EECS4421: Lab 5

Thu Mar 22, 2018
Due: End of class on Fri Mar 30, 2018

1. Implement an extended Kalman filter (EKF) where the plant model and the measurement model are non-linear. Use your EKF to solve the following localization 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 and a known goal location g. The coordinates of the landmarks are $m_1 = [5\ 0]^T$ and $m_2 = [10\ 0]^T$. The coordinates of the goal are $g = [15\ \ -2]^T$. At each time step, the robot can move one unit towards the goal (in a straight line from its estimated current position towards the goal); this is the control input u_t. Assume that the control input is accurate (has zero mean error) but has covariance R:

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

The plant model for the robot is:

$$x_t = \begin{bmatrix} x \\ y \end{bmatrix}_{t-1} + u_t + \varepsilon$$

where $\begin{bmatrix} x \\ y \end{bmatrix}_{t-1}$ is the estimated location of the robot at time $t-1$ and ε is a zero mean Gaussian random variable with covariance R. The plant model is not linear, because the control u_t depends on the estimated state $\begin{bmatrix} x \\ y \end{bmatrix}_{t-1}$.

Assume that the robot is equipped with a sensor that can measure the distance to each landmark simultaneously and unambiguously (i.e., the robot can measure the distance from its true location to both landmarks and it knows which landmark produces which distance measurement). Assume that the distance measurement is accurate (has zero mean error) but has variance equal to 0.25^2. In other words, each measurement is

$$z_t = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} + \delta$$

where d_1 is the distance to landmark 1, d_2 is the distance to landmark 2, and δ is the zero mean Gaussian measurement noise with covariance

$$Q = \begin{bmatrix} 0.25^2 & 0 \\ 0 & 0.25^2 \end{bmatrix}$$

(a) What is the matrix G_t obtained after linearizing the plant model? Write a Matlab function:

```
function G = plant_jacobian(xs)
```

that returns the plant model Jacobian matrix `G` evaluated at `xs` where `xs` is $\begin{bmatrix} x \\ y \end{bmatrix}_{t-1}$ the estimated location of the robot at time $t-1$.

(b) What is the matrix H_t obtained after linearizing the measurement model? Write a Matlab function:

```
function H = meas_jacobian(xt)
```

that returns the measurement model Jacobian matrix `H` evaluated at `xt` where `xt` is $\begin{bmatrix} x \\ y \end{bmatrix}_{t}$ the predicted location of the robot at time t.

(c) Implement the EKF (using the two functions that you wrote in parts (a) and (b)) for the robot. Write the Matlab function:

```
function [mu, Sigma] = ekf(mu, Sigma, u, z)
```

that returns the estimated state `mu` and covariance matrix `Sigma` at time t given the estimated state `mu` at time $(t-1)$, the estimated covariance `Sigma` at time $(t-1)$, the control input `u` at time t, and the observation `z` at time t (Note: Matlab allows return variables having the same name as the parameters which is why `mu` and `Sigma` appear as return values and function parameters; if this bothers you, feel free to change either the return variable names or the parameter names).

Using the script from the course web site, show the localization results using your EKF at each time step $t = 0, 1, 2, ..., 25$. Your plots should show the estimated location of the robot (using the EKF) and the true location of the robot. Show the results for two different estimates of the starting location x_0:

$$x_0 = \begin{bmatrix} 0 \\ 9 \end{bmatrix}$$

and

$$x_0 = \begin{bmatrix} 0 \\ -5 \end{bmatrix}$$

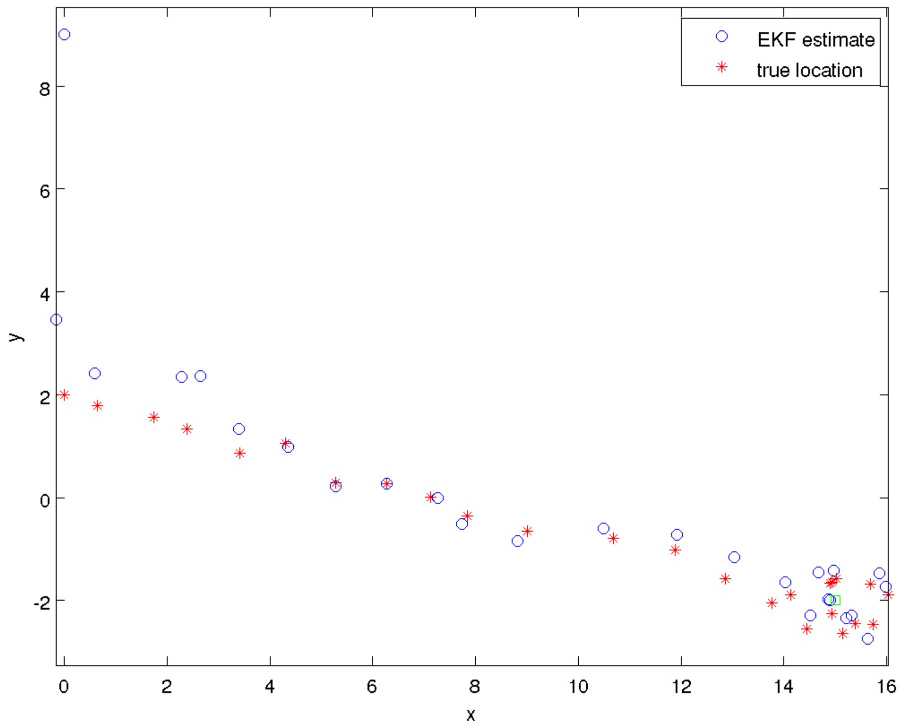
For the estimated state covariance use

$$\Sigma_0 = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

The true starting position of the robot is always

$$x_{\text{true}} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

My results using $x_0 = [0\ 9]^T$ are shown below:



(d) The results using $x_0 = [0 \;\; -5]^T$ should be quite different. Explain why the two different starting points produce such different results.

Submit your written explanation for question (d); you may submit this electronically with your Matlab files if you wish. Submit Matlab scripts (your implementation of the EKF and the scripts needed to produce your two plots) for question (c).

```
submit 4421 lab5 your-matlab-files
```