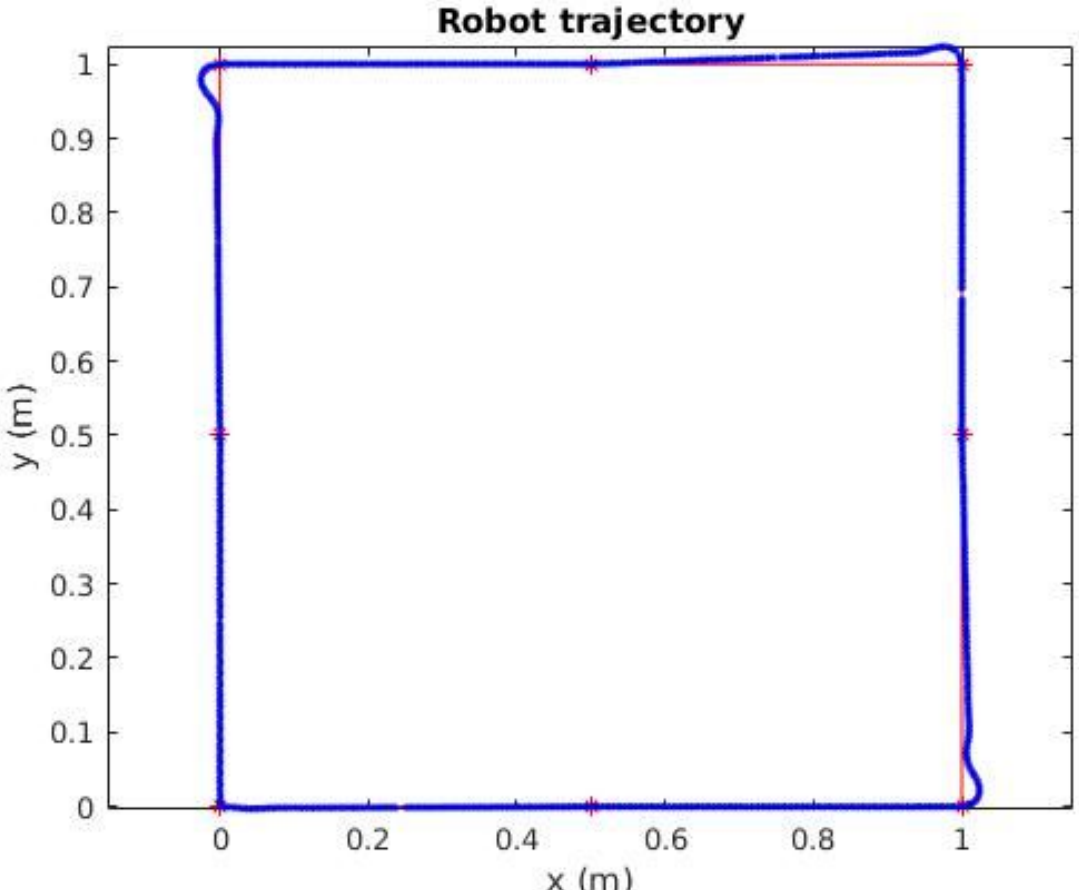
EE144 Introduction to Robotics

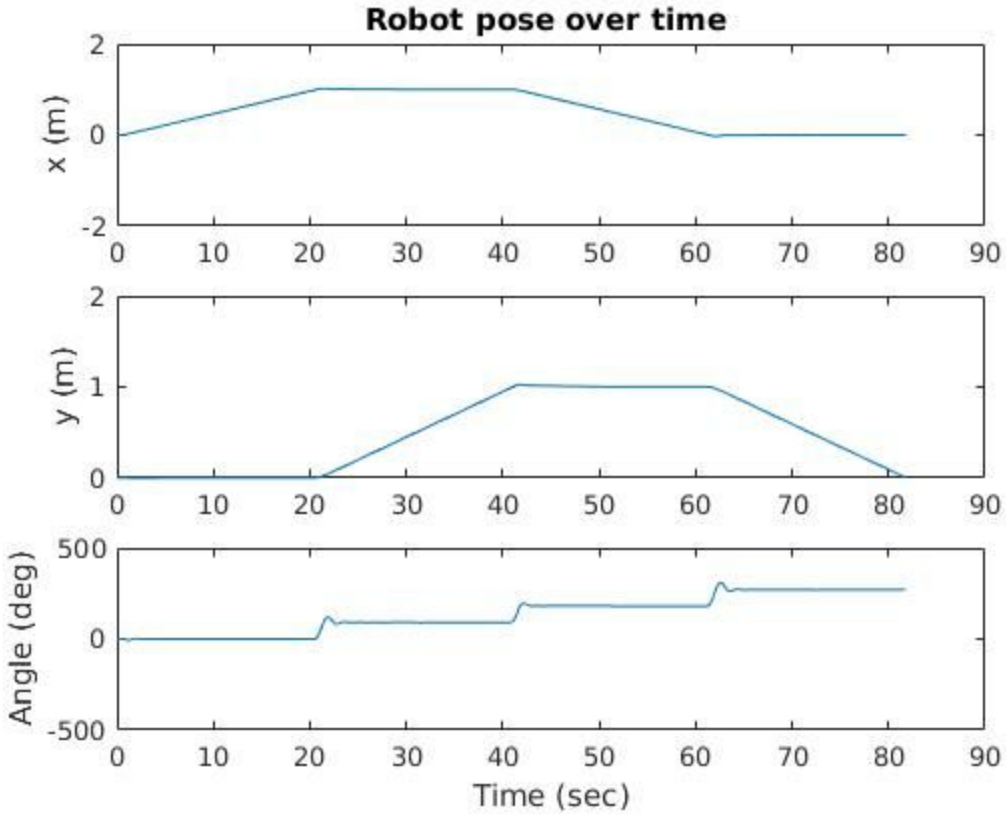
Lab 2: Waypoint Navigation

Yecheng Xiang (X640545)

