University of California, Riverside

EE/ME144, EE283A
Foundations of Robotics

Fall 2022

Lab 2 Report

10/13, 2022

Name	SID	Section	Group Number
Tianzi Luo	862251424	021	06

1. Problem Statement

Lab 2: Open-loop Control

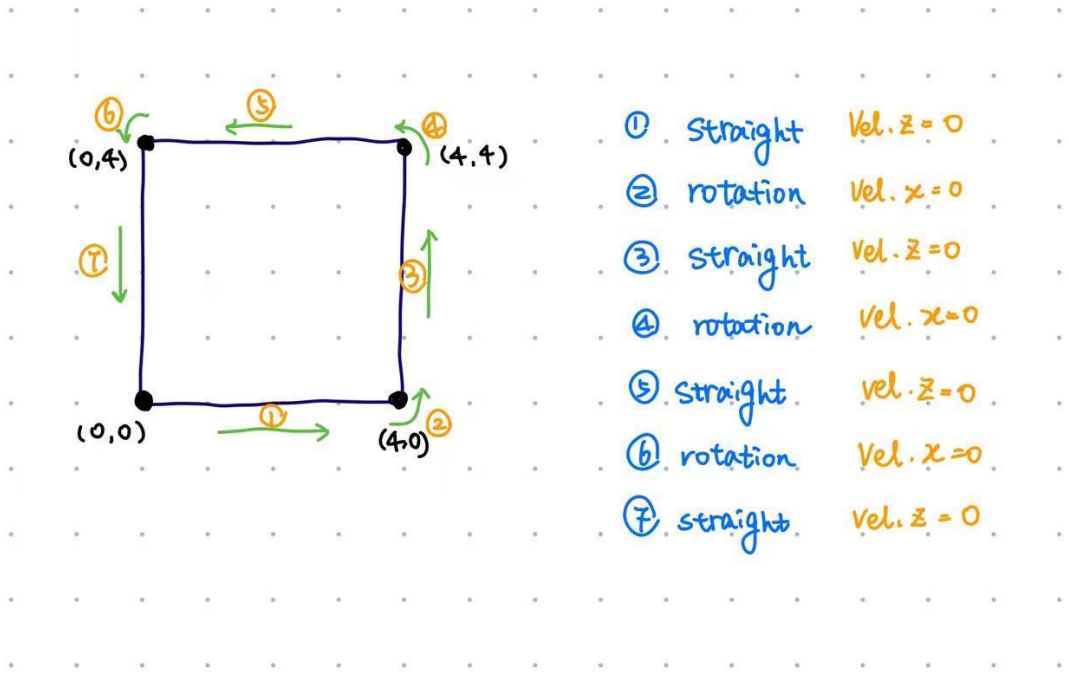
In an open-loop controller, the control action from the controller is independent of the "process output", which is the process variable that is being controlled. It does not use feedback to determine if its output has achieved the desired goal of the input command or process "set point".

Writing for loops in Python script to control the robot. Learning basic ROS concepts, including ROS Node, ROS Topic and ROS Publisher.

The task is using open-loop control to make the robot move in a square in Gazebo, visiting [4, 0], [4, 4], [0, 4] and [0, 0]. In other words, the robot should move forward 4 meters, turn left 90 degrees, move forward again 4 meters, and so on, until going back to the origin.The tolerance for distance error is 1.0m.

2. Design Idea

(a) Split the project into 7 small procedures. First go straight for 4 meters, then turn 90 degrees. Next, go straight again, until going back to the origin.

