Assignment 3

Vignesh Raja Ponnamblam

April 10, 2017

1 Exercise 1: Bayes Filter and EKF

1.1 Main Steps of the Bayes Filter

The two main steps of the bayes filter are **prediction** and **correction** steps.

Prediction step basically estimates the next state of the robot from the current robot state i.e where the robot wants to be in order to reach the desired goal. Therefore it creates a belief function to assume the next state of the robot.

Correction step takes sensor values into consideration and corrects the predicted next robot state based on the landmarks detection from the sensors. It generates an optimal path to the goal and also avoids the obstacles at the same time.

1.2 Probability Density functions

1.2.1 State transition probability

The state x at time t will have probability density function (the probability that x will occur under the density which is bounded by specific intervals) based on the previous states, measurements and inputs. Hence it can be written as

$$p(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t}) \tag{1}$$

Hence we know that the x_{t-1} contains all the variables of its past states, measurements and inputs until u_{t-1} and z_{t-1}. So if we know the state x_{t-1} beforehand then only u_t has to be considered since it has a major influence on the current state x at time t. After this theory, we can say that

$$p(x_t|x_{t-1}, u_t) \tag{2}$$

Thus the above equation obtained was based on the **conditional independence**. It can depicted as x_t occurs only during the condition when we have x_{t-1} and u_t and it is completely independent of the other previous states, measurements or inputs.

1.2.2 Measurement probability

The measurement z at time t will have probability density function until the occurrence of the current states, inputs and previous measurements. Hence it can be written as

$$p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t}) \tag{3}$$

Hence we know that the x_t contains all the variables of its past states, measurements and inputs until u_t and z_{t-1}. So if we know the state x_t beforehand then we can eliminate others. Measurements are usually noisy projections of the state. On this concept, we can say that our measurement will have the density function like

$$p(z_t|x_t) \tag{4}$$

1.2.3 Prediction and correction

One of the important term in robot's knowledge was **belief**. It represents the robot's internal knowledge about the state x. The beliefs can be formulated through conditional probability distributions based on the sensors data as

$$bel(x_t) = p(x_t|z_{1:t}, u_{1:t}) \tag{5}$$

A belief distribution will assign a probability (density value) to each possible hypothesis with respect to the true state. Thus it was treated as posterior probabilities over state variables x_t which is conditioned on the interoceptive and exteroceptive sensors data. Thus on considering the measurement probability here, we cannot make a condition z_t before we know that we have the information about x_t .So the belief has to be estimated before finding z_t and the above equation can be changed as

$$\bar{bel}(x_t) = p(x_t|z_{1:t-1}, u_{1:t}) \tag{6}$$

Thus this equation is called as Measurement update or correction.

1.3 Normal distributions of the PDFs

From the bayes terms that we have discussed earlier, we are combining all the terms in order to achieve the final derived formula as

$$p(x_t|z_{1:t}, u_{1:t}) = \frac{p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t})p(x_t|z_{1:t-1}, u_{1:t})}{p(z_t|z_{1:t-1}, u_{1:t})} \tag{7}$$

Which is equivalent to

$Posterior = \frac{Prior\ estimation * Likelihood}{Normalizing factor}$

Thus each element in the above equation has a distribution as in figure 1.

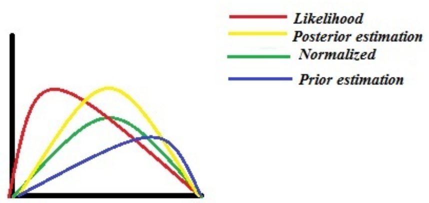

Figure 1: Normal distributions of all bayes terms

- **Likelihood (Belief)**: $p(x_t|z_{1:t-1}, u_{1:t})$-We initialized the prior belief $bel(x_0)$ at time t = 0.Then the Correct estimation has been followed for the update.

- **Prior estimation (Measurement Probability)**: $p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t})$ - Thus the prior information obtained from both the sensors will give the necessary information for estimating the next state by multiplying it with the likelihood distributions.

