CSE 483: Mobile Robotics

Lecture by: Prof. K. Madhava Krishna **Lecture** # 04
Scribe: Dhaivat Bhatt, Isha Dua **Date**: 14th November, 2016 (Monday)

Extended Kalman filter for localization(worked out example)

This document briefly walks through computational aspects of **EKF** localization for a toy example. This document expects a reader to have familiarity with **EKF** filtering technique to solve localization problem.

Here is the algorithm to perform **EKF localization**, At each iteration, state space representation of the the robot($\mu_{\mathbf{t-1}}$), co-variance of the state space($\mathbf{\Sigma_{t-1}}$) and control($\mathbf{u_t}$) at time $\mathbf{t}$ are fed to the algorithm. The algorithm returns updated state space representation($\mu_{\mathbf{t}}$) of the robot and uncertainty associated with the robot state. ($\mathbf{\Sigma_t}$).

$$\begin{aligned}
&\mathbf{1} : \mathbf{EXTENDED_KALMAN_FILTER}(\mu_{\mathbf{t-1}}, \mathbf{\Sigma_{t-1}}, \mathbf{u_t}, \mathbf{z_t}) \\
&\mathbf{2} : \bar{\mu_{\mathbf{t}}} \quad = \quad \mathbf{g}\ (\ \mathbf{u_t}\ , \mu_{\mathbf{t-1}}) \\
&\mathbf{3} : \overline{\mathbf{\Sigma_t}} \quad = \quad \mathbf{G_t}\ \mathbf{\Sigma_{t-1}}\ \mathbf{G_t^T} \quad + \quad \mathbf{R_t} \\
&\mathbf{4} : \mathbf{K_t} \quad = \quad \bar{\mathbf{\Sigma_t}}\ \mathbf{H_t^T}\ (\ \mathbf{H_t}\ \bar{\mathbf{\Sigma_t}}\ \mathbf{H_t^T} + \mathbf{Q_t})^{-\mathbf{1}} \\
&\mathbf{5} : \mu_{\mathbf{t}} \quad = \quad \bar{\mu_{\mathbf{t}}}\ +\ \mathbf{K_t}\ (\ \mathbf{z_t}\ -\ \mathbf{h}\ (\bar{\mu_{\mathbf{t}}})) \\
&\mathbf{6} : \mathbf{\Sigma_t} \quad = \quad (\ \mathbf{I}\ -\ \mathbf{K_t}\ \mathbf{H_t}\ )\bar{\mathbf{\Sigma_t}} \\
&\mathbf{7} : \mathbf{return} \quad \mu_{\mathbf{t}}, \quad \mathbf{\Sigma_t}
\end{aligned}$$

Overview:

Here, we are considering a simple problem where our robot can navigate in a 2D environment. Features of the environment(landmarks) are already known. For this problem, our environment has 4 landmarks.

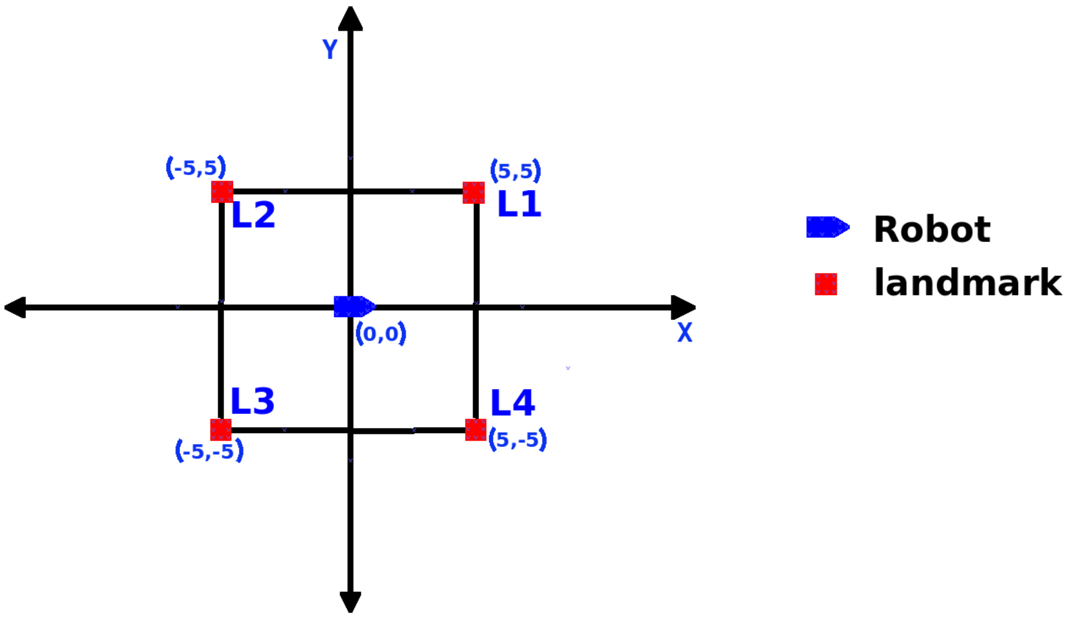
In this document, we are considering **2** timesteps. We are solving this problem using batch mode of Extended Kalman filter. Meaning, we execute correction step of EKF only once. We incorporate entire sensor observation in a single go to correct predicted state of the robot.