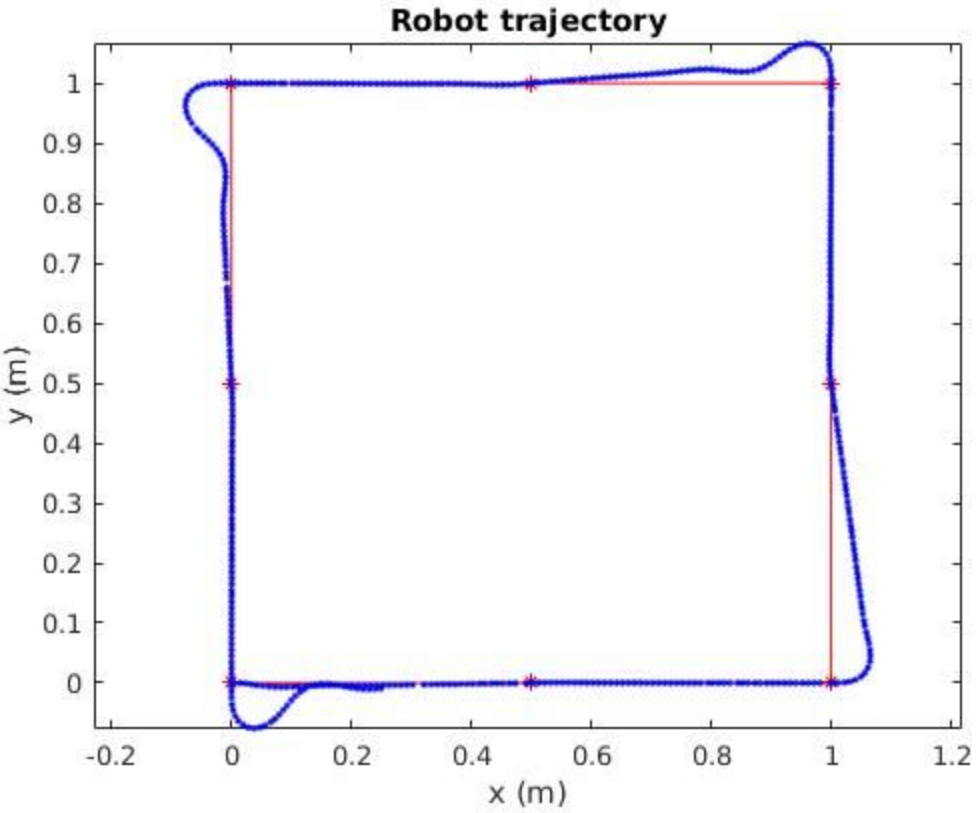
Part 1: The trajectory with different V

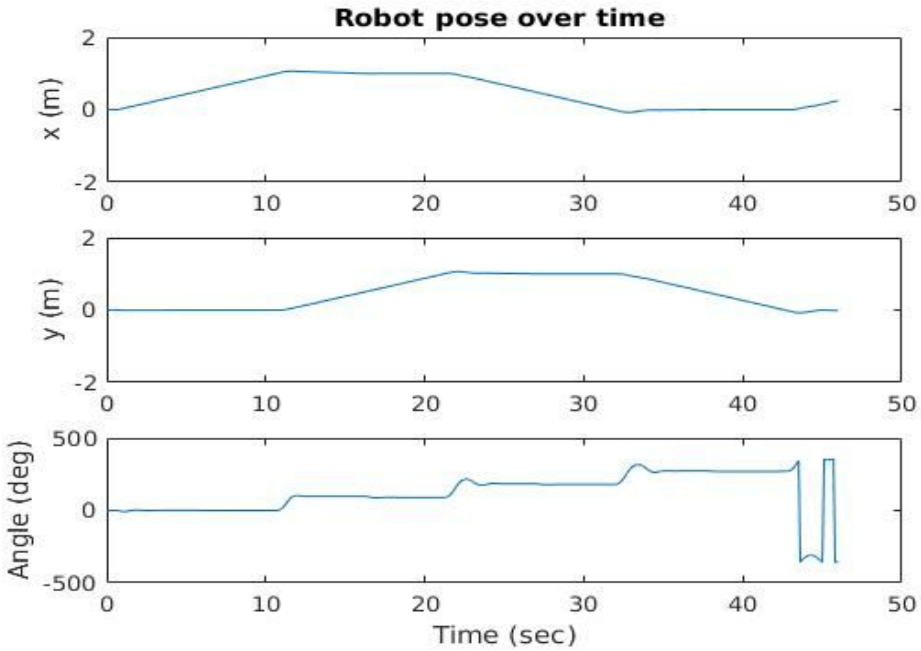
1) **Kp = 2 V = 0.05**



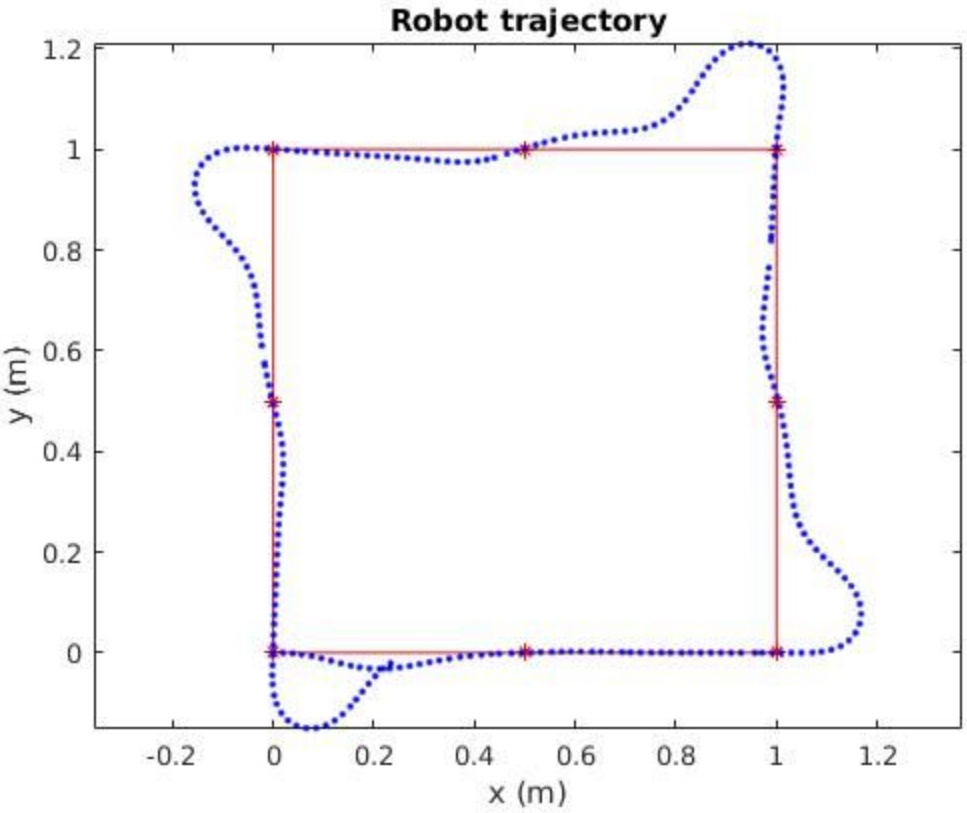

Robot pose over time
x (m)
y (m)
Angle (deg)
Time (sec)
2
0
-2
1
500
-500
0
10
20
30
40
50
60
70
80
90