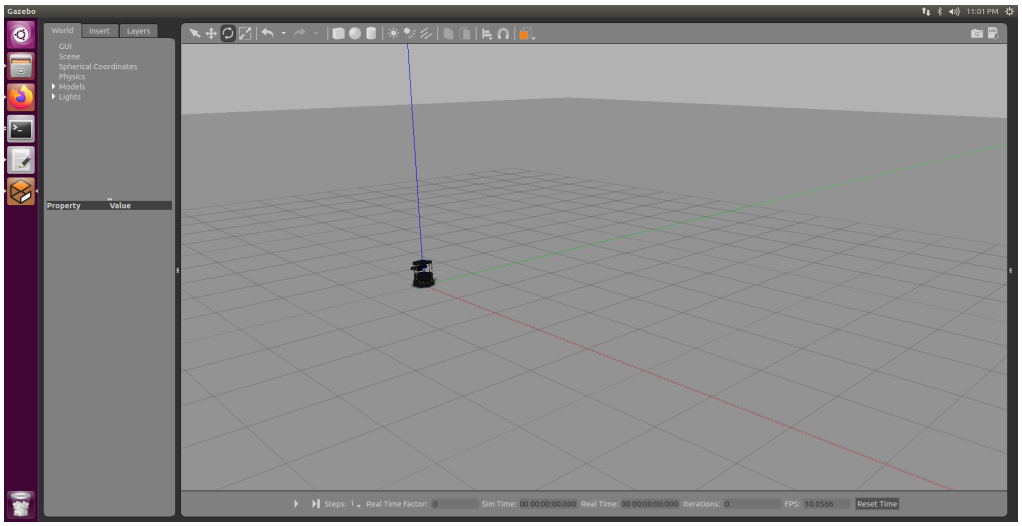


(b) Finish the first step, going straight for 4 meters. We need to write a for loop in python script and publish the linear velocity and angular velocity. In order to make the robot go straight, the angular velocity (vel.z) should be zero and assign a constant value in linear velocity (vel.x). Editing the code until the robot can reach the waypoint(4,0).

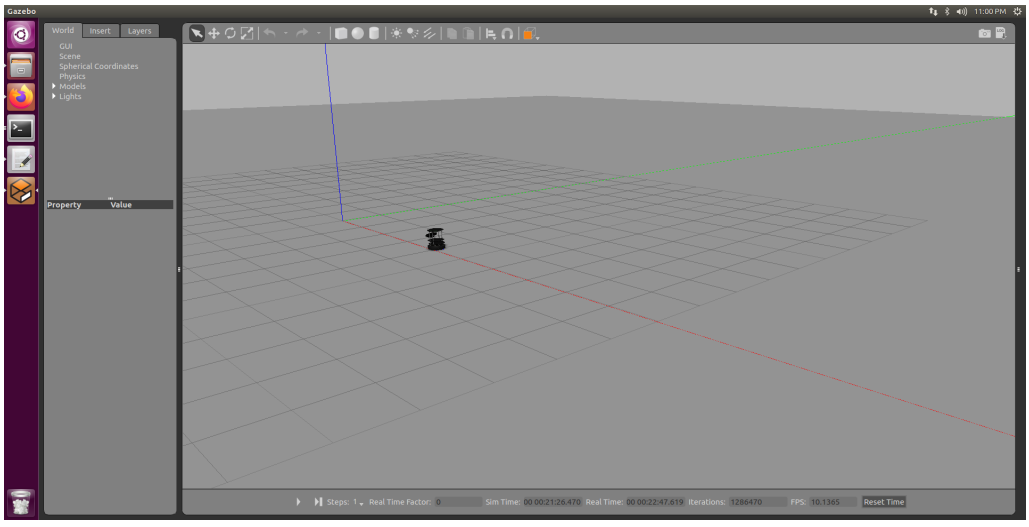
(c) Copy and edit the code for 7 times until the robot can go back to the origin. During the rotation, the linear velocity(vel.x) should be zero and the linear velocity(vel.x) should be a constant.

3. Results

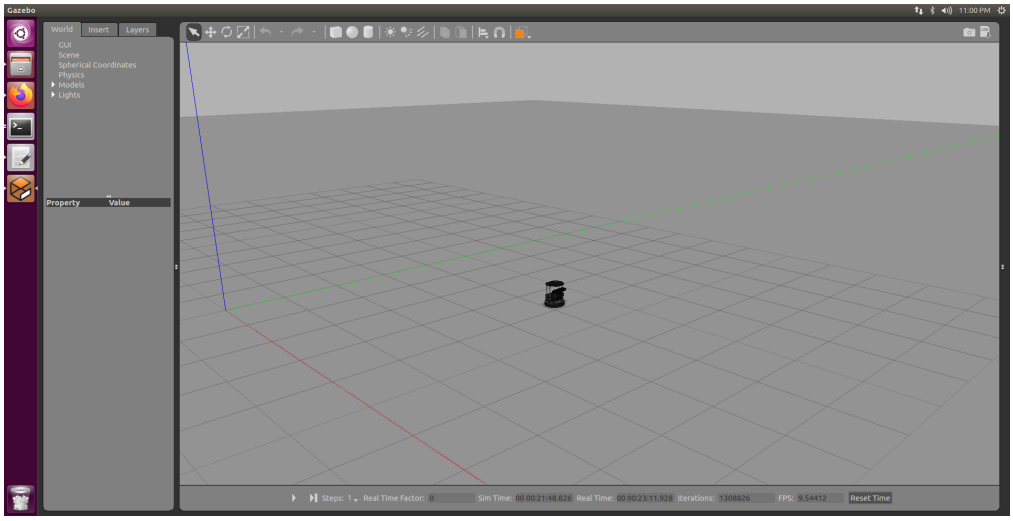
(a) Origin point (0,0): Go straight for 4 meters.



