CSE360 Lab Report 1

William Han - https://github.com/willH177/CSE360MobileRobotics2024

I. Exercise 1

In exercise 1 it asks to for a point robot in 2-D, find a control policy u to make the robot move in an ellipse shape. The ellipse's major axis is equal to 4m, and the minor axis is equal to 2m. The major axis of the ellipse is aligned with the x-axis of the world frame. The center of the ellipse is in the point [3, 2].

The equations of the eclipse are:

$$x(t) = a\cos(t) \tag{1}$$

$$y(t) = b\sin(t) \tag{2}$$

That a is the major axis and b is the minor axis.

Since the center of the ellipse is at (3,2), and major axis is 4, and the minor axis is 2, then the equation should be:

$$x(t) = 4\cos(t) + 3 \tag{3}$$

$$y(t) = 2\sin(t) + 2 \tag{4}$$

Therefore we can calculate the initial position when the time is 0:

$$x(0) = 7 \tag{5}$$

$$y(0) = 2 \tag{6}$$

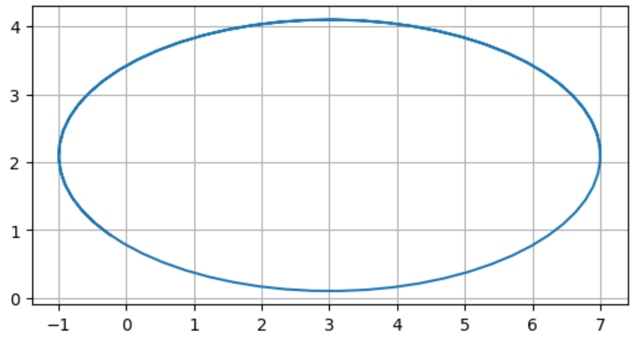
So the starting point is (7,2).

After that, we will take the derivative of x and y for the velocity that we will use in the simulation.

$$\dot{x}(t) = -a\sin(t) \tag{7}$$

$$\dot{y}(t) = b\cos(t) \tag{8}$$

Here's the final plot:



II. Exercise 3

In exercise 3 we are asked to write a control policy to follow the shape of a rotated eight curve.

The equations are:

$$x(t) = a\cos(t) \tag{9}$$

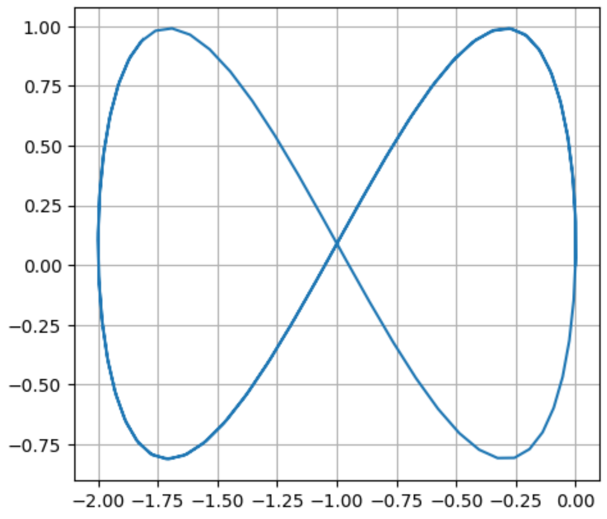
$$y(t) = b\sin(t)\cos(t) \tag{10}$$

Then we will take the derivative of x and y for the velocity that we will be using in the simulation.

$$\dot{x}(t) = a \cdot -\sin(t) \tag{11}$$

$$\dot{y}(t) = b\left(\cos^2(t) - \sin^2(t)\right) \tag{12}$$

Here's the final plot:



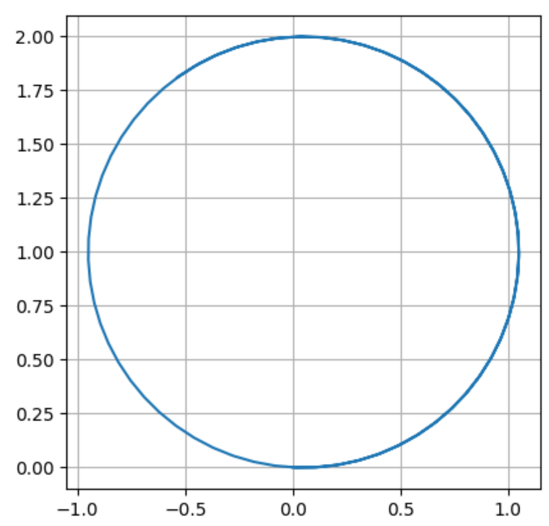
III. Exercise 4

In exercise 4 we are asked to simulate wind in the environment and run your control policy for circular motion.

