EECS4421: Lab 6

Thu Mar 29, 2018
Due: before 11:59PM Fri Apr 6, 2018

Implement the components of a Kalman filter for solving the following SLAM problem:

Suppose that you have an omnidirectional (slide 32 from the Kalman filter examples lecture) robot moving in an environment with two point landmarks m_1 and m_2 whose locations are unknown. At each time step, the robot can move in any direction by an arbitrary distance; this is the control input u_t. Assume that the control input is accurate (has zero mean error) but has covariance R proportional to the squared distance moved:

$$R = \begin{bmatrix} 0.25\|u_t\|^2 & 0 \\ 0 & 0.25\|u_t\|^2 \end{bmatrix}$$

where $\|u_t\|^2$ is the squared magnitude of the control vector.

The plant model for the robot is:

$$x_t = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \\ m_{1,x} \\ m_{1,y} \\ m_{2,x} \\ m_{2,y} \end{bmatrix}_{t-1}}_{x_{t-1}} + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}_t}_{u_t} + \varepsilon$$

where $\begin{bmatrix} x \\ y \end{bmatrix}_{t-1}$ is the estimated location of the robot at time $t-1$, $\begin{bmatrix} m_{1,x} \\ m_{1,y} \end{bmatrix}_{t-1}$ is the estimated location of landmark 1 at time $t-1$, $\begin{bmatrix} m_{2,x} \\ m_{2,y} \end{bmatrix}_{t-1}$ is the estimated location of landmark 2 robot at time $t-1$, and ε is a zero mean Gaussian random variable whose first two components have covariance R and the remaining four components are all zero (because the landmarks do not move).

Assume that the robot is equipped with a sensor that can measure the vectors from the robot's location to multiple landmarks simultaneously and unambiguously (i.e., the robot can measure the vector from its location to both landmarks and it knows which landmark produces which measurement) as long as the distance to the landmark is less than or equal to 5; if the distance to a landmark is greater than 5 then the sensor returns the vector $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Assume that the measurement to a landmark accurate (has zero mean error) but has covariance proportional to the squared

distance to the landmark. In other words, if both landmarks are measurable then the measurement model is:

$$z_t = \underbrace{\begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 1 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} x \\ y \\ m_{1,x} \\ m_{1,y} \\ m_{2,x} \\ m_{2,y} \end{bmatrix}_t}_{x_t} + \delta$$

where δ is the zero mean Gaussian measurement noise with covariance

$$Q = \begin{bmatrix} 0.01\|d_1\|^2 & 0 & 0 & 0 \\ 0 & 0.01\|d_1\|^2 & 0 & 0 \\ 0 & 0 & 0.01\|d_2\|^2 & 0 \\ 0 & 0 & 0 & 0.01\|d_2\|^2 \end{bmatrix}$$

where $\|d_1\|^2$ is the squared distance between the robot and landmark 1 and $\|d_2\|^2$ is the squared distance between the robot and landmark 2.

If landmark 1 is out of range then the matrix C becomes:

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

If landmark 2 is out of range then the matrix C becomes:

$$C = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

If both landmarks are out of range then $C = 0$.

On the course web site you will find `kf.m`, a Matlab implementation of a Kalman filter that requires the user to supply 4 functions desribed below:

(a) Implement the function:

```
function [A, B] = slam_plant_model()
```

that returns the matrices A and B for the plant model.

(b) Implement the function:

```
function R = slam_plant_cov(ut)
```

that returns the plant covariance matrix R. Note that you need to examine the elements of u_t to compute the magnitude of the control vector.