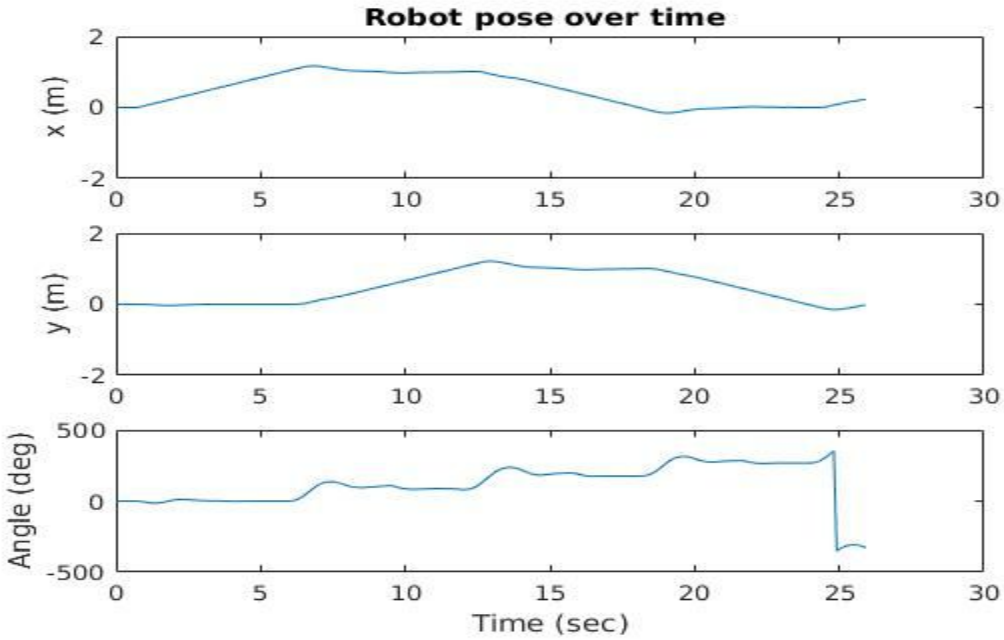
2) Kp = 2 V = 0.10


Robot trajectory
y (m)
x (m)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.2
0
0.2
0.4
0.6
0.8
1
1.2

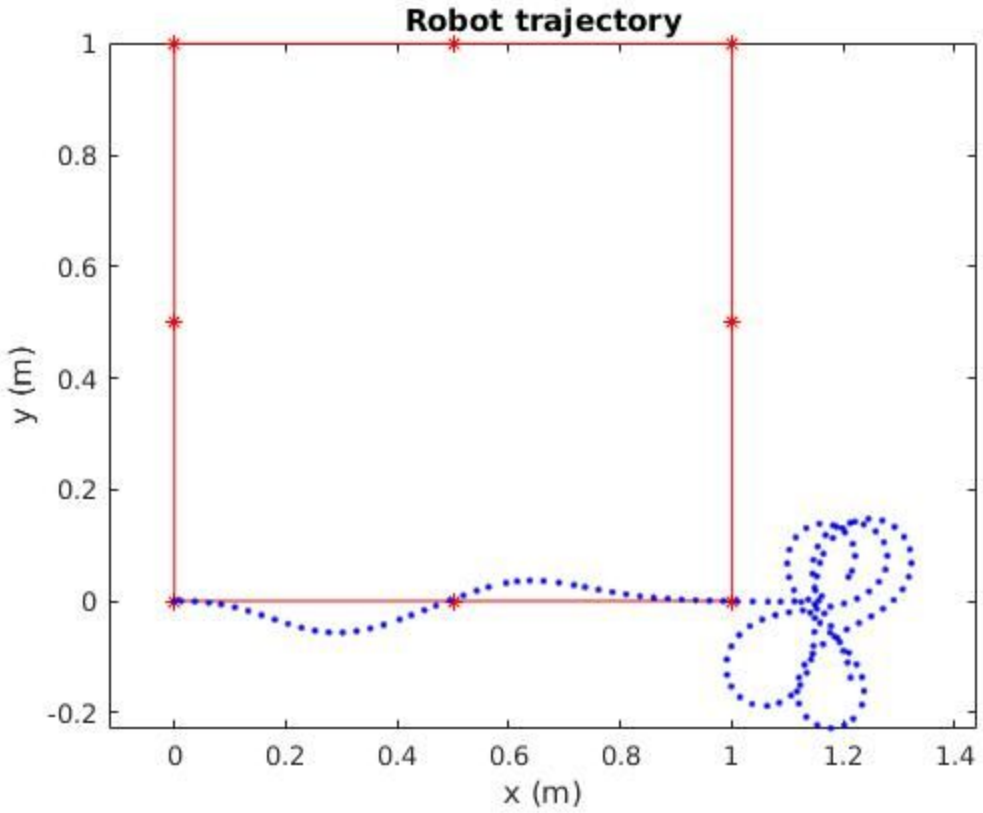

Robot pose over time
x (m)
y (m)
Angle (deg)
Time (sec)
2
0
-2
500
-500
0
10
20
30
40
50

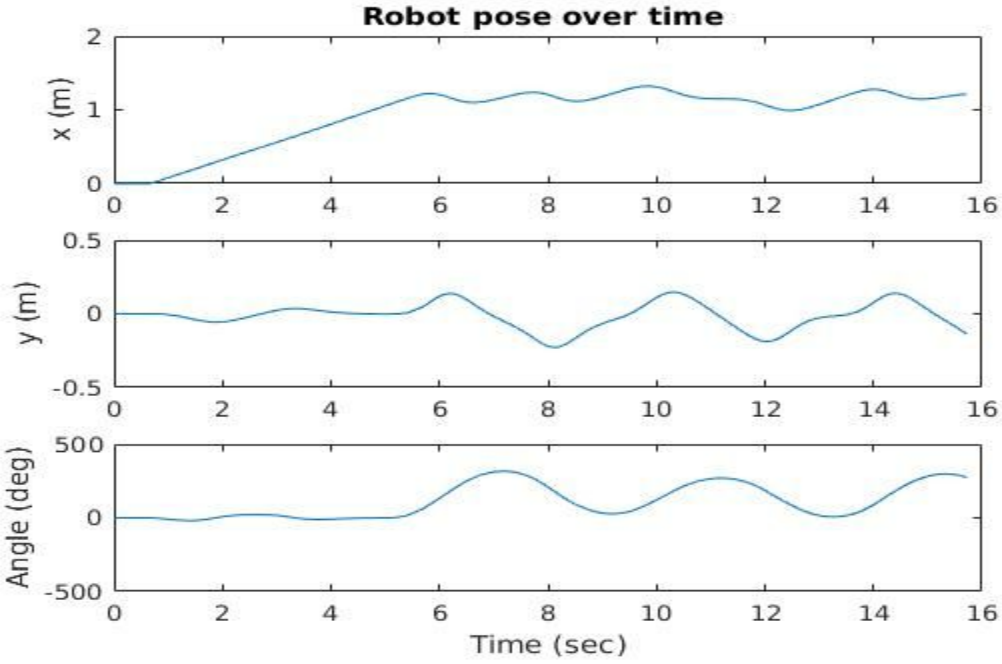
3) Kp = 2 V = 0.2


Robot trajectory
y (m)
x (m)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1
1.2


