16-833A Robot Localisation and Mapping, Spring 2023
Homework 2: SLAM using Extended Kalman Filter (EKF-SLAM)

Sushanth Jayanth

Andrew ID: sushantj

March 16, 2023

Contents

Introduction to EKF

In this assignment, we use EKF to localize a robot and landmarks in 2D space. The state space of the robot and landmarks are therefore fused into one large state vector.

$$p_t = (\underbrace{x, y, \theta}_{\text{robot's pose}}, \underbrace{l_{1,x}, l_{1,y}}_{\text{landmark 1}}, \ldots, \underbrace{l_{n,x}, l_{n,y}}_{\text{landmark n}})^T$$

The above state vector captures **robot pose and landmark position in global coordinates**. Robot pose has three variables (x,y,theta) and landmark position has two variables (x,y) for each landmark. The robot can move only along the x-axis of the robot frame defined as d_t and rotate by α. This is captured in the diagram below:

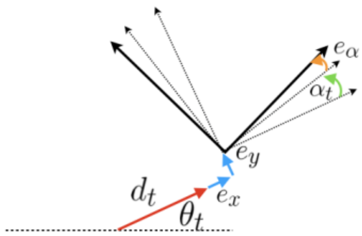


Figure 1: Robot Motion Model

As seen above, the movement has some noise. This is the noise introduced due to **control (process noise)** and this alone effects the robot's pose. Another type of noise exists in the laser rangefinder which gives the landmark locations. This is the **measurement noise**. Since both control and measurement noise affects the robot state, it is easy to maintain a combined covariance matrix P. The below equation shows how the state vector relates to the mean and covariance matrices.

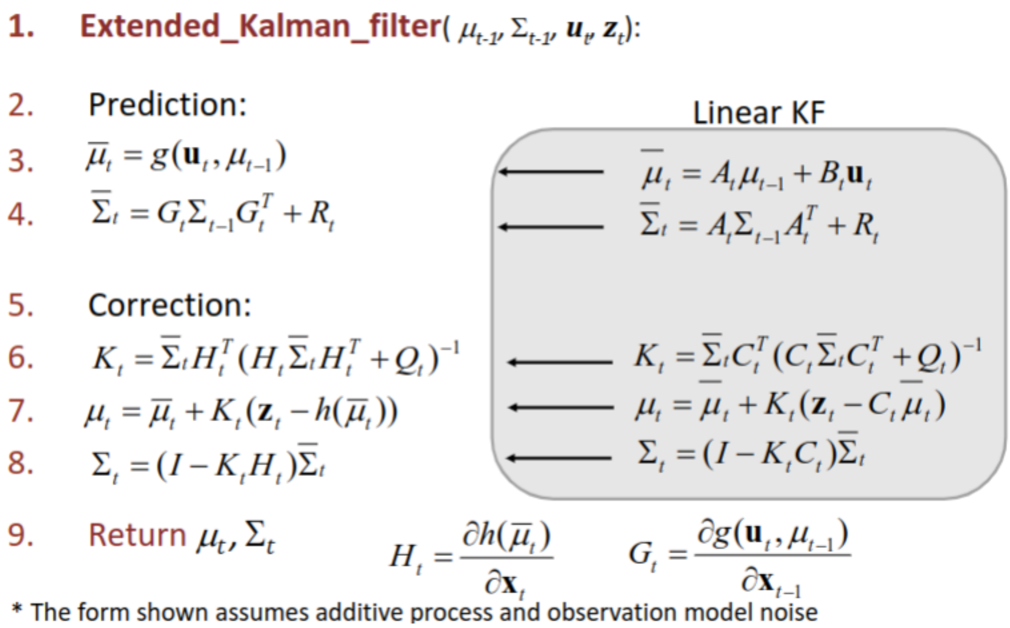
$$\underbrace{\begin{pmatrix} x_t \\ l_1 \\ \vdots \\ l_n \end{pmatrix}}_{\mu} \underbrace{\begin{pmatrix} \Sigma_{x_t x_t} & \Sigma_{x_t l_1} & \cdots & \Sigma_{x_t l_n} \\ \Sigma_{l_1 x_t} & \Sigma_{l_1 l_1} & \cdots & \Sigma_{l_1 l_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{l_n x_t} & \Sigma_{l_n l_1} & \cdots & \Sigma_{l_n l_n} \end{pmatrix}}_{\Sigma}$$

$$x_t = \text{robot pose} = (x, y, \theta)$$

In the EKF algorithm we will update parts the above covariance matrix in different steps:

- Given Control Reading : we update the $\Sigma_{x_t x_t}$ primarily plus the first row and first column elements
- Given Sensor Reading : we update the whole covariance matrix

However, prior to every update step, we will also make predictions on where the robot pose or landmark pose is supposed to be and then compare that to the measurement during our update/correction steps (lines 6,7,8 in EKF algorithm below) and (1)



1 Theory

1.1 Pose Prediction in Noise Free Case

$$p_{t+1} = p_t + \begin{bmatrix} d_t \cdot \cos(\theta_t) \\ d_t \cdot \sin(\theta_t) \\ \alpha_t \end{bmatrix} \tag{1}$$

$$p_{t+1} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} d_t \cdot \cos(\theta_t) \\ d_t \cdot \sin(\theta_t) \\ \alpha_t \end{bmatrix} \tag{2}$$

1.2 Pose Prediction with Uncertainty

$$p_{t+1} = \begin{bmatrix} x_t + d_t \cdot \cos(\theta_t) \\ y_t + d_t \cdot \sin(\theta_t) \\ \theta_t + \alpha_t \end{bmatrix} + \begin{bmatrix} e_y \sin(\theta_t) \\ e_x \cos(\theta_t) \\ e_\alpha \end{bmatrix} \tag{3}$$

The above equation can be represented as a non-linear function $g(x_t, u_t)$ with some added noise ϵ_t

$$p_{t+1} = g(x_t, u_t) + \epsilon_t \tag{4}$$

The covariance of the predicted pose p_{t+1} is defined as:

$$\Sigma_{t+1} = G_t \cdot \Sigma_t \cdot G_t^T + V_t \cdot Q_t \cdot V_t^T \tag{5}$$

G_t = Jacobian of $g(x_t, u_t)$ with respect to robot pose

$$G_t = \left.\frac{\partial g(x_t, u_t)}{\partial p}\right|_{p_t} \tag{6}$$

$$= \begin{bmatrix} \frac{\partial g_1}{\partial x_t} & \frac{\partial g_1}{\partial y_t} & \frac{\partial g_1}{\partial \theta_t} \\ \frac{\partial g_2}{\partial x_t} & \frac{\partial g_2}{\partial y_t} & \frac{\partial g_2}{\partial \theta_t} \end{bmatrix} \tag{7}$$

$$= \begin{bmatrix} 1 & 0 & -d_t \cdot \sin(\theta_t) \\ 0 & 1 & d_t \cdot \cos(\theta_t) \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

V_t = Transform mapping error in robot frame to local frame (also can be computed as the Jacobian of $g(x_t, u_t)$ with respect to e_x, e_y, e_α

Here we define V_t as a rotation matrix (about the z-axis)

$$V_t = \begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) & 0 \\ \sin(\theta_t) & \cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Q_t is the Control noise.

$$Q_t = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \tag{10}$$

1.3 Estimate Landmark Position

Given the range r and bearing β readings from a laser rangefinder, we can estimate the location of landmarks in the global frame given a known robot state. We will use this as our measurement prediction model $h(p_t, \beta, r)$

$$\begin{bmatrix} lx \\ ly \end{bmatrix} = \begin{bmatrix} x_t + r \cdot \cos(\beta + \theta_t) \\ y_t + r \cdot \sin(\beta + \theta_t) \end{bmatrix} \tag{11}$$

The noise in measurement of the rangefinder is given below.

$$\eta_\beta \sim \mathcal{N}(0,\, \sigma_\beta^2)\,. \tag{12}$$

$$\eta_r \sim \mathcal{N}(0,\, \sigma_r^2)\,. \tag{13}$$

The predicted covariance in the landmark measurement is shown below.

$$\Sigma_{l_{t+1}} = G_{l_t} \cdot \Sigma_{l_t} \cdot G_{l_t} + L_{l_t} \cdot R_t \cdot L_{l_t}^T \tag{14}$$