Data association problem is assumed to be solved. One landmark measurement will have two components. Absolute distance between robot and the landmark, relative heading of the robot and landmark.

The four landmarks considered are $\mathbf{l1} = (\mathbf{5}, \mathbf{5})$, $\mathbf{l2} = (-\mathbf{5}, \mathbf{5})$, $\mathbf{l3} = (-\mathbf{5}, -\mathbf{5})$ and $\mathbf{l4} = (\mathbf{5}, -\mathbf{5})$. ID of **l1** is **1**, ID of **l2** is **2**, ID of **l3** is **3** and ID of **l4** is **4**.

Environmental configuration:

Here is the figure of initial state of the environment. Red objects in the figure are the landmarks and Blue object at the origin is initial position of the robot. The pointy end of the object is heading of the robot.



At time $\mathbf{t = 0}$, robot is at the origin of the co-ordinate system. Heading of the robot is in positive direction of $\mathbf{X-axis}$. Heading of the robot for this problem is measured with respect to positive direction of $\mathbf{X-axis}$. Robot orientation varies between $(-\pi,\ \pi]$. At time $\mathbf{t = 0}$, uncertainty in robot position is assumed to be $\mathbf{0}$. Meaning, co-variance matrix of state space representation is a **3x3 zero** matrix at initialization.

$$\mu_{\mathbf{0}} = (\mathbf{0}, \mathbf{0}, \mathbf{0})^T$$

$$\mathbf{\Sigma_0} \ = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Controls:

$$\mathbf{u_1} = \begin{pmatrix} \mathbf{3} \\ \pi/\mathbf{6} \end{pmatrix} \qquad \mathbf{u_2} = \begin{pmatrix} \mathbf{4} \\ \mathbf{7}\pi/\mathbf{36} \end{pmatrix}$$

Sensor observations are initialized with zeros.

$$\mathbf{Z_t} = (\underbrace{\mathbf{0}, \mathbf{0}}_{\mathbf{l1}}, \underbrace{\mathbf{0}, \mathbf{0}}_{\mathbf{l2}}, \underbrace{\mathbf{0}, \mathbf{0}}_{\mathbf{l3}}, \underbrace{\mathbf{0}, \mathbf{0}}_{\mathbf{l4}})^T$$

Control noise($\mathbf{R_t}$) is a **3x3** matrix initialized as below.

$$\text{Control noise } \mathbf{R_t} = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$$

$$\text{Observation noise } \mathbf{Q} = \begin{pmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 \end{pmatrix}$$

Here, $\mathbf{Q_t}$ is co-variance of the sensor noise. Let's say the control $\mathbf{u_1}$ is executed. Now out task is to get best estimate of the new state by incorporating all the uncertainties we have modeled. Now, we will extensively go through aspects of applying **EKF** for localization setting.