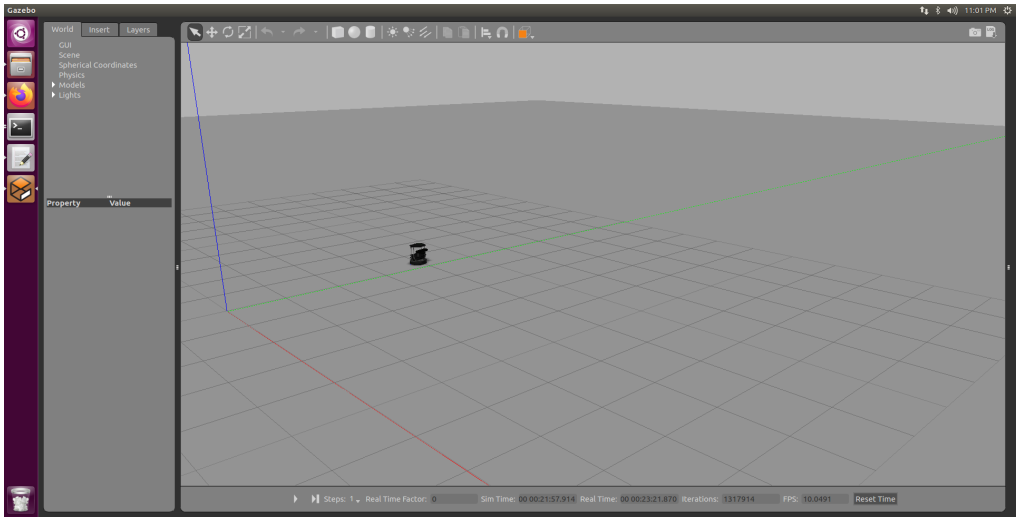
(b) First waypoint (4,0): Rotate 90 degree and go straight for 4 meters



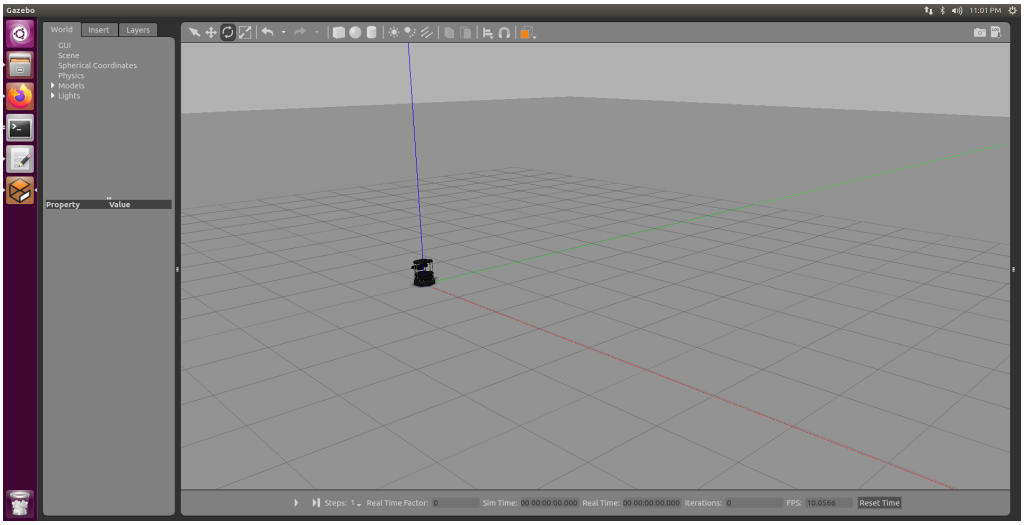
(c) Second waypoint (4,4): Rotate 90 degree and go straight for 4 meters