Where G_{l_t} = Jacobian of the measurement model w.r.t the robot pose

$$G_{l_t} = \left.\frac{\partial l}{\partial p}\right|_{p_t} \tag{15}$$

$$= \begin{bmatrix} \frac{\partial l_1}{\partial x_t} & \frac{\partial l_1}{\partial y_t} & \frac{\partial l_1}{\partial \theta_t} \\ \frac{\partial l_2}{\partial x_t} & \frac{\partial l_2}{\partial y_t} & \frac{\partial l_2}{\partial \theta_t} \end{bmatrix} \tag{16}$$

$$= \begin{bmatrix} 1 & 0 & -r \cdot \sin(theta_t + \beta) \\ 0 & 1 & r \cdot \cos(theta_t + \beta) \end{bmatrix} \tag{17}$$

L_{l_t} = Jacobian of the measurement model w.r.t the range and bearing

$$L_{l_t} = \left.\frac{\partial l}{\partial z}\right|_{p_t} \tag{18}$$

$$= \begin{bmatrix} \frac{\partial l_1}{\partial \beta} & \frac{\partial l_1}{\partial r} \\ \frac{\partial l_2}{\partial \beta} & \frac{\partial l_2}{\partial r} \end{bmatrix} \tag{19}$$

$$L_{l_t} = \begin{bmatrix} \cos(theta_t + \beta) & -r \cdot \sin(theta_t + \beta) \\ \sin(theta_t + \beta) & r \cdot \cos(theta_t + \beta) \end{bmatrix} \tag{20}$$

R_t is the measurement noise.

$$R_t = \begin{bmatrix} \sigma_\beta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} \tag{21}$$

1.4 Bearing and Range from Measurement

Using predicted landmark location l_x and l_y, the bearing β and range r is captured as $h(\beta, r)$

$$h(\beta, r) = \begin{bmatrix} \beta \\ r \end{bmatrix} = \begin{bmatrix} \text{warp2pi}\left(\arctan 2\left(l_y - y_t, l_x - x_t\right) - \theta_t\right) \\ \sqrt{\left(l_x - x_t\right)^2 + \left(l_y - y_t\right)^2} \end{bmatrix} \tag{22}$$

1.5 Jacobian of Measurement w.r.t Robot Pose (H_p)

Jacobian of $h(\beta, r)$ w.r.t to robot pose is defined below.

$$H_p = \left.\frac{\partial h}{\partial p}\right|_{p_t} \tag{23}$$

$$= \begin{bmatrix} \frac{\partial h_1}{\partial x_t} & \frac{\partial h_1}{\partial y_t} & \frac{\partial h_1}{\partial \theta_t} \\ \frac{\partial h_2}{\partial x_t} & \frac{\partial h_2}{\partial y_t} & \frac{\partial h_2}{\partial \theta_t} \end{bmatrix} \tag{24}$$

$$\therefore H_p = \begin{bmatrix} \frac{l_y - y_t}{(l_x - x_t)^2 + (l_y - y_t)^2} & \frac{-(l_x - x_t)}{(l_x - x_t)^2 + (l_y - y_t)^2} & -1 \\ \frac{-(l_x - x_t)}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} & \frac{-(l_y - y_t)}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} & 0 \end{bmatrix} \tag{25}$$

1.6 Jacobian of Measurement w.r.t Landmark Position ($\mathbf{H}_l$)

Jacobian of $h(\beta, r)$ w.r.t to landmark (l_x and l_y) is defined below.

$$H_l = \left.\frac{\partial h}{\partial l}\right|_{p_t} \tag{26}$$

$$= \begin{bmatrix} \frac{\partial h_1}{\partial l_x} & \frac{\partial h_1}{\partial l_y} \\ \frac{\partial h_2}{\partial l_x} & \frac{\partial h_2}{\partial l_y} \end{bmatrix} \tag{27}$$

$$\therefore H_l = \begin{bmatrix} \frac{-(l_y - y_t)}{(l_x - x_t)^2 + (l_y - y_t)^2} & \frac{l_x - x_t}{(l_x - x_t)^2 + (l_y - y_t)^2} \\ \frac{l_x - x_t}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} & \frac{l_t}{\sqrt{(l_x - x_t)^2 + (l_y - y_t)^2}} \end{bmatrix} \tag{28}$$

References

[1] Wolfram Burgard Sebastian Thrun and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.