CPE 470/670 Fall 2025 Homework 2

Instructor: Parikshit Maini Submission Deadline: 11:59 pm Tuesday, October 21, 2025

All text answers must be written in latex and submitted as a pdf document. Handwritten or scanned submissions will not be accepted and will not be graded. All code must be written in Python. Zip together all submission material and submit the Zip folder. Question 1 is required for everyone. Question 2 is optional for CPE 470 students and is **required** for CPE 670 students. CPE 470 students can attempt question 2 for bonus points.

Your submission to this exam must be your own work. You are not permitted to collaborate with other students on the exam, seek help from other people or AI based tools. You are permitted to refer to class notes, lecture materials, books and other online teaching resources.

You are given the data output from two LiDAR sensors in files "lidar1.txt" and "lidar2.txt". The LiDARs were placed at the point P shown in Figure 1. The lidar data is in polar coordinate format i.e. (angle,range) where *angle* is the angular position of the datapoint in the LiDAR's frame of reference and *range* is the distance of the datapoint from the LiDAR center. Angluar values are given in degrees and range is given in cm. Both LiDARs use an orthogonal frame of reference with origin at the center of the LiDAR located at P.

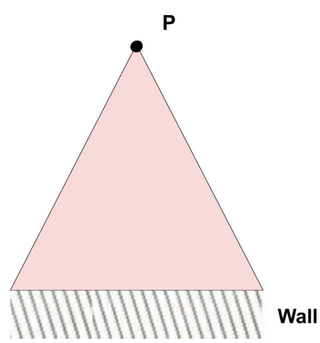

Figure 1: The LiDARs were placed at point P in front of a wall to collect data.

LiDAR 1 has a *range* measurement variance of 1 cm^2 and LiDAR 2 has a *range* measurement variance of 4 cm^2. Each measurement is independent and identically distributed.

Question 1:

1. (20 points) The LiDAR data is generated in polar coordinate form. Write a python script to convert the data to Cartesian form (i.e. from (angle,range) to (x,y))

2. You will use the LiDAR data to find the equation of line that represents the wall. To find the equation you will use weighted least squares algorithm taught in class. The equation of a line is $y = mx + c$, where m is the slope and c is the y-intercept. You will use the Cartesian coordinates for the datapoints (derived in the previous question) to estimate the equation of the line. Assume that the value of the x-coordinate of the datapoints is known perfectly and sensor range measurement noise only exists in the y-coordinate. The two unknowns to be estimated to find the equation of the line are m and c. Hence the two parameters being estimated are $\begin{bmatrix} m \\ c \end{bmatrix}$.

   (a) (30 points) The measurement model to be used is $y = \mathbf{H}\mathbf{X} + v$, where $\mathbf{X} = \begin{bmatrix} m \\ c \end{bmatrix}$, v is the LiDAR measurement noise and y is the y-coordinate of the point. Write the Jacobian matrix $\mathbf{H}$ for the given measurement equation. Write your answer in text.

   (b) (30 points) Explain in a text writeup how you will use the weighted least squares algorithm to estimate the equation of the line using the Jacobian matrix and the Cartesian form of the points. Show the instantiation of each equation and computation of the estimates of m and c. Show step-by-step calculations. For the text writeup show your calculations using 4 datapoints (use first two datapoints from each LiDAR).

   (c) (70 points) Write a python script to implement the weighted least squares (WLS) to estimate the equation of the line. You are not permitted to use Python library implementations of WLS and must implement it yourself. Use all 40 points (20 from each LiDAR) to compute the estimates of m and c using the python script.

Question 2: File "streamdata.txt" contains 4 datapoints from the two LiDARs. Each datapoint gives LiDAR id (1 or 2), angular value and range. Angular values are in degrees and range values are in cm. Use Recursive Least Squares (RLS) Algorithm to use one datapoint at a time to update your estimate of the line equation and estimator covariance. Use the estimates from Question 1 as the initial estimate of the parameters needed for RLS. Use standard deviation value of tan(5°) and 1 cm for m and c, respectively, to initialize estimator covariance. Convert the datapoints to Cartesian coordinates. Use the measurement model and sensor range measurement variance values from Question 1.

1. (40 points) Explain in a text writeup how you will use the recursive least squares algorithm to estimate the equation of the line using individual datapoints. Show the instantiation of each equation and computation using first two datapoints given in streamdata.txt.

2. (40 points) Write a python script to implement the recursive least squares to improve the estimate of the equation of the line using one datapoint at a time. You are not permitted to use Python library implementations of RLS and must implement it yourself. Print the value of $\begin{bmatrix} m \\ c \end{bmatrix}$ and estimator covariance after each of the four datapoints given in streamdata.txt.

————————xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx————————-