Robot pose over time
x (m)
y (m)
Angle (deg)
Time (sec)
2
0
-2
500
-500
0
5
10
15
20
25
30

4) Kp = 2 V = 0.25 (fail to trace the points)



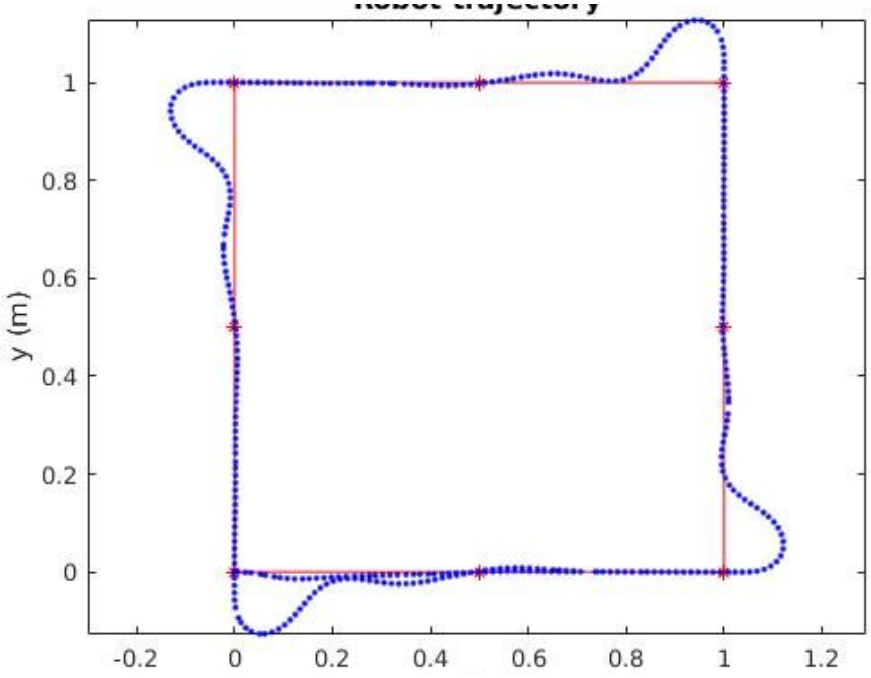


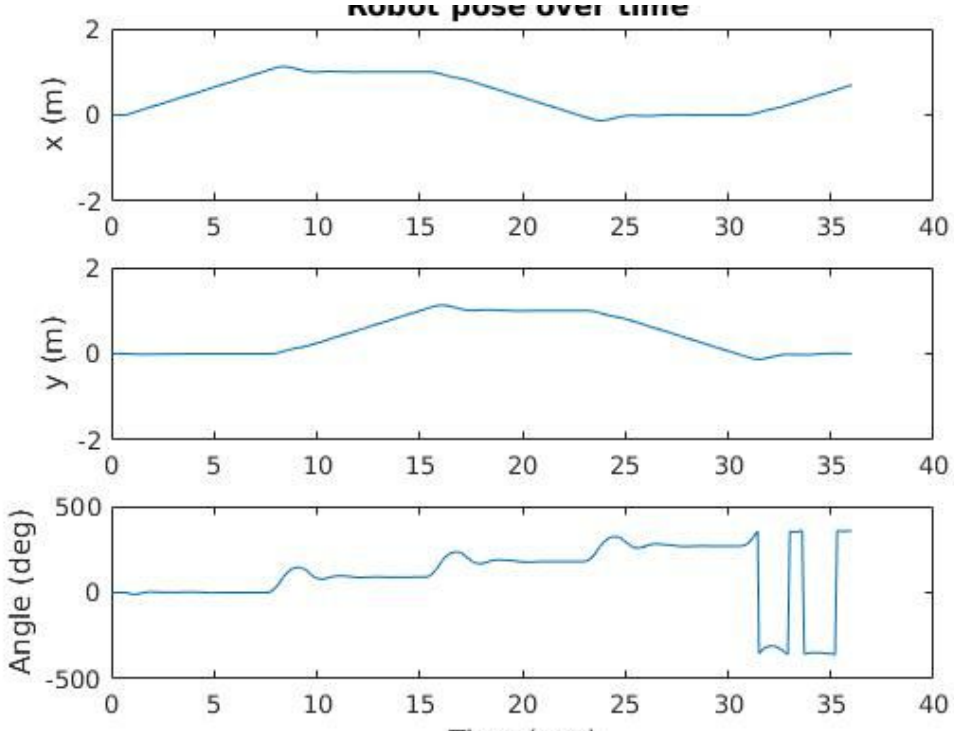
Discussions

- From the above result, we can clearly see that with the V growing, the trajectory of the robot is farther from the desired one. The reason is simple: with the speed growing and the response time for the P control remaining the same, the robot travels farther at the corner as the velocity multiplied by responding time is larger, which means a larger distance.
- The maximum velocity works for my algorithm is 0.2. As shown in the 4) section, if the velocity reaches 0.25, the algorithm does not work anymore.