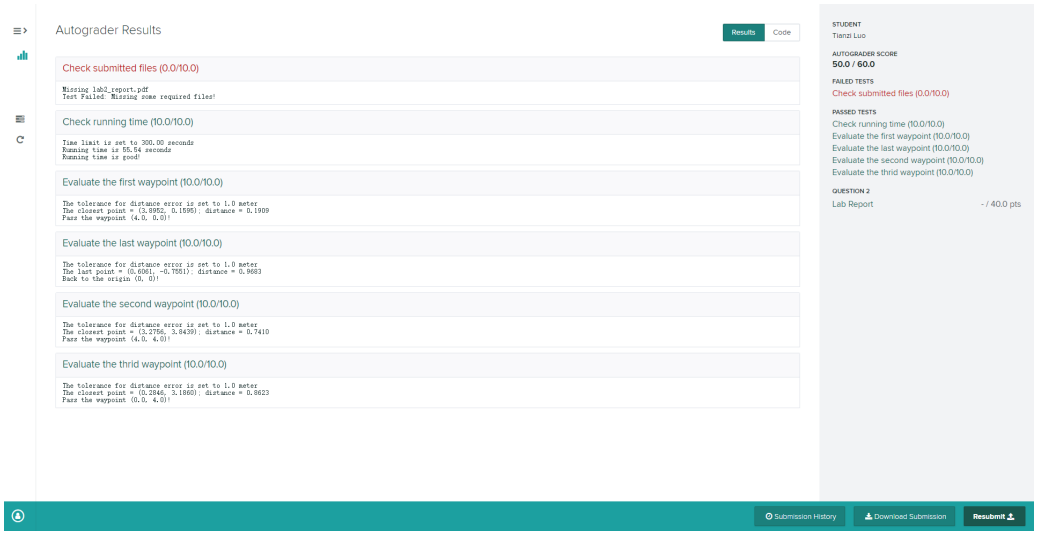
(d) Third waypoint (0,4): Rotate 90 degree and go straight for 4 meters



(e) Last waypoint (0,0):



Autograder result:



4. Appendix (optional)

1) How to run the code

open a terminal, launch the Gazebo:

```
roslaunch ee144f22 gazebo.launch
```

open a new terminal, run the open_loop script:

```
roscd ee144f22
cd scripts
python open_loop.py
```

2) References

None.

3) Scripts

Main script:

```
def run(self):
    vel = Twist()

    #1.Straight line
    vel.linear.x = 0.5
```

```
vel.angular.z = 0.01
#while not rospy.is_shutdown():  # uncomment to use while loop
for i in range(82):
    self.vel_pub.publish(vel)
    self.rate.sleep()

#2.Rotation
vel.linear.x = 0
vel.angular.z = 0.2
#while not rospy.is_shutdown():  # uncomment to use while loop
for i in range(80):
    self.vel_pub.publish(vel)
    self.rate.sleep()

    #3.Straight line
    vel.linear.x = 0.5
vel.angular.z = 0.01
#while not rospy.is_shutdown():  # uncomment to use while loop
for i in range(82):
    self.vel_pub.publish(vel)
    self.rate.sleep()

    #4.Rotation
vel.linear.x = 0
vel.angular.z = 0.2
#while not rospy.is_shutdown():  # uncomment to use while loop
for i in range(80):
    self.vel_pub.publish(vel)
```

```
    self.rate.sleep()



    #5.Straight line
    vel.linear.x = 0.5
vel.angular.z = 0.01
#while not rospy.is_shutdown():  # uncomment to use while loop
for i in range(82):
    self.vel_pub.publish(vel)
    self.rate.sleep()

    #6.Rotation
vel.linear.x = 0
vel.angular.z = 0.2
#while not rospy.is_shutdown():  # uncomment to use while loop
for i in range(70):
    self.vel_pub.publish(vel)
    self.rate.sleep()

    #7.Straight line
    vel.linear.x = 0.5
vel.angular.z = 0.01
#while not rospy.is_shutdown():  # uncomment to use while loop
for i in range(75):
    self.vel_pub.publish(vel)
    self.rate.sleep()
```