Position Control

Karen Li - https://github.com/kkl224/MobileRobotics

***Abstract*— In this lab practice, I present the FreeNove Robot with an ultrasound to accomplish position control. I employ the technique of proportional control, or P-Control, a closed-loop control policy that uses feedback at every time step to correct the error. Moreoever, P-Control ensures the robot moves in the direction of the desired location, and the speed of the robot is proportional to the distance. In this lab, I demonstrate the use of proportional controller to drive the robot to a desired location using the actual robot and a simulator on Jupyter Notebook. In the end, I will also assess the performance of the robot and compare it with the theory.**

I. FreeNove Robot with Ultrasound

A. Time takes to reach X_d from different X_0

In my program, the default value of k is 100.

- 60 cm: 0.39 sec
- 70 cm: 0.43 sec
- 80 cm: 0.66 sec
- 90 cm: 0.72 sec
- 100 cm: 0.78 sec

B. Time takes to reach X_d with different value for k

In my program, I derive the speed or the mode of each DC motor u using the function

$$u = k * (distance - 50) \tag{1}$$

where *distance* is the current output from the Ultrasound. The output u is used here: PWM.setMotorModel(u, u, u, u).

- k = *100*: 0.78 sec
- k = *150*: 0.88 sec
- k = *200*: 1.29 sec

The result shows the robot works the best when k = 100, at a relative lower speed.

C. Robot Performance Analysis

Both sets of results show that the robot does not work as expected from theory, or rather partially work as expected. The reason is fairly simple and intuitive: since the theory assumes things are perfect, it does not take into account the external errors like friction force and robot hardware issues. Therefore, in real world practices, we are expected to see things not always following what theories suggest.

II. Simulation

This simulation tests the position control in 2 dimensions. In the simulation, the robot starts at location (4, 5) and moves to the desired location (2, 10). The time interval for this simulation is 10 secs for all 3 trials.

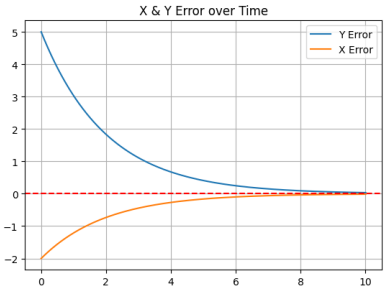

Fig. 1. k = *0.5*

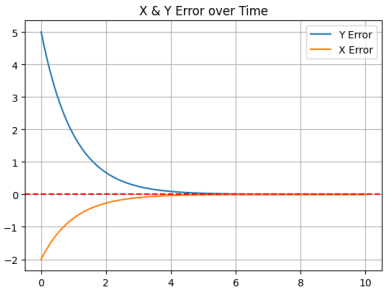

Fig. 2. k = *1*

III. Conclusion

The main difference between a real robot and the simulated robot is that a simulated robot will always work as expected if programmed properly, whereas a real robot, may not always work as we wished.

To improve the performance of a real robot, one way is to consider as many external source of errors as possible. That way, we can improve the accuracy of our result. Another way is to make the robot move slower as a slower movement will decrease the magnitude of the errors.

IV. Link to the recorded video

https://drive.google.com/drive/u/0/folders/10FM9CwagYL-kOuy7-5ChBe9Kex7KyDwf

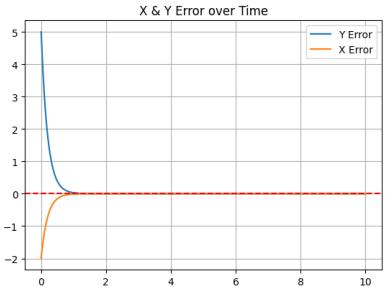

Fig. 3. k = *5*