MAE 4180/5180, ECE 4772/5772, CS 3758

Autonomous Mobile Robots

Homework 3 - due **Wed. Mar 17** by 11:59PM

A document containing the answers and figures should be uploaded to Gradescope as `HW3writeup.pdf`. All the code and the files needed to run the code are to be uploaded to Canvas as a zip file named `HW3code.zip`. Note, the code zip file should contain ONLY .m files and files containing data (if needed) and **not any other file** (writeup, assignment, etc). Specific functions, as indicated below, should be uploaded to their corresponding assignments on **Canvas**.

Notes about autograded assignments:

- We **highly** recommend you develop and debug the autograded functions in Matlab. The error messages that the autograder provides are not as explicit as in Matlab
- You may submit as many times as you like until the deadline
- Make sure to also include the functions in the zip file you upload to Canvas
- **Reusing code:** We encourage you to reuse your prior functions in the new functions. For the autograded assignments, if you want to use a previous function you have written (for example robot2global), simply copy-paste it below the function you are writing

Localizing a stationary robot (80 Points)

A point robot is standing in place in the map described in `HW3map.mat` which contains the coordinates of all walls in the environment: $[x_1, y_1, x_2, y_2]$. The file `stationary.mat` contains the sensor measurements for the robot. Each line contains the range measurements in the global North, East, South and West directions. The maximum range of the sensor is less than 3 meters (a measurement of NaN indicates no obstacle within the sensing range).

Assume that there is no uncertainty in the angle and that the North and South ranges have noise that is distributed N(0,0.1) and the East and West ranges have noise that is distributed N(0,0.3). All range measurements are independent of each other.

Grid localization (30 points)

1. Write the function `gridLocalizationStationary.m` that returns the pdf representing the location of the robot. This function should accept as input the size of the grid $(n \times m)$ where n is the number of cells along the X dimension and m the number fo cells along the Y dimension. Explain how you addressed the measurement issue discussed in class (with respect to where is the measurement?).
2. Assume a 10x10 grid. What is your initial distribution?
3. Plot the pdf - the initial one and the final one (after incorporating all measurements). You may find the function `plotGridBelief.m` useful.
4. Calculate and plot a pdf for a grid size of 40x22. Plot the final pdf. Do you get the same possible robot location(s) as in part 3? explain.

Kalman filter (30 points)

1. Write the function `KFStationary.m` that estimates the position of the robot using a Kalman Filter.
2. What is your prediction step. Do you need one?
3. How is the update performed? (write the equation)

4. What are you choosing as your initial distribution? why those values?

5. Plot the initial and final position and covariance estimates (1 σ) on the map. (Hint: for plotting the ellipse, do not forget the functions posted at the beginning of the semester on Canvas)

6. Try a different initial distribution. How does that affect the position estimate?

7. Would an EKF be more appropriate for this problem? explain.

Test function (autograded) (10 points)

1. Edit the test function on Canvas `TestFunHW3.m` look for comments that begin with 'STUDENTS' and follow the directions. Course staff will use this function to test your code with different inputs and we are using the autograder as a way to ensure the test function runs. We are NOT testing for correctness of the algorithm.

   Make sure the zip file you submit on Canvas contains this function and all of the functions necessary to run it.

Location estimate (10 points)

1. For both filters, would you be able to estimate the location exactly given perfect range information? Explain. (5 points)

2. You are told that the robot's y coordinate is less than 8. Based on the various pdfs found in the previous sections, can you estimate the true position of the robot? Explain (no need to run the filters) (5 points)