Estimating new robot state after applying control $\mathbf{u_1}$:

Step 1: Predicting new state of the robot using motion model

$$\bar{\mu_1} \quad = \quad \mathbf{g}\ (\ \mathbf{u_1}\ , \mu_0)$$

$$\bar{\mu_1} = \begin{pmatrix} \bar{\mu}_{x_1} \\ \bar{\mu}_{y_1} \\ \bar{\mu}_{\theta_1} \end{pmatrix} = \begin{pmatrix} \mu_{x_0} \\ \mu_{y_0} \\ \mu_{\theta_0} \end{pmatrix} + \begin{pmatrix} T\cos(\mu_{\theta_0} + \phi) \\ T\sin(\mu_{\theta_0} + \phi) \\ \phi \end{pmatrix}$$

$$\bar{\mu_1} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 3\cos(0 + \pi/6) \\ 3\sin(0 + \pi/6) \\ \pi/6 \end{pmatrix} = \begin{pmatrix} 2.5981 \\ 1.5000 \\ 0.5236 \end{pmatrix}$$

Step 2: Predicting new co-variance of the robot state

$$\bar{\mathbf{\Sigma_1}} \quad = \quad \mathbf{G_1}\ \mathbf{\Sigma_0}\ \mathbf{G_1^T} \quad + \quad \mathbf{R_t}$$

$$\mathbf{G_1} = \begin{pmatrix} 1.0000 & 0 & -1.5000 \\ 0 & 1.0000 & 2.5981 \\ 0 & 0 & 1.0000 \end{pmatrix}$$

$$\bar{\mathbf{\Sigma_1}} = \begin{pmatrix} 1.0000 & 0 & -1.5000 \\ 0 & 1.0000 & 2.5981 \\ 0 & 0 & 1.0000 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1.0000 & 0 & 0 \\ 0 & 1.0000 & 0 \\ -1.5000 & 2.5981 & 1.0000 \end{pmatrix} + \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$$

$$\bar{\mathbf{\Sigma_1}} = \begin{pmatrix} 0.1000 & 0 & 0 \\ 0 & 0.2000 & 0 \\ 0 & 0 & 0.3000 \end{pmatrix} \tag{1}$$

Above equation(**1**) is predicted estimate of uncertainty in the state of the robot. As we can see, uncertainty in robot state has increased because of the control co-variance.($\mathbf{R_t}$) Because, uncertainty in robot state was zero before applying control($\mathbf{u_1}$).

Step 3: Estimating Kalman Gain($\mathbf{K_t}$)

$$\mathbf{K_t} \quad = \quad \bar{\mathbf{\Sigma_1}}\ \mathbf{H_t^T}\ (\ \mathbf{H_t}\ \bar{\mathbf{\Sigma_t}}\ \mathbf{H_t^T} + \mathbf{Q})^{-\mathbf{1}}$$

In this step, we compute Kalman gain($\mathbf{K_t}$). $\mathbf{K_t}$ signifies which of the two estimates to trust more. We fuse two independent estimates of robot poses. One we get from motion model(**Step 1**) and other we get by incorporating sensor observation of the surrounding environment. The disparity between what robot is perceiving($\mathbf{Z_t}$) and what robot is expected to perceive($\mathbf{h}(\bar{\mu_\mathbf{t}})$) will help us to get more precise estimate of robot state and will also reduce uncertainty of the same.

$$\begin{pmatrix} \mathbf{r} \\ \psi \end{pmatrix} = \begin{pmatrix} \sqrt{(\overline{\mu}_{x_{t+1}} - m_x)^2 + (\overline{\mu}_{y_{t+1}} - m_y)^2} \\ tan^{-1}((\overline{\mu}_{y_{t+1}} - m_y)/(\overline{\mu}_{x_{t+1}} - m_x)) - \overline{\mu}_{\theta_{t+1}} \end{pmatrix} \tag{2}$$

In order to estimate Kalman gain, we need to estimate observation jacobian($\mathbf{H_t}$). Since we are performing batch mode update, size of $\mathbf{H_t}$ will be **2nx3**. Where, **n** is the number of landmarks. In our case, size of $\mathbf{H_t}$ will be **8x3**.

Let's say our robot is firing sensors, and is able to perceive two landmarks **l1** and **l2** out of four. Hence, first two rows(for **l1** and next two rows(for **l2** will have non-zero entries. Rows $\mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{8}$ will have all entries as $\mathbf{0}$.

