CMPE185 Autonomous Mobile Robots

Fall 2024 Homework 1

Problem 1. (20pts) Suppose a two-wheel differential drive mobile robot equipped with a 2D range sensor starts at position $x = 1.0$m, $y = 2.0$m, with heading $\theta = \pi/4$. A range sensor is attached to the center of the robot. The range sensor detects an obstacle and returns a reading of $\alpha = -\pi/6$ and $d = 1.0$m.

1) What is the position of the obstacle in the global coordinate frame?
2) For the same robot, suppose the wheel radius is 0.3m and the length of the axles is 1.6m. For the wheel encoder, the total ticks per revolution is 50. After a while, 20 ticks were recorded for the left wheel, and 40 ticks were recorded for the right wheel, will the car collide with the obstacle? Write down all your work.

Problem 2. (40pts) A differential drive robot has two wheels, each of radius r, separated by a distance L (the wheelbase). The robot moves by controlling the angular velocities of the left and right wheels (ω_l and ω_r). Your task is to compute the robot's position (x, y, θ) over time using the kinematic equations given on Page 20 in Lecture Note 4.

1) In the given code skeleton "Fall24_CMPE185_HW1.ipynb", finish the Python function "update_position(x, y, theta, omega_l, omega_r, r, L, dt)" that takes the current state of the robot, wheel velocities, and time step 'dt', and returns the new position (x_new, y_new, theta_new).

2) Simulate the robot's motion over time. Assume a time step Δt of 0.1 seconds and simulate for 100 seconds. Test the function for the following cases:
   (a) The robot moves in a straight line (ω_r = ω_l).
   (b) The robot makes a right turn (ω_r > ω_l).
   (c) The robot makes a left turn (ω_l > ω_r).
   Plot the robot's trajectory over time.

Problem 3. (40pts) In a differential drive robot, the odometry can be computed using the number of encoder ticks on each wheel. Given the number of ticks (ticks_l, ticks_r) and the number of ticks per revolution (N), we can calculate the distance traveled by each wheel and use that to update the robot's position. Your task is to compute the robot's position (x, y, θ) over time using the odometry equations given on Page 21 in Lecture Note 5.

1) In the same skeleton code, finish the Python function "update_position_with_ticks (x, y, theta, ticks_l, ticks_r, r, L, N)" and return the new position (x_new, y_new, theta_new)
2) Simulate the robot's motion over time. Assume a time step Δt of 0.1 seconds and simulate for 100 seconds. Test the function for the following cases:
   (a) The robot moves in a straight line (ticks_l = ticks_r).
   (b) The robot makes a right turn (ticks_l < ticks_r).
   (c) The robot makes a left turn (ticks_l > ticks_r).
   Plot the robot's trajectory over time.

Deliverables for Problem 2 and Problem 3:

1. Submit your Python code.
2. A plot of the robot's trajectory for each test case.
3. A brief explanation of the robot's behavior in each case (straight, right turn, left turn).