University of California, Riverside

EE/ME144/EE283A
Foundations of Robotics

Fall 2021

Lab 1 Report

Oct 14, 2021

Name	SID	Section	Group Number
Shaheriar Malik	862154387	Thursday	

1. Problem Statement

In this lab, we learned how to write a Python script to control the robot. The task was to make the robot move in a square shape using open-loop control (i.e. sending commands only; no feedback). The waypoints to visit were [4, 0], [4, 4], [0, 4] and [0, 0]. This meant we needed to make the robot go straight, and then after it reaches the waypoint, turn 90 degrees. This was done 4 times for 4 waypoints.

2. Design Idea

My code contained two ‘for’ loops to give commands for linear and angular velocities, nested in a while loop that made sure to do those set of commands 4 times. The first for loop goes for 80 ticks and gives the linear velocity command. This is to have the robot go straight. After the 80 ticks are done, the code moves to the second ‘for’ loop which gives the command for the angular velocity. This is to have the robot move 90 degrees. This is done for 10 ticks since that seemed like a good time duration for it to complete the 90 degree turn.

3. Results

The lab was successful, and the robot visited all 4 waypoints in a square shape. We successfully learned how to connect Python with Gazebo and interact with the TurtleBot using a Python script. We also familiarized ourselves with the basic ROS concepts including ROS Node, ROS Topic and ROS Publisher.

4. Appendix (optional)

Scripts

Following is the run function we used to operate the robot.

```
    def run(self):

        vel = Twist()

        vel.linear.x = 0.5

        vel.angular.z = 0


        rotate = Twist()

        rotate.linear.x = 0

        rotate.angular.z = 0.5 * pi

        #rotate.angular.z = 1.58


        count = 0


        #while not rospy.is_shutdown():  # uncomment to use while loop

        while (count < 4):  #to have the linear and angular velocity commands run 4 times

                for i in range(80): #linear velocity for 80 ticks

                        self.vel_pub.publish(vel)

                        self.rate.sleep()

                rospy.loginfo('Linear movement complete')

                for i in range(10): #angular velocity to 10 ticks

                        self.vel_pub.publish(rotate)

                        self.rate.sleep()

                rospy.loginfo('Rotational movement complete\n')

                count += 1 #increment count

        rospy.loginfo('Visited all checkpoints. Closing.')


if __name__ == '__main__':

    try:

        whatever = Turtlebot()

    except rospy.ROSInterruptException:

        rospy.loginfo("Action terminated.")
```