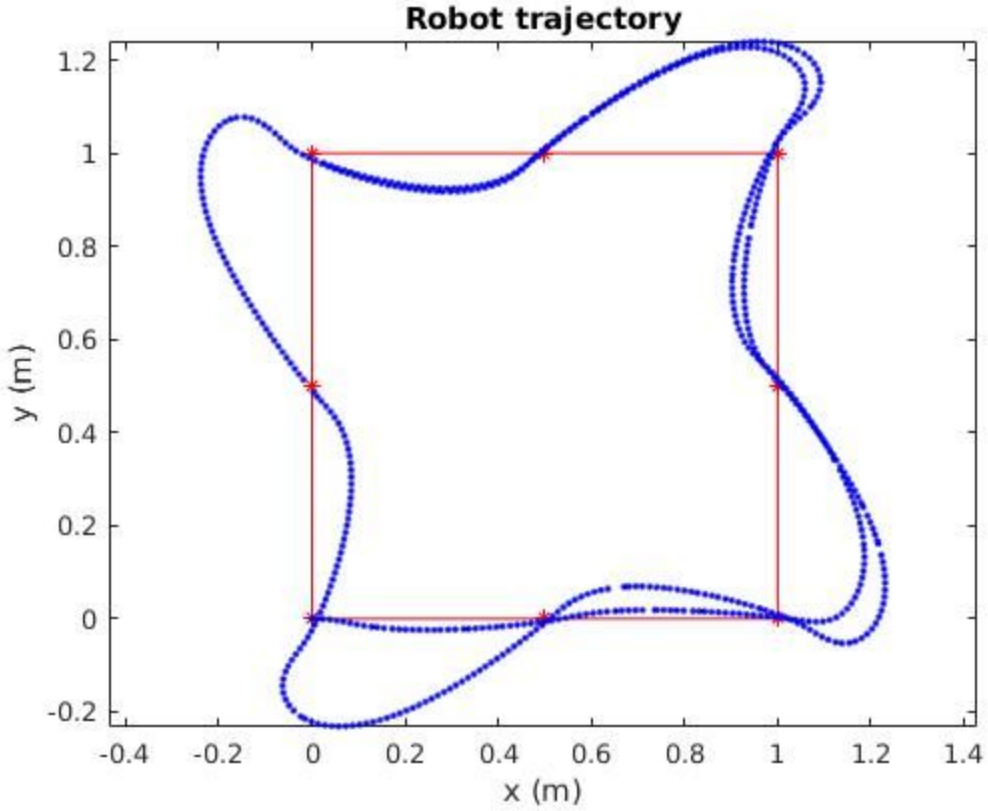
Part 2: The trajectory with different Kp with constant V