Let's say following are the expected observations for landmarks **l1** and **l2**. These are computed using equation **2**.

$$\begin{pmatrix} \mathbf{r1} \\ \psi\mathbf{1} \end{pmatrix} = \begin{pmatrix} \mathbf{4.2445} \\ \mathbf{0.4457} \end{pmatrix} \tag{3}$$

$$\begin{pmatrix} \mathbf{r2} \\ \psi\mathbf{2} \end{pmatrix} = \begin{pmatrix} \mathbf{8.3654} \\ \mathbf{1.9775} \end{pmatrix} \tag{4}$$

$$\mathbf{H_1} = \begin{pmatrix} -\cos(\overline{\mu}_{\theta_1} + \psi 1) & -\sin(\overline{\mu}_{\theta_1} + \psi 1) & 0 \\ \sin(\overline{\mu}_{\theta_1} + \psi 1)/r1 & -\cos(\overline{\mu}_{\theta_1} + \psi 1)/r1 & -1 \\ -\cos(\overline{\mu}_{\theta_1} + \psi 2) & -\sin(\overline{\mu}_{\theta_1} + \psi 2) & 0 \\ \sin(\overline{\mu}_{\theta_1} + \psi 2)/r2 & -\cos(\overline{\mu}_{\theta_1} + \psi 2)/r2 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Now, we estimate jacobian of the observation model and evaluate it at our expected observations($\mathbf{3}, \mathbf{4}$). we plug appropriate values in the above equation to compute the jacobian.

$$\mathbf{H_1} = \begin{pmatrix} -0.5660 & -0.8244 & 0 \\ 0.1942 & -01330 & -1.0000 \\ 0.8018 & -0.5976 & 0 \\ 0.0714 & 0.0958 & -1.0000 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Now, we have $\bar{\mathbf{\Sigma_1}}$, $\mathbf{H_1}$ and $\mathbf{Q_t}$. We plug in these matrices and estimate the Kalman gain($\mathbf{K_1}$).

$$\textbf{Kalman gain } \mathbf{K_1} = \begin{pmatrix} -0.2926 & 0.0236 & -0.4028 & -0.0068 & 0 & 0 & 0 & 0 \\ -0.5357 & -0.0381 & -0.3803 & 0.0353 & 0 & 0 & 0 & 0 \\ -0.0217 & -0.3722 & 0.0454 & -0.3762 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Step 4: Correcting robot state:

Following is the observation vector($\mathbf{h}(\bar{\mu_t})$) for expected observation.

$$\mathbf{h}(\bar{\mu_1}) = \begin{pmatrix} \mathbf{r1} \\ \phi\mathbf{1} \\ \mathbf{r2} \\ \phi\mathbf{2} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 4.2425 \\ 0.4457 \\ 8.3654 \\ 1.9775 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Let's say this is what robot is perceiving after through it sensors. Let's say following are the readings of robot sensors($\mathbf{Z_t}$).

$$\mathbf{Z_1} = \begin{pmatrix} 4.2194 \\ 0.4861 \\ 8.3076 \\ 2.0483 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Now, we take disparity between sensor observations($\mathbf{Z_t}$) and expected observations($\mathbf{h}(\bar{\mu_t})$) to correct predicted state of the robot. Following is the updated estimate of robot pose(μ_1).

