

SE 423 Mechatronics Homework Assignment #6

Due On Gradescope, by 9AM Friday, May 08.

Exercise 1: Path Planning

The goal of this problem is to have you read through the given code and implement the A* path-planning method. Most of the code has been given to you. Your job is either to fill in the blank with some code or add some comments indicating that you understand the algorithm. I suggest you perform this homework exercise in the lab. The lab PCs have the correct version of Microsoft Visual Studio 2022 installed. All the code and project files you need for this problem are posted at the course website <http://coecsl.ece.illinois.edu/se423/hw/astarhw6.zip>. Download this zip file and unzip it into a folder.

Perform the following:

1. In this project's directory, you will find the executable file "astarDemo.exe." This is given to you to see how the program should run when you fill in the required code. Run "astarDemo.exe" and play around with the different maps that are created for you to test the algorithm. Use `astarDemo.exe` in the below items to double-check that your code is running correctly.
2. Load the Visual Studio 2022 project by double clicking on the `astar.sln` file. Suppose Windows asks to select Visual Studio 2022. This will load Visual Studio 2022. It may take longer to load the project the first time it is opened. Under "source files," find and open `astar.cpp`. Take some time to run through the code in this file. There are many helpful comments in the file that will help you with this exercise. Also, the pseudocode for the A* algorithm we went through in class and posted here could be helpful for this exercise. 12 items are numbered in the A* function of `astar.cpp`. Some of the items have you add a line or two of source code. Other items ask you to write comments explaining what certain lines of code accomplish. Complete these 12 items. Then compile (use menu item "Rebuild Solution") and run your project by pressing the arrow/play button. Once your program is compiled, it should run identically to `astarDemo.exe`. For your homework submission, print out the changes and comments you made for these twelve items.
3. Change the location of at least three obstacles in the course map defined by the array "`char mapCourseStart[176]`". This is also the map used when item 7 is selected in the application. For your homework submission, print the output of your compiled program when you have selected a start and finish point with several obstacles between them.
4. As a final exercise, add enough obstacles so that there is no path A* can find from the starting point to the ending point. Verify that A* returns an error when it cannot find a path.

Exercise 2: Kalman filtering

The goal of this problem is to give you experience with the Kalman filter and to help you understand how it is helping with the robot's navigation. Given a set of actual data collected during a run of the robot car, your goal is to find a Q and R matrix that makes the Kalman filter fuse the dead-reckoning data from the wheel encoders and rate gyro with the OptiTrack data. The "ideal" merge of this data is not to track precisely the OptiTrack data. It should definitely be close to the OptiTrack data, but filter it so that significant errors are not precisely followed. The OptiTrack's resolution is $\pm 2\text{cm}$, and it is possible (but not very often) for the OptiTrack to send

incorrect measurements. I purposely added a 1-tile error to one of the OptiTrack data points so you can see how your Kalman filter reacts to such a significant error.

What have you given:

1. The data file `data_forKalman_Filter_HWProblem.mat` found at <http://coecsl.ece.illinois.edu/se423/hw/HW6.htm>. It has 6 columns: index, average wheel velocity, gyro reading in rad/s, x OptiTrack position, y OptiTrack position, theta OptiTrack. After you load the file, you will see a lot of zeros in the OptiTrack columns. The OptiTrack only samples at 100 Hz, while the dead-reckoning data (wheel velocity, gyro) is updated at 1000 Hz. Your code will not use the OptiTrack data when it is 0.0. Load the “data” variable by loading the `data_forKalman_Filter_HWProblem.mat` file. This can be done using the “load” command or double-clicking the file in the “current directory” window of the MATLAB window.
2. A Matlab script file that gives you most of what you need for the code for this problem, `KalmanFilterHWProblem.m` located <http://coecsl.ece.illinois.edu/se423/hw/HW6.htm>. You will need to make two modifications to this file: (1) adding the Kalman equations to the large for loop, and (2) assigning values to the Q and R matrices used in the filter.

In the m-file’s for loop, there are comments to guide you through adding the code. Within the for loop, you need to implement the prediction and correction steps. Referring to the given equations at the end of this homework might be helpful during this coding.

After adding the Kalman filter code to the for loop, you need to choose values for the Q and R matrices. The following form works well for the robot car.

$$U_P = \text{Process (Dead Reckoning) Uncertainty}$$

$$Q = \begin{bmatrix} U_P & 0 & \frac{U_P}{10} \\ 0 & U_P & \frac{U_P}{10} \\ \frac{U_P}{10} & \frac{U_P}{10} & 0 \end{bmatrix}$$

$$U_M = \text{Measurement (OptiTrack) Uncertainty}$$

$$R = \begin{bmatrix} U_M & 0 & \frac{U_M}{10} \\ 0 & U_M & \frac{U_M}{10} \\ \frac{U_M}{10} & \frac{U_P}{10} & U_M \end{bmatrix}$$

Start with U_P and U_M 1.0 and run the M-file. (U_P and U_M are named “MeasUncert” and “ProcUncert” in the code.) Zoom in on the third plot that plots both data sets on top of each other. Does it look like the Kalman filtered data is following the OptiTrack data too closely and jumping around? OptiTrack data is the dots. Also, make sure to see what happens at the one error point that is off by one tile in the y direction. Then tune U_P and U_M values by changing them by a positive or negative power of 10 relative to each other. Come up with 3 Q and R matrices pairs that show three cases:

- (a) Kalman output data follows the OptiTrack data too closely.
- (b) Kalman output is filtered too much and converges slowly to the OptiTrack data.
- (c) Kalman output is just right, converges pretty quickly, but does not jump immediately to every OptiTrack data point.

Make sure to produce a plot for each of these cases showing how the Kalman filter is working. In your plots, zoom in on the behavior of your fused data near the discontinuity in the OptiTrack data. Be sure to show how quickly the fused data converges to the OptiTrack data.

What to turn in:

- A printout of your edited M-file
- Three plots for the three Q and R pairs that you chose, and somewhere in the plot, the value of the Q and R matrices.

0.1 Kalman Equations

For the robot car:

$$F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} \cos(\theta_k)\Delta t & 0 \\ \sin(\theta_k)\Delta t & 0 \\ 0 & \Delta t \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ and start } P \text{ as identity } P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

K is 3×3 and S is 3×3

$$\begin{aligned} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix} &= \begin{bmatrix} x \text{ position of robot} \\ y \text{ position of robot} \\ \theta \text{ of robot} \end{bmatrix}, \\ \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} &= \begin{bmatrix} \text{average velocity of wheels} \\ \text{rate gyro measurement} \end{bmatrix}, \\ \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} &= \begin{bmatrix} \text{OptiTrack x position of robot} \\ \text{OptiTrack y position of robot} \\ \text{OptiTrack } \theta \text{ of robot} \end{bmatrix} \end{aligned}$$

Q and R are 3×3 matrices that you need to tune

Prediction Steps

$$\begin{aligned} \hat{x}_{k+1|k} &= F\hat{x}_{k|k} + Bu_k \\ P_{k+1|k} &= FP_{k|k}F^\top + Q_k \end{aligned}$$

Correction Steps

$$\begin{aligned} \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K\tilde{y} \\ P_{k+1|k+1} &= (I - KH)P_{k+1|k} \end{aligned}$$

where

$$\begin{aligned} \tilde{y} &= z_{k+1} - H\hat{x}_{k+1|k} \\ S &= HP_{k+1|k}H^T + R_{k+1} \\ K &= P_{k+1|k}H^T(S)^{-1} \end{aligned}$$