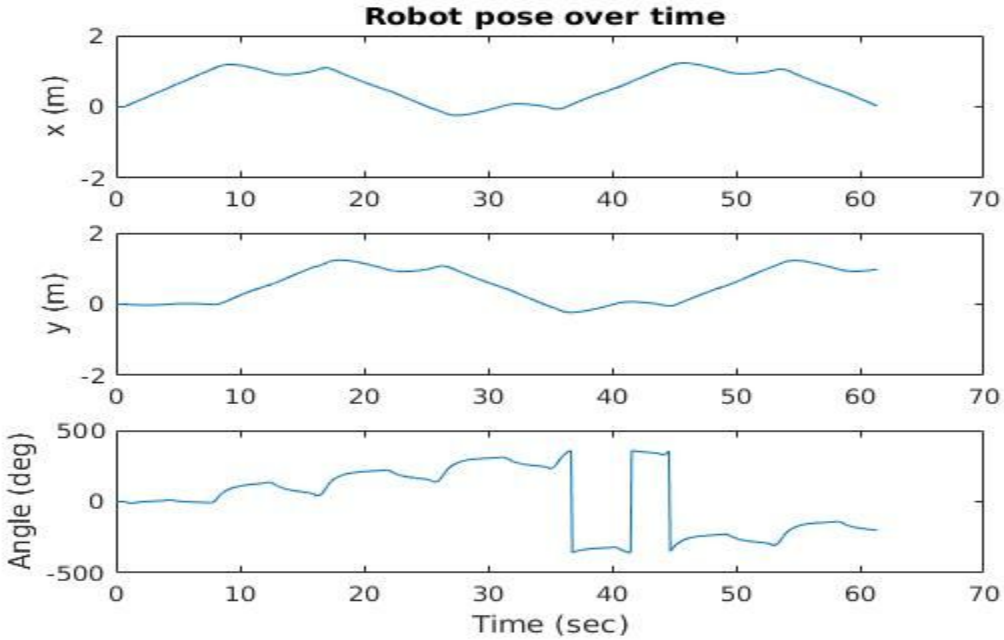
1) Kp = 2 V = 0.15



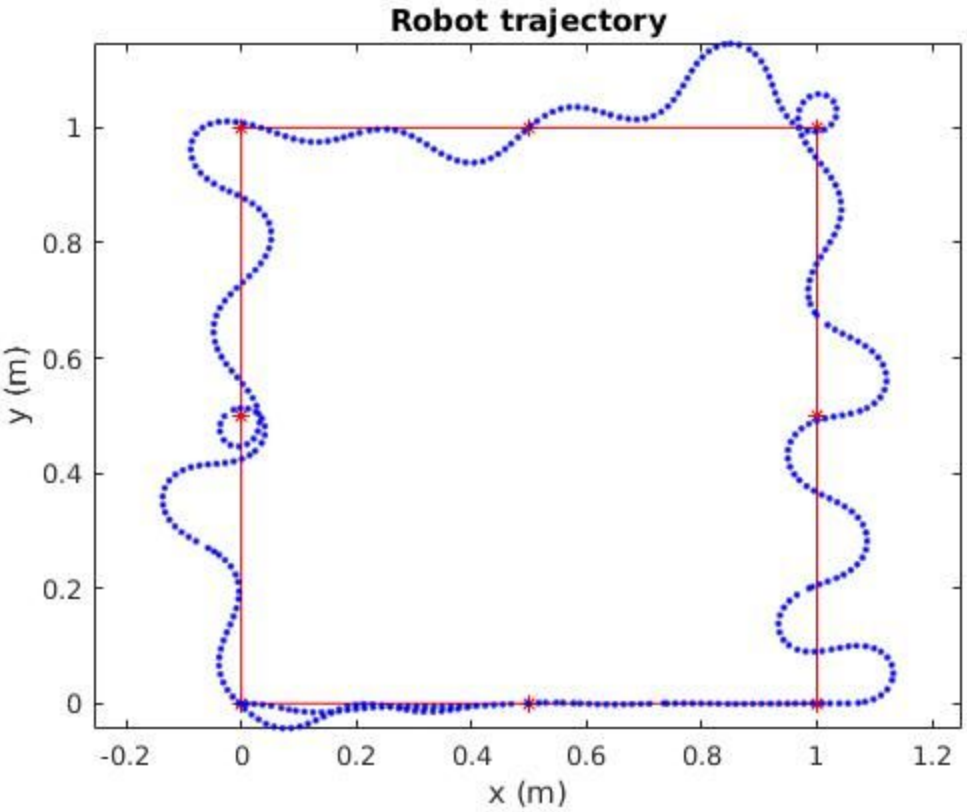


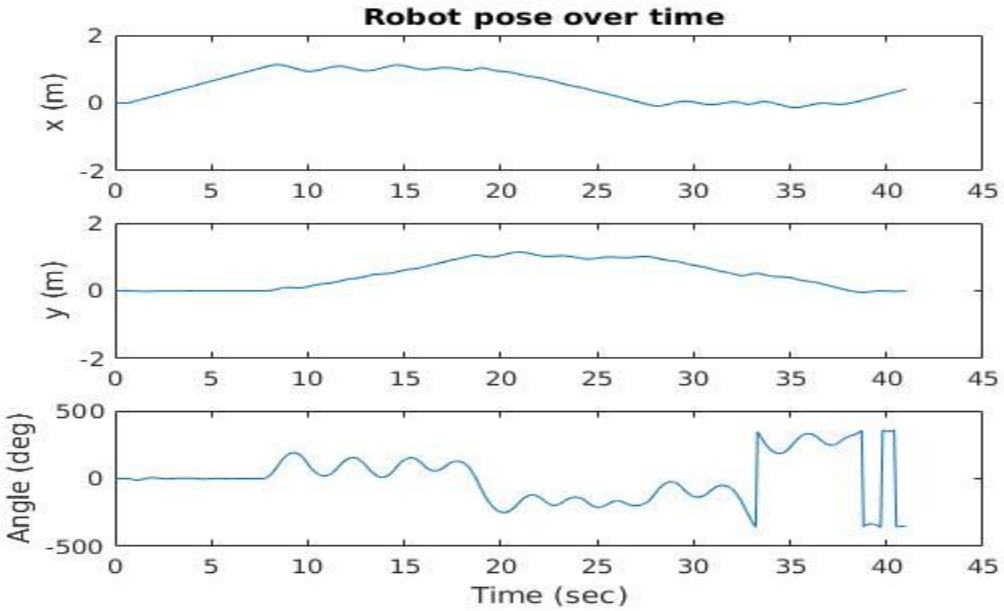
2) Kp = 1 V = 0.15


Robot trajectory
y (m)
x (m)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4


Robot pose over time
x (m)
y (m)
Angle (deg)
Time (sec)
2
0
-2
500
-500
0
10
20
30
40
50
60
70

3) Kp = 3 V = 0.15

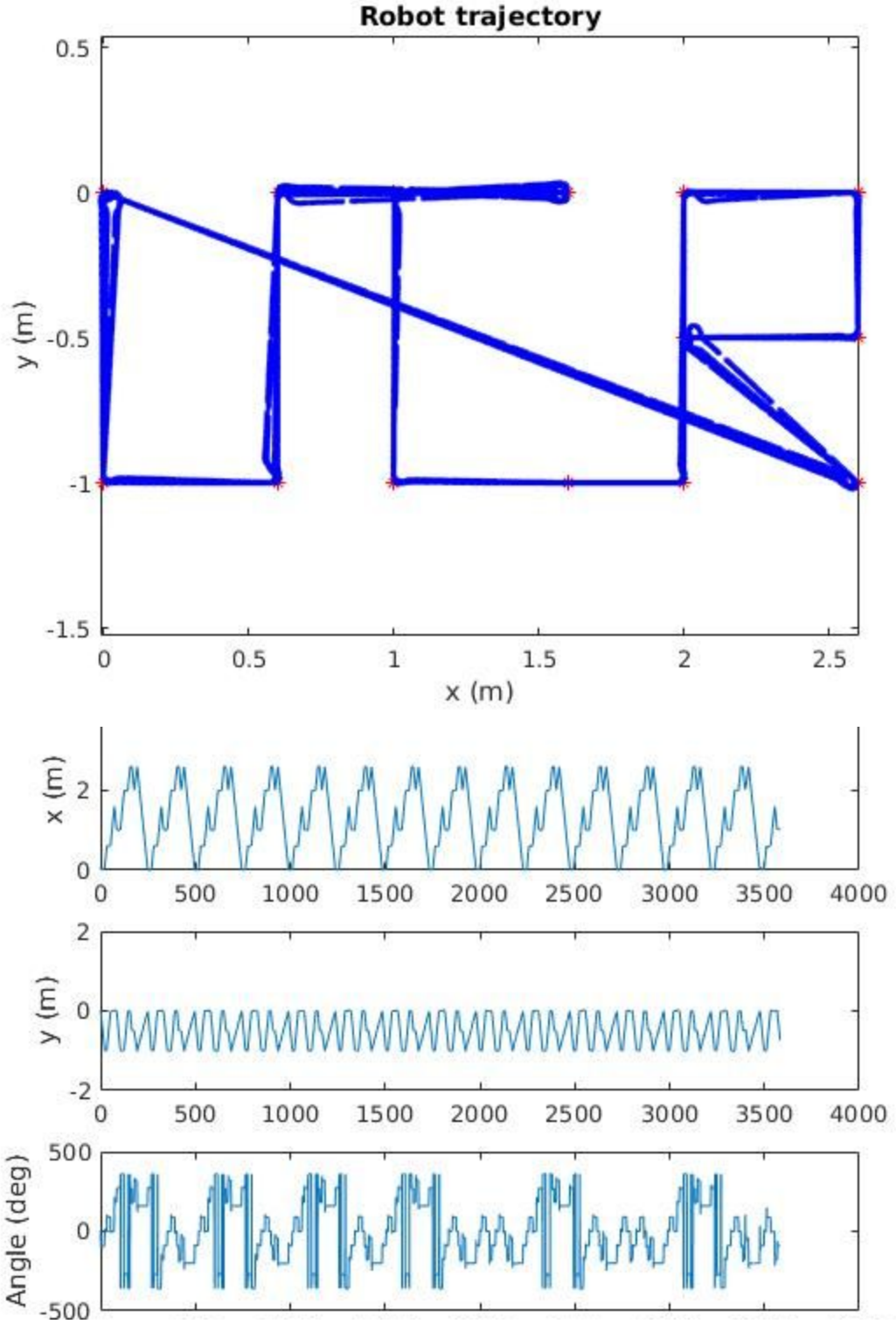

Robot trajectory
y (m)
x (m)
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1
1.2


Robot pose over time
x (m)
y (m)
Angle (deg)
Time (sec)
2
0
-2
500
-500
0
5
10
15
20
25
30
35
40
45

