COMP7308 Introduction to unmanned systems 2021

Assignment 2

:

1 Extended Kalman filter (60 points)

Now you will implement an extended Kalman filter (EKF). A starter code with the EKF work flow is provided for you. A visualization of the EKF state is also provided in the starter code.

The following folders are contained in the ekf.zip achieved file:

data This folder contains files representing the world definition and sensor readings used by the filter.

starter_code This folder contains the extended kalman filter starter code with stubs for you to complete.

You can run the EKF in the terminal: `python kalman_filter.py`. It will only work properly once you filled in the blanks in the code. For all the questions below, beside the code, please also draw some figures in the report to illustrate your result.

Some implementation tips:

- To read in the sensor and landmark data, we have used dictionaries. Dictionaries provide an easier way to access data structs based on single or multiple keys. The functions read sensor data and read world data in the `read_data.py` file read in the data from the files and build a dictionary for each of them with time stamps as the primary keys.

  To access the sensor data from the sensor readings dictionary, you can use

  ```
  sensor_readings[timestamp,'sensor']['id']
  sensor_readings[timestamp,'sensor']['range']
  sensor_readings[timestamp,'sensor']['bearing']
  ```

  and for odometry you can access the dictionary as

  ```
  sensor_readings[timestamp,'odometry']['r1']
  sensor_readings[timestamp,'odometry']['t']
  sensor_readings[timestamp,'odometry']['r2']
  ```

  To access the positions of the landmarks from landmarks dictionary , you can use

  ```
  position x = landmarks[id][0]
  position y = landmarks[id][1]
  ```

(a) Theoretical considerations:

1. (5 points) The Bayes filter processes three probability density functions, i. e., $p(x_t \mid u_t, x_{t-1})$, $p(z_t \mid x_t)$, and $bel(x_t)$. State the normal distributions of the EKF which correspond to these probabilities.
2. (5 points) Explain in a few sentences all of the components of the EKF, i. e., μ_t, Σ_t, g, G_t, h, H_t, Q_t, R_t, K_t and why they are needed. What are the differences and similarities between the KF and the EKF?

(b) EKF prediction step We assume a differential drive robot operating on a 2-dimensional plane, i.e., its state is defined by $\langle x, y, \theta \rangle$. Its motion model is defined according to the odometry model of the motion model lecture (hint: you may want to look at the slide describing `sample_motion_model` to get the state-space formulation of the motion model) .

1. (5 points) Derive the Jacobian matrix G_t of the noise-free motion function g. Do not use Python. Please check the basic derivation we provided in lecture.
2. (5 points) Implement the prediction step of the EKF in the function prediction step using your Jacobian G_t. For the noise in the motion model assume

$$R_t = \begin{pmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.02 \end{pmatrix}.$$

(c) EKF correction step

1. (10 points) Derive the Jacobian matrix H_t of the noise-free measurement function h of a range-only sensor. Do not use Python. Please check the basic derivation we provided in lecture.
2. (10 points) Implement the correction step of the EKF in the function correction step using your Jacobian H_t. For the noise in the sensor model assume that Q_t is the diagonal square matrix

$$Q_t = \begin{pmatrix} 0.5 & 0 & 0 & \cdots \\ 0 & 0.5 & 0 & \cdots \\ 0 & 0 & 0.5 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \in \mathcal{R}^{\text{size}(z_t) \times \text{size}(z_t)}.$$

Once you have successfully implemented all the functions, after running the filter script you should see the state of the robot being plotted incrementally with each time stamp.

(d) (10 points) Add in the python code for taking into account the bearing measurement in the EKF localization.

(e) (10 points) Change the noise level (i.e. the value of Q and R) and discuss the performance of EKF localization under different noise levels using some figures and numbers, e.g. how the estimation error changes when these parameters vary.

2 Estimate the location of an object on a circle (40 points)

An object B moves randomly on a circle with radius 1. The distance to the object can be measured from a sensor installed at the observation point P. The goal is to estimate the location of the object, as shown in Figure 1.