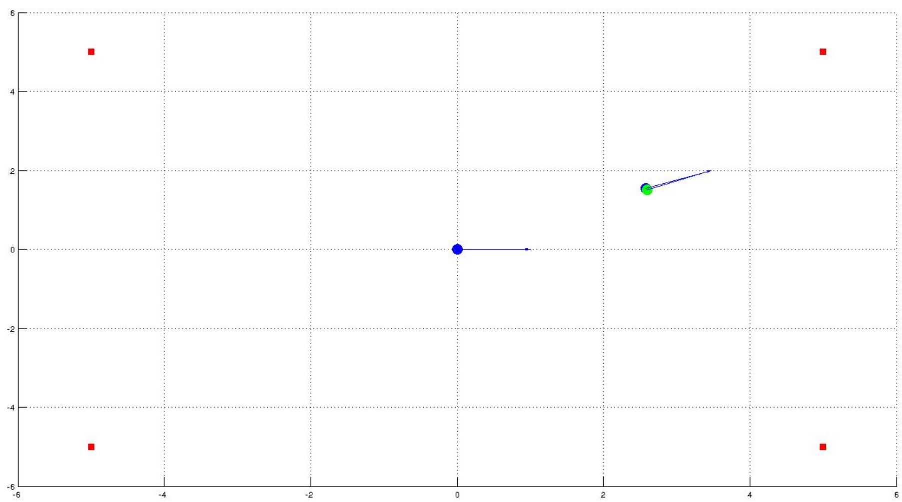
$$\mu_1 = \begin{pmatrix} \mathbf{2.5826} \\ \mathbf{1.5364} \\ \mathbf{0.4799} \end{pmatrix}$$

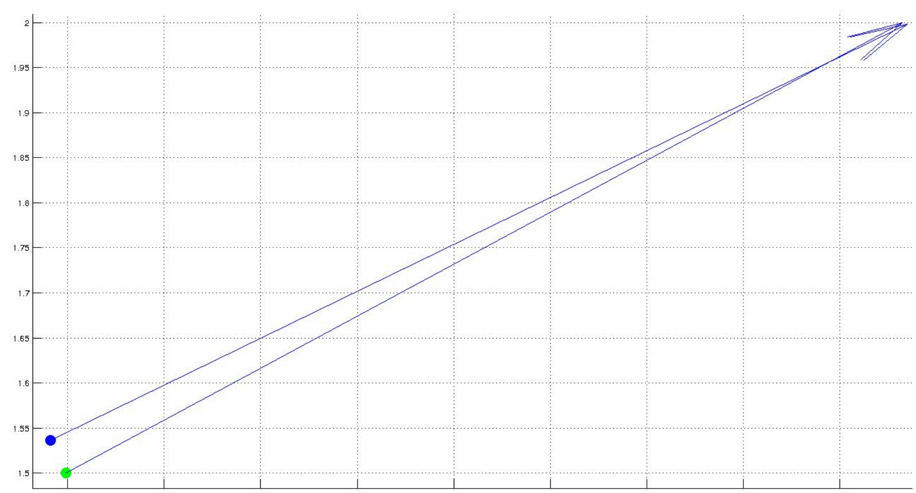
Step 5: Updating robot uncertainty:

$$\boldsymbol{\Sigma_1} \quad = \quad (\,\mathbf{I}\,-\,\mathbf{K_1}\,\mathbf{H_1}\,)\bar{\boldsymbol{\Sigma}}_{\mathbf{1}}$$

This is the estimated robot uncertainty after fusing sensor data($\boldsymbol{\Sigma_1}$).

$$\boldsymbol{\Sigma_1} = \begin{pmatrix} 0.0507 & 0.0007 & 0.0050 \\ 0.0007 & 0.0645 & -0.0008 \\ 0.0050 & -0.0008 & 0.0755 \end{pmatrix}$$





This is end of the algorithm. We run it each time we apply control. In the above figure. Red are the landmarks locations. Robot was initially at $(\mathbf{0}, \mathbf{0})$. After applying $\mathbf{u_1}$, it should have reached at $(\mathbf{2.5981}, \mathbf{1.5})$, which is denoted by green dot in second figure of the previous page. But because of control error, it didn't reach its destination. Once we estimate final state using Kalman filter, the best estimate is denoted by blue dot previous figure.

This is full cycle of Extended Kalman filter for localization setting. Now, new state of the robot is $\mu_{\mathbf{1}} = (\mathbf{2.5826}, \mathbf{1.5364})$. New co-variance will be $\mathbf{\Sigma_1}$. After executing control $\mathbf{u_2}$, $\mathbf{\Sigma_1}$, $\mu_{\mathbf{1}}$ and $\mathbf{u_2}$ will be fed to the **EKF** algorithm, and again same procedure will be followed to estimate new state.

If you find any mistake in calculation, feel free to shoot an email at ***dhaivat1994@gmail.com***.