Discussions

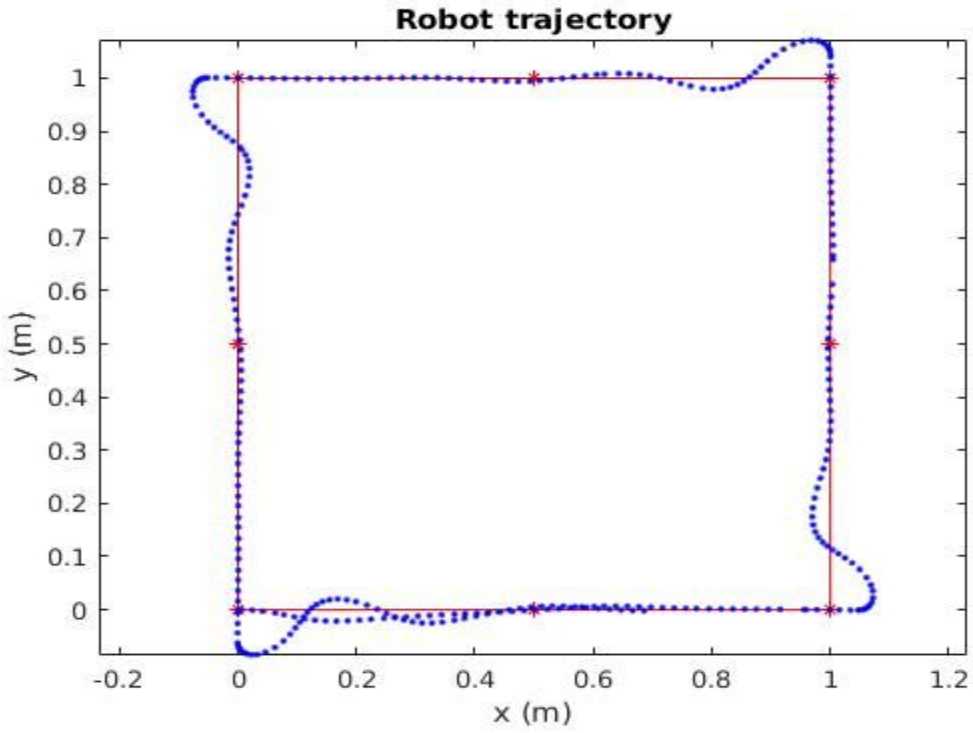
- The above plots is showing that when a larger Kp will decrease the responding time at the corner, but will bring in larger fluctuation. Moreover, if the Kp is too large, the algorithm will fail, as the robot will always fluctuating and does work any more.
- A smaller Kp will increase the responding time for the robot at a turning corner, which may lead to failure of the algorithm with a large velocity.
- Kp is a critical parameter of the algorithm, a nice Kp can allow the robot to travel at a significant speed.

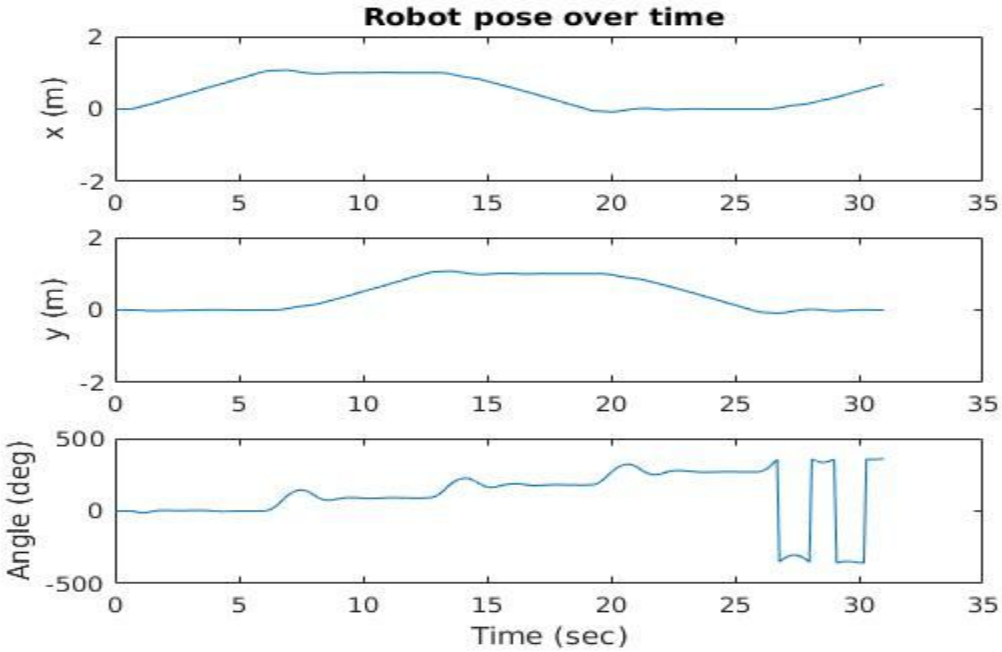
PART3. Proper Parameters for the Algorithm to Draw 'UCR'



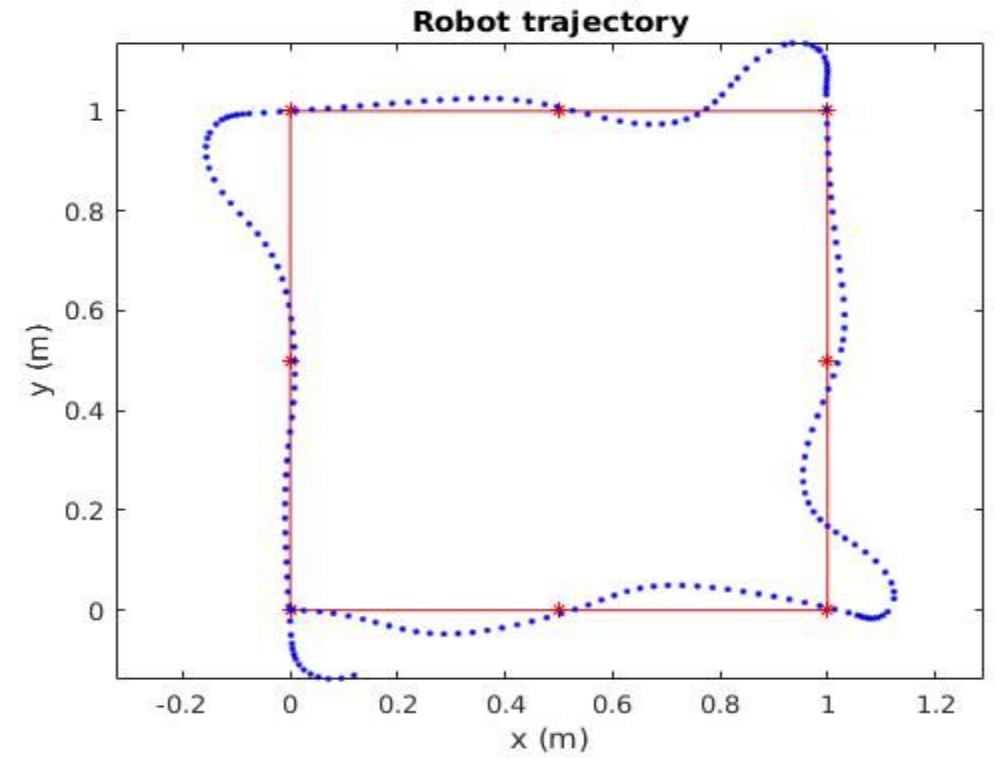
- **PART4. The trajectory with V a function of rotation velocity**