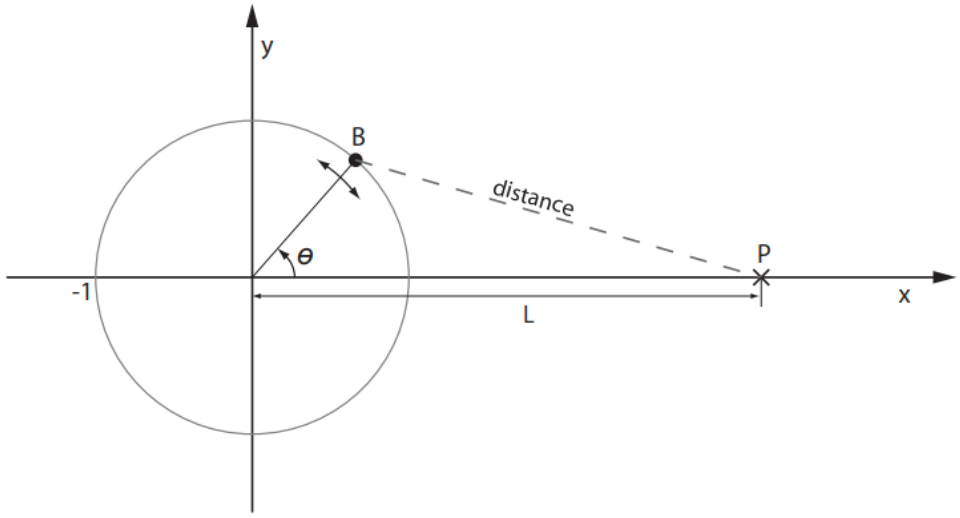

Figure 1: Illustration of the estimation problem

The object B can only move in discrete steps. The object's location at time k is given by $x_k \in \{0, 1, \cdots, N-1\}$, where $\theta_k = 2\pi \frac{x_k}{N}$.

The object dynamics is given by

$$x_k = mod(x_{k-1} + v_k, N), \ k = 1, 2, \cdots,$$

where $v_k = 1$ with probability p and $v_k = -1$ with probability $1 - p$. Recall that the modulo operator has the property $mod(N, N) = 0$ and $mod(-1, N) = N - 1$.

The distance sensor measures $z_k = \sqrt{(L - \cos\theta_k)^2 + (\sin\theta_k)^2} + w_k$, where w_k represents the sensor error which is uniformly distributed on $[-e, e]$. We assume that x_0 is uniformly distributed and x_0, v_k and w_k are independent.

(a) (15 points) Implement the python code to simulate the object movement and implement an estimation algorithm that calculates for each time step k the probability distribution $p(x_k \mid z_1, \cdots, z_k)$. [Hint: the discrete_filter code in self test question 5 may be helpful for you.]

(b) (10 points) Test the following settings and discuss the results: $N = 100$, $x_0 = \frac{N}{4}$, $e = 0.5$.

1. $L = 2$, $p = 0.5$;
2. $L = 2$, $p = 0.55$;
3. $L = 0.1$, $p = 0.55$;
4. $L = 0$, $p = 0.55$.

(c) (15 points) How robust is the algorithm? Set $N = 100$, $x_0 = \frac{N}{4}$, $e = 0.5$, $L = 2$, $p = 0.55$ in the simulation, but use slightly different values for p and e in your estimation algorithm, $\hat{p}$ and $\hat{e}$, respectively. Test the algorithm and explain the result.

1. $\hat{p} = 0.45$, $\hat{e} = e$;
2. $\hat{p} = 0.5$, $\hat{e} = e$;
3. $\hat{p} = 0.9$, $\hat{e} = e$;
4. $\hat{p} = p$, $\hat{e} = 0.9$;
5. $\hat{p} = p$, $\hat{e} = 0.45$.