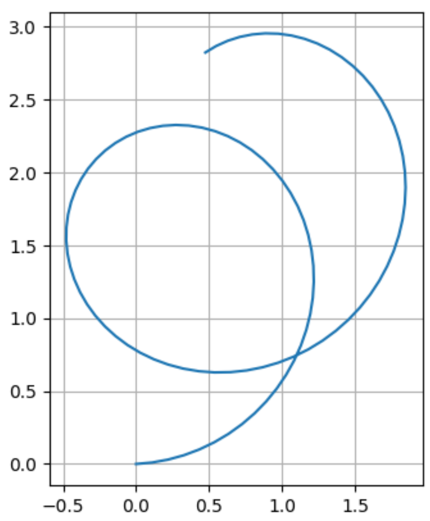
A. Constant Wind

The first case is add a constant wind w = [0.1, 0.1]. In order to simulate the wind, I added the wind constant in my integration function. The function calculates the total effect by adding the control input u and the constant wind vector windc. This addition simulates how both the control input and the wind together influence the movement of the robot.

The following is the plot without the wind:



Here's the plot with the 0.1 wind constant:

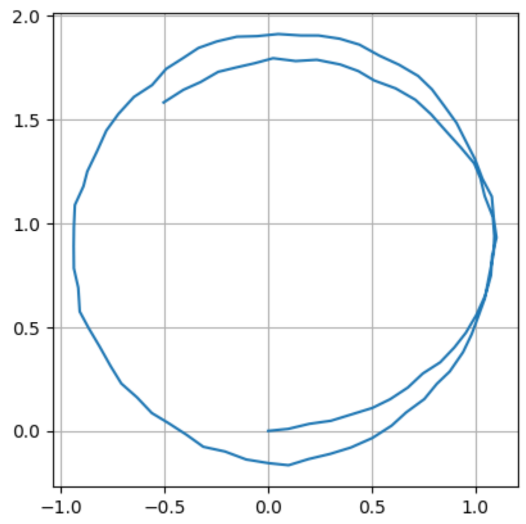


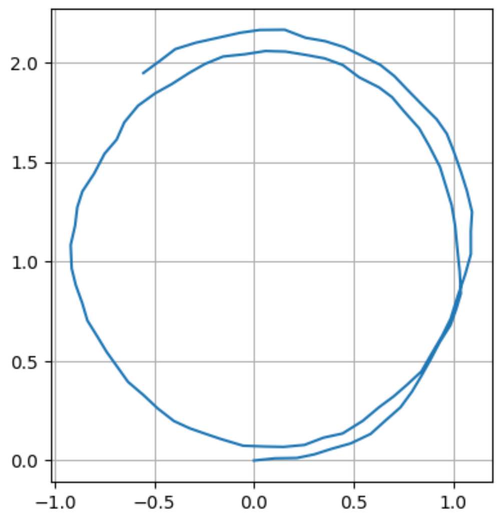
B. Random Wind

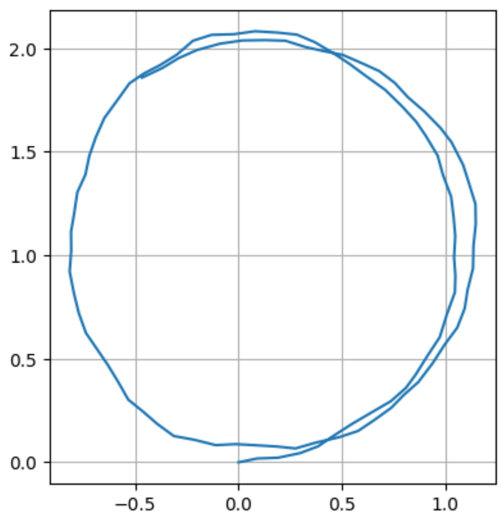
The second case is to add a random wind with a mean zero in x and y direction, and a standard deviation of 0.1. I replaced the wind constant in the first case with a random number with a standard deviation of 0.1.

I did this using the numpy.random.normal from the numpy library. Which draws random samples from a normal distribution.

The following are three plots with different wind constants generated randomly by the above function.


2.0
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0


2.0
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0


2.0
1.5
1.0
0.5
0.0
−0.5
0.0
0.5
1.0