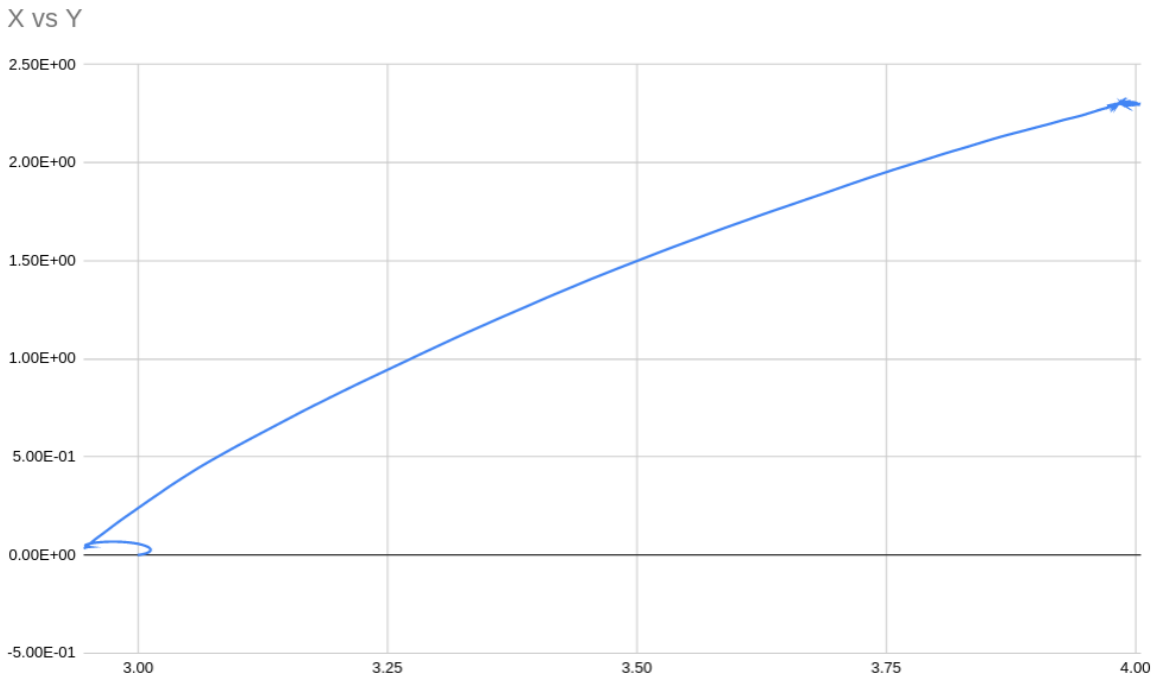


X vs time

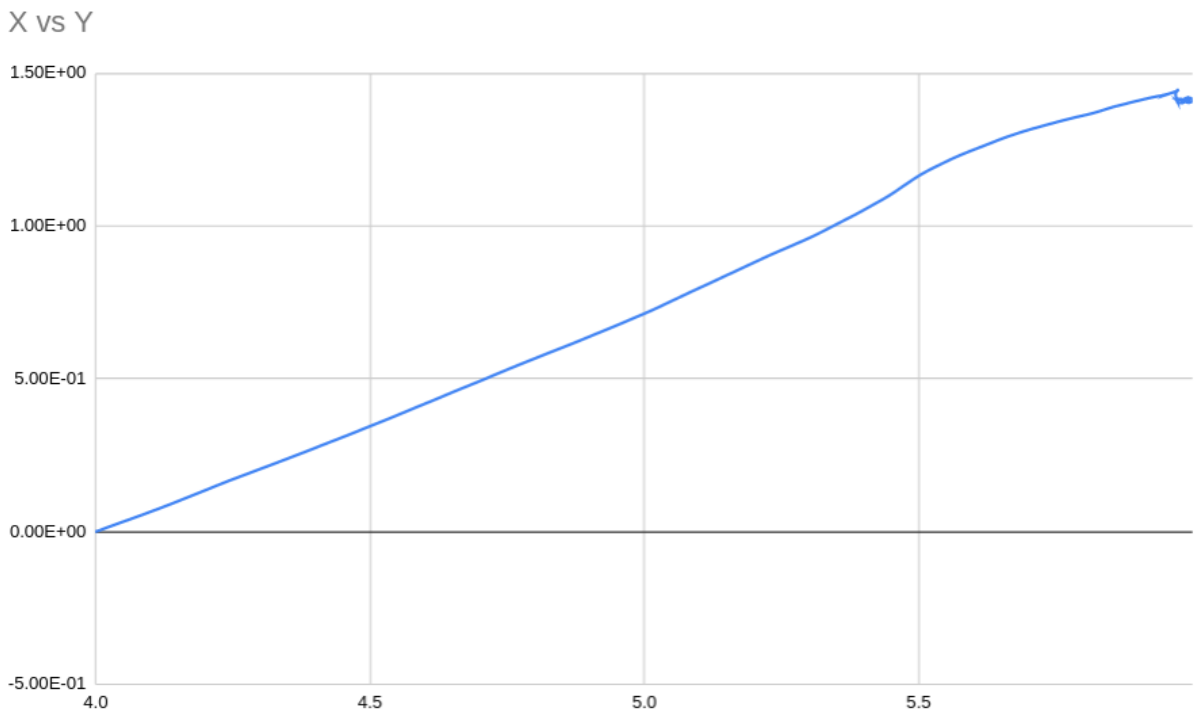
- X vs Y for each bot :