(c) Implement the function:

```
function [C, z] = slam_meas_model(xt)
```

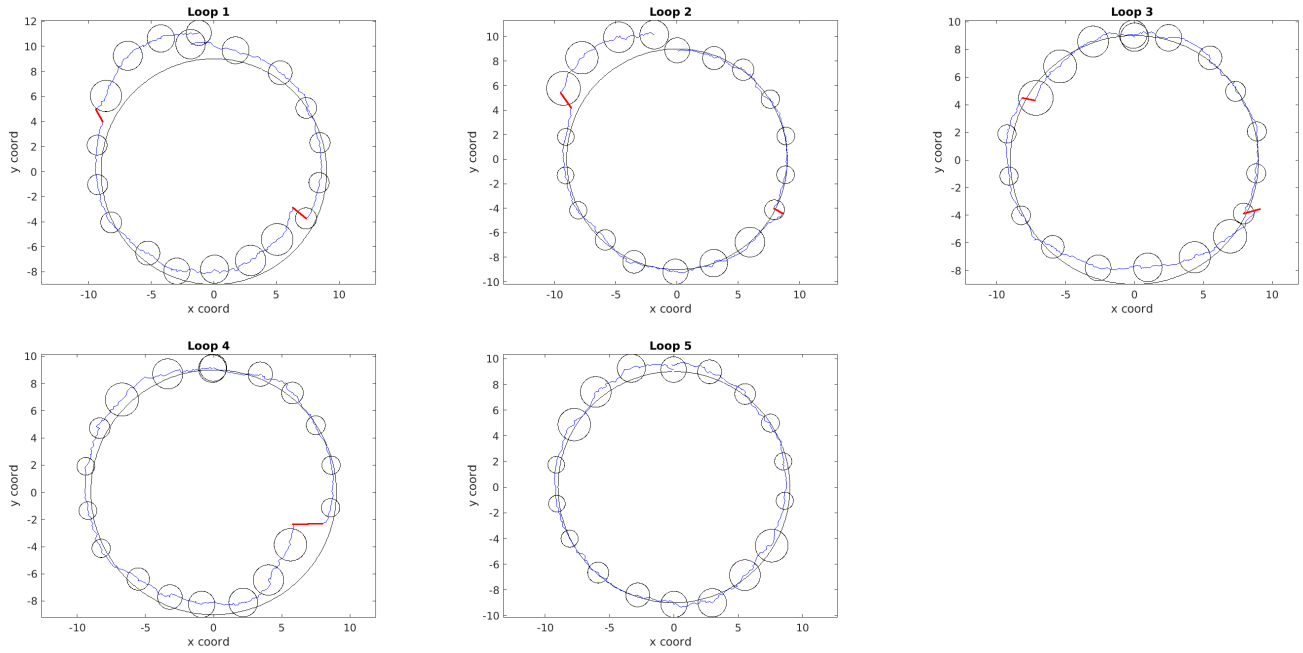
that returns the measurement model matrix C and a noise-free measurement given a state vector x_t. Note that you will need to examine the elements of x_t to determine whether or not the landmarks are measurable.

(d) Implement the function:

```
function Q = slam_meas_cov(xt)
```

that returns the measurement covariance matrix Q. Note that you will need to examine the elements of x_t to compute the distances between the robot and the landmarks.

(e) On the course web site you will find `kf_slam.m`, a Matlab script that simulates a robot driving on a circular path between two landmarks. The robot completes several loops of the path, and the results are plotted for each loop of the path. On each plot, the large black circle is the true path of the robot, the blue line is the estimated path of the robot, and smaller black circles are graphical representations of the estimated covariance of the robot's position (centered at the estimated robot's position). See the figure below for an example of one run of the script.



Run the script to check if your implementation of the four functions appears to be correct. Note that the randomness of the control inputs will cause your results to differ from the figure above, but you should get a roughly similar set of plots.

(f) Occassionally, large jumps in the estimated robot position occur (shown in red in the figure above). Explain why these jumps occur.

Submit your written explanation for question (f); you may submit this electronically with your Matlab files if you wish. Submit Matlab scripts (your implementation of the 4 functions).

```
submit 4421 lab6 your-matlab-files
```