MAE 145: Intro Robot Planning and Estimation
Homework #7

Assigned February 27. Due on March 5, Thursday by 10:30am

Notice:

Please upload 3 files for this homework. One to include the written part of the homework (problem 1), and two python files for the second exercise and third exercises. In your assignment include the names of the other students you have collaborated with to do the homework problems. The Collaboration Policy for this course is detailed in the syllabus.

Homework exercises:

1. (Differential drive.) A robot with a differential drive starts at position $x = 1.0m$, $y = 2.0m$ and with heading $\theta = \frac{\pi}{2}$ (the direction of the y axis is $\frac{\pi}{2}$). It has to move to the position $x = 1.5m$, $y = 2.0m$ and heading $\theta = \frac{\pi}{2}$. The movement of the vehicle is described by steering commands c each consisting of three terms, as in $c = (v_l =$ angular speed of left wheel, $v_r =$ angular speed of right wheel, $t =$ driving time), where the speeds are constant values that are applied during t seconds.

   (a) What is the minimal number of steering commands (v_l, v_r, t) needed to guide the vehicle to the desired target location?

   (b) What is the length of the shortest trajectory under this constraint?

   (c) Which sequence of steering commands guides the robot on the shortest trajectory to the desired location if an arbitrary number of steering commands can be used?

   (d) What is the length of this trajectory? (Note: the length of a trajectory refers to the traveled distance along the trajectory.)

2. A working probabilistic motion model is a requirement for all Bayes implementations. In the following, you will implement the simple odometry-based motion model and generate samples according to it.

   (a) Implement the odometry-based motion model in `Python`. Your function should take the following three arguments:

$$\mathbf{x}_t = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \qquad \mathbf{u}_{t+1} = \begin{bmatrix} \delta_{1,\text{rot}} \\ \delta_{2,\text{rot}} \\ \delta_{\text{trans}} \end{bmatrix}, \qquad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix},$$

   where $\mathbf{x}_t$ is the current pose of the robot, $\mathbf{u}_{t+1}$ is the odometry reading obtained from the robot, and α are the noise parameters of the motion model. The return value of the function should be the new pose $\mathbf{x}_{t+1}$ of the robot. As we do not expect the odometry measurements to be perfect, you will have to take measurement errors into account when implementing your function. For this purpose, use the method to generate samples from a normal distribution that you obtained in a previous exercise.

Programming guidelines:

The name of your python file should be **python_program_p2.py** with the first line in the file having your PID as a comment, Ex: # A123456.

The example shown below shows the data structures to be used in python. The values are merely representative.

$\mathbf{x}\_t = [x, y, \theta]$

$\mathbf{u}\_t\text{\_plus\_1} = [\delta_{1,\text{rot}}, \delta_{2,\text{rot}}, \delta_{\text{trans}}]$

$\mathbf{alpha} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]$

$\mathbf{x}\_t\text{\_plus\_1} = \texttt{predict}(\mathbf{x}\_t, \mathbf{u}\_t\text{\_plus\_1}, \mathbf{alpha})$

$\mathbf{x}\_t\text{\_plus\_1} = [x, y, \theta]$

All angles are defined in radians.

(b) If you evaluate your motion model over and over again with the same starting position, odometry reading, and noise values, what is the result that you would expect?

(c) Evaluate your motion model 5000 times for the following values:

$$\mathbf{x}_t = \begin{bmatrix} 2 \\ 4 \\ 0 \end{bmatrix}, \mathbf{u}_t = \begin{bmatrix} \frac{\pi}{2} \\ 0 \\ 1 \end{bmatrix}, \alpha = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.01 \\ 0.01 \end{bmatrix},$$

Plot the resulting positions of the 5000 evaluations in a single plot.

Note: You need not indicate the orientation of the robot.

3. (Differential drive implementation.) Write a function in `Python` that implements the forward kinematics for the differential drive as explained in the lecture notes. The input parameters of the function should be:

   (a) the pose of the robot x, y, and θ,

   (b) the speed of the left and right wheels, v_l and v_r,

   (c) the driving time t

   (d) and the distance between the wheels of the robot l and the wheel radius R.

   (e) The output of the function should be the new pose of the robot x, y, and θ

Programming guidelines: The name of your python file should be **python_program_p3.py** with the first line in the file having your PID as a comment, Ex: # A123456.

The example shown below shows the data structures to be used in python. The values are merely representative.

x, y is the position and theta is the orientation of the robot.

vel_l and vel_r are the speeds of left and right wheel respectively.

t is driving time

l is the distance between the wheels and R is the wheel radius

x_1, y_1 , theta_1 = `predict_next_pos`(x, y, theta, vel_l, vel_r, t, l, R)

All angles are defined in radians