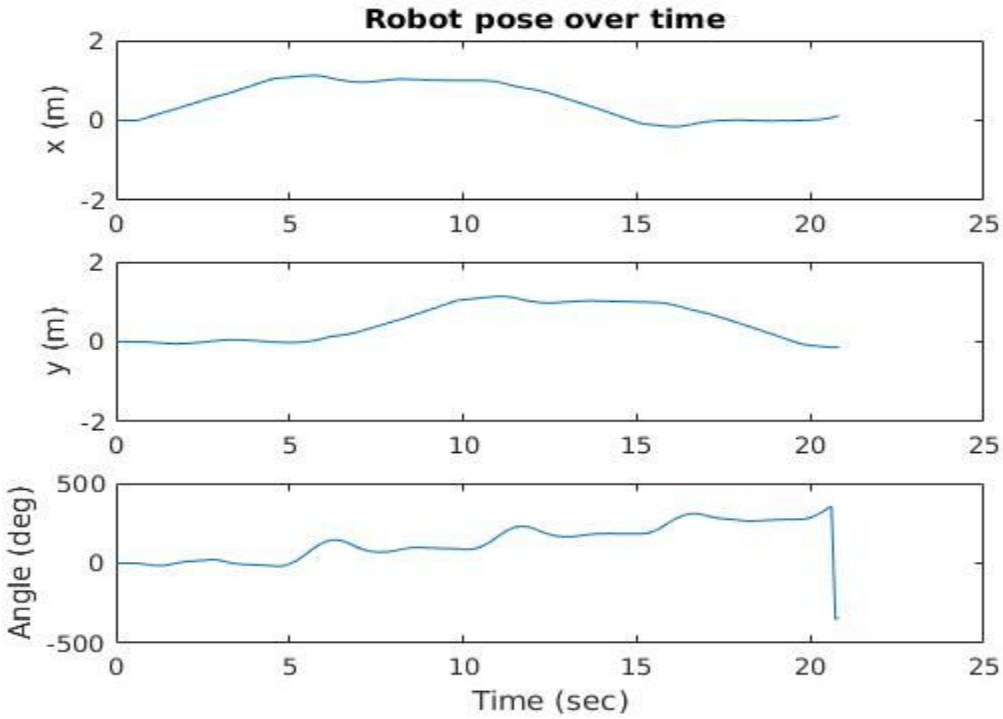
1) **Kp = 2 V = abs(0.2 - 0.04 * abs(rotation_vel))**



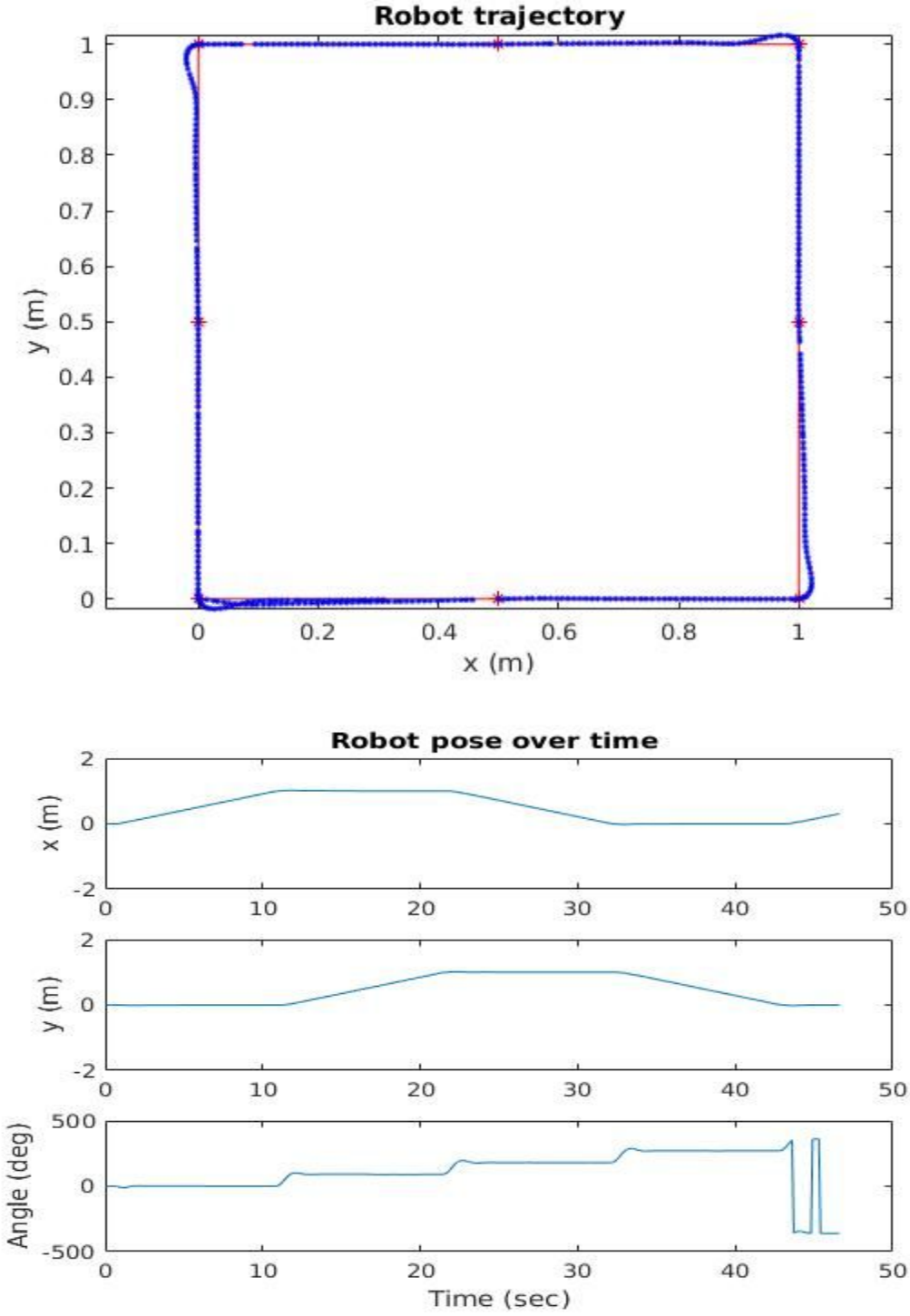


2) Kp = 2 V = abs(0.3 - 0.06 * abs(rotation_vel))





3) Kp = 2 V = abs(0.1 - 0.02 * abs(rotation_vel))



Discussions

- The above plots is showing that with the improvements in the linear velocity function, the error distance is much smaller at the corner of the square. After this improvement, the robot can even travel at a higher speed with the same P controller.
- At the lower speed, the robot travels even better compared to the previous ones with constant velocity.