- **Posterior estimation**: $p(x_t|z_{1:t}, u_{1:t})$ - On multiplying both the above terms, the posterior estimation for the next state as in the figure 1(yellow).

- **Normalized** Only the prior estimation is probability distribution but not the likelihood (just a assumed space so called belief). So we are required to do a normalization in order to integrate the area graph to be one (likelihood can have values more than 1 since it is not a probability value) and to scale our equation. Consequently this can be done by dividing the entire equation with the probability of the current sensor value(z_t) that acts as an normalizing factor.

Therefore the posterior estimation will act as an prior estimation during the next iteration(t=1). After some period of time the distribution shrinks to a smaller density distribution where the probability is **highly likely**.

1.4 EKF paramters

In general, the initialization of map which contains both the robot and landmarks

$$\mu_t = \begin{bmatrix} x \\ M \end{bmatrix} \tag{8}$$

Where x is the robot and M= $L_1, L_2, ..., L_n$ is the set of Landmark states with n number of states. But in the EKF, this initialization was done by mean and co-variance matrix of the state vector. They are named as mean x and variance $\sum_t$.

$$\mu_t = \begin{bmatrix} \bar{x} \\ \bar{M} \end{bmatrix} \tag{9}$$

$$\sum_t = \begin{bmatrix} \sum_{xR} & \sum_{xM} \\ \sum_{Mx} & \sum_{xM} \end{bmatrix} \tag{10}$$

g is a non-linear function which creates velocity motion model based on Odometry data. Therefore it sums up the current robot pose and the estimated next pose by the known linear and angular velocities. G_t^x is a 3*3 Jacobian matrix which involves the motion function g with respect to the pose of the robot.

$$G_t^x = \frac{\delta}{\delta(x, y, \theta)} * g(u_t, y_{t-1}) \tag{11}$$

G_t is updating the G_t^x matrix along with the 2N*2N identity matrices representing the number of landmarks. Q correspond to the process noise co-variance matrices respectively which will happen while taking the input readings. So it was taken into account that thus the process noise w is drawn from N(0,Q) distribution with co-variance matrix Q. Consequently it's value was added in the kalman filter general equations. The measurement and process noises Q_t and R_t have to be defined and their variances are q and r, respectively.

$$Q_t = diag([0.1 \quad 0.1]^2); \tag{12}$$

$$R_t^x = diag([0.1 \quad 1 * pi/180]^2); \tag{13}$$

Just like G_t, R_t is also updating the R_t^x matrix along with the 2N*2N identity matrices representing the number of landmarks.

The observation function h is a non-linear function that provides the estimated measurements from the estimated states. The measurements will be range and bearing values (in general) so it will be $2*1$ vector usually. Hence for the case of the generalization for multiple variables and equations we need the **Jacobian matrix H**. Consequently, the jacobian matrix H was derived with respect to the estimated state variables. The H matrix will be $2*5$ in this case with respect to x, y, θ, m_x, m_y.

Basically the gain controls how much the user can trust the measurements over the estimation. By the way, kalman gain can be formulated using Jacobian H as

$$K = PH' * (R + HPH')^{-1} \tag{14}$$

Kalman gain will be just a scalar value finally after all the computations.

2 Exercise 2

2.1 Obtaining G_t^x

The jacobian matrix G_t^x is determined by differentiating the g function with respect to x,y and θ

$$G_t^x = \begin{bmatrix} 1 & 0 & -\delta_{trans} cos(\theta_{t-1} + \delta_t) \\ 0 & 1 & \delta_{trans} sin(\theta_{t-1} + \delta_t) \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

2.2 Obtaining H_t

The jacobian matrix H_t is determined by differentiating the h function with respect to x,y, θ, m_x and m_y.

$$H_t = \frac{1}{q} \begin{bmatrix} -\sqrt[2]{q}\delta_x & -\sqrt[2]{q}\delta_y & 0 & \sqrt[2]{q}\delta_x & \sqrt[2]{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{bmatrix} \tag{16}$$