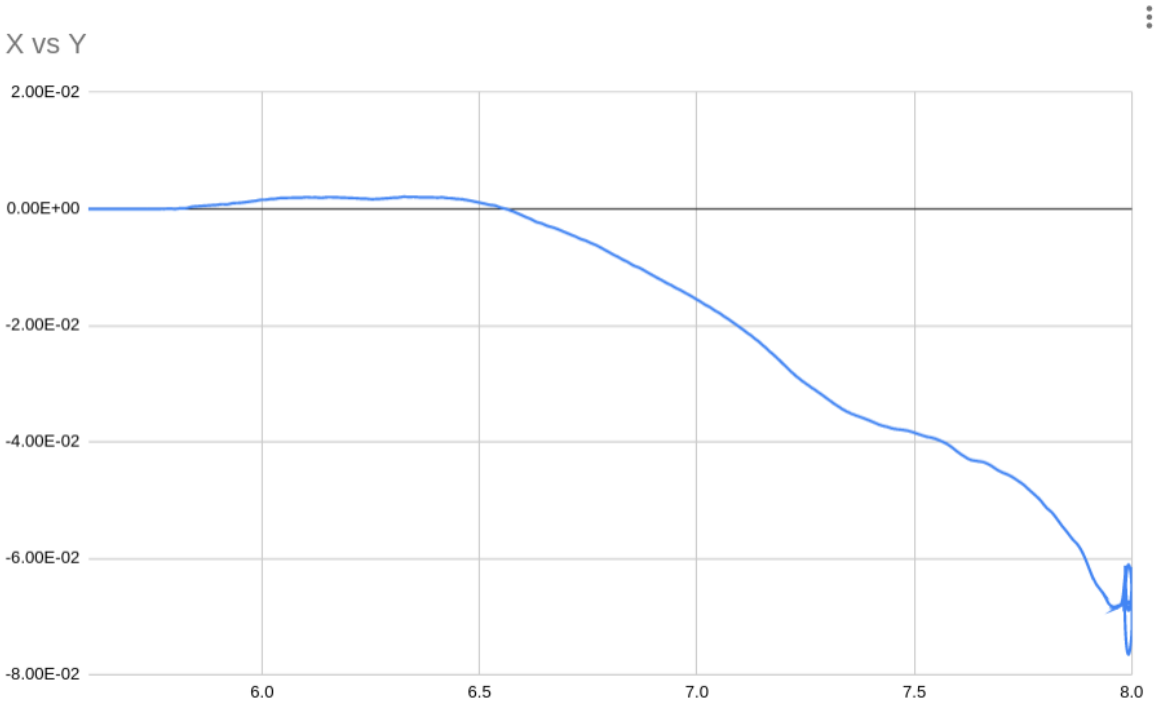
Bot 2



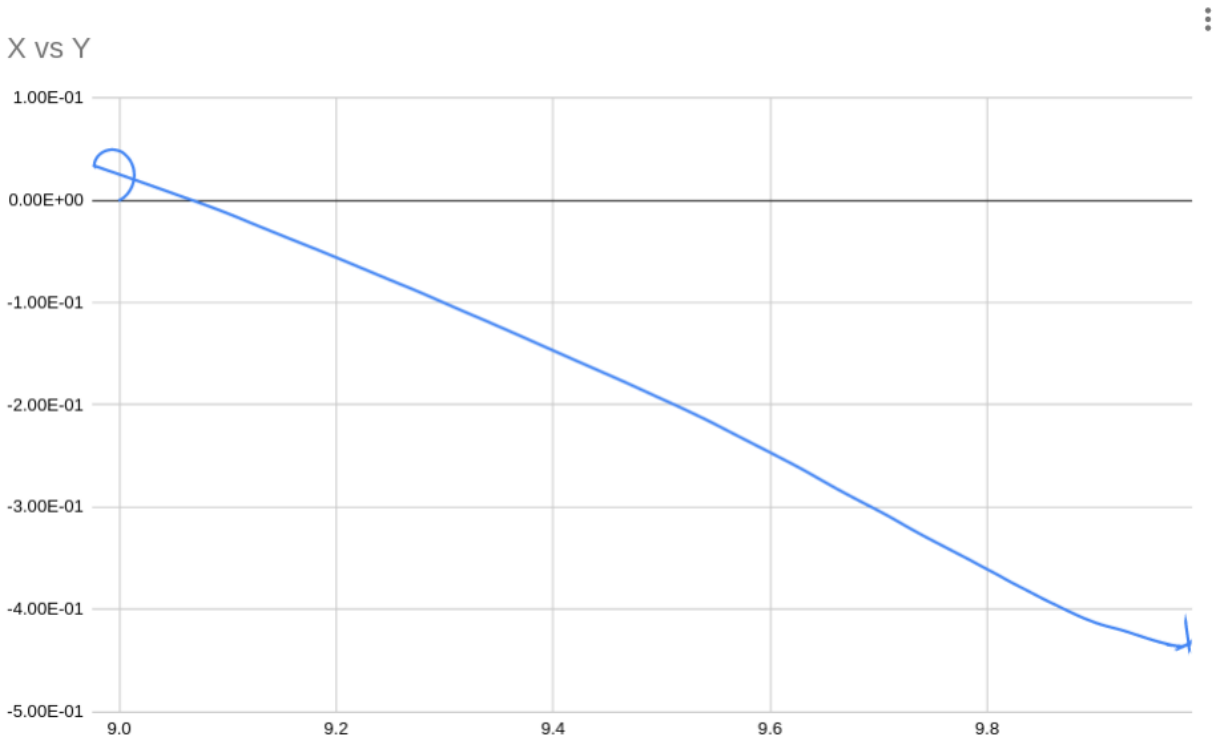
Bot 3


X vs Y
1.50E+00
1.00E+00
5.00E-01
0.00E+00
-5.00E-01
4.0
4.5
5.0
5.5

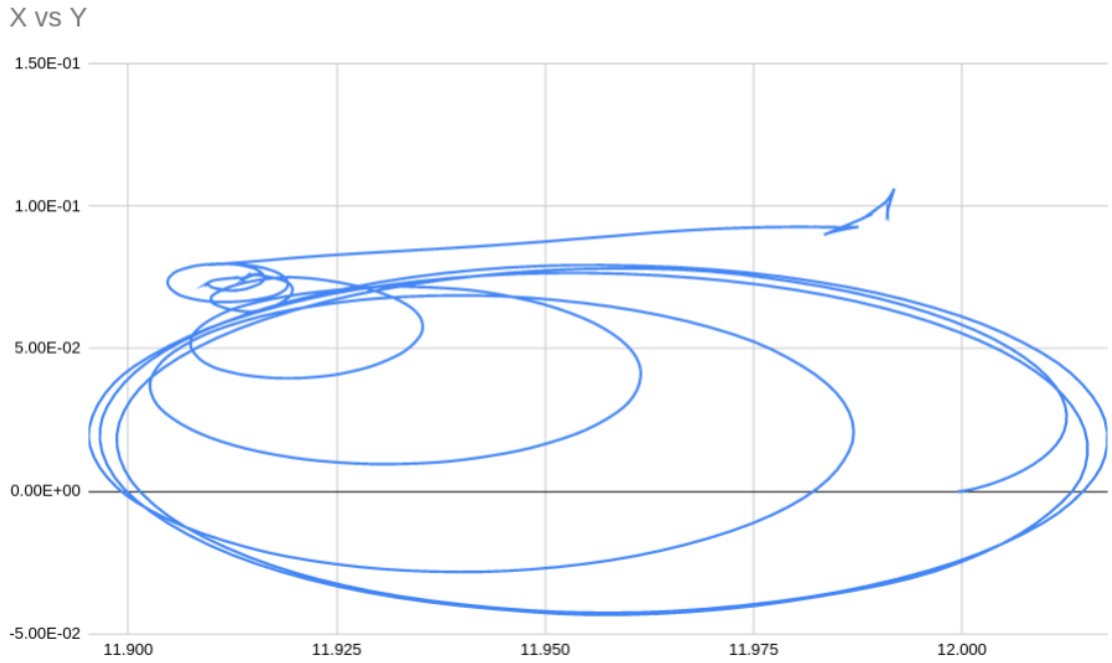
Bot 4


X vs Y
2.00E-02
0.00E+00
-2.00E-02
-4.00E-02
-6.00E-02
-8.00E-02
6.0
6.5
7.0
7.5
8.0

Bot 5


X vs Y
1.00E-01
0.00E+00
-1.00E-01
-2.00E-01
-3.00E-01
-4.00E-01
-5.00E-01
9.0
9.2
9.4
9.6
9.8

Bot 6


X vs Y
1.50E-01
1.00E-01
5.00E-02
0.00E+00
-5.00E-02
11.900
11.925
11.950
11.975
12.000

Bot 7