LU 7 - Individual Assignment: Localization with EKF

Sayeeda Begam Mohamed Ikbal

December 9, 2024

Question

Exercise 1: EKF Prediction Step

We assume a differential drive robot operating on a 2-dimensional plane, i.e., its state is defined by (x, y, θ). Its motion model is defined on the slides of the odometry motion model module.

(a) Derive the Jacobian matrix G_t of the noise-free motion function. Do not use Python.

(b) 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:

$$Q_t = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$$

Exercise 2: EKF Correction Step

We consider a sensor that measures distances to landmarks in the environment. The sensor also provides the IDs of the currently visible landmarks, thus, we do not have to take care of data association.

(a) Derive the Jacobian matrix H_t of the noise-free measurement function of this range-only sensor. Do not use Python.

(b) 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 R_t is the diagonal square matrix.

$$R_t = \begin{bmatrix} 0.5 & 0 & 0 & \cdots \\ 0 & 0.5 & 0 & \cdots \\ 0 & 0 & 0.5 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \in \mathbb{R}^{\text{size}(z_t) \times \text{size}(z_t)}$$

1 Exercise : EKF Prediction Step

Jacobian G_t of the noise-free motion function

The motion function $g(u_t, x_{t-1})$ describes the deterministic motion of the robot, given by:

$$x_t = g(u_t, x_{t-1}) = \begin{bmatrix} x_{t-1} + \frac{v_t}{\omega_t}\big(- \sin(\theta_{t-1}) + \sin(\theta_{t-1} + \omega_t \Delta t)\big) \\ y_{t-1} + \frac{v_t}{\omega_t}\big(\cos(\theta_{t-1}) - \cos(\theta_{t-1} + \omega_t \Delta t)\big) \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix}$$

where:

- v_t: translational velocity,
- ω_t: rotational velocity,
- Δt: time step.

Definition of Jacobian G_t

The Jacobian G_t is defined as:

$$G_t = \frac{\partial g(u_t, x_{t-1})}{\partial x_{t-1}}$$

From the motion model, the partial derivatives are:

$$G_t = \begin{bmatrix} \frac{\partial g_x}{\partial x_{t-1}} & \frac{\partial g_x}{\partial y_{t-1}} & \frac{\partial g_x}{\partial \theta_{t-1}} \\ \frac{\partial g_y}{\partial x_{t-1}} & \frac{\partial g_y}{\partial y_{t-1}} & \frac{\partial g_y}{\partial \theta_{t-1}} \\ \frac{\partial g_\theta}{\partial x_{t-1}} & \frac{\partial g_\theta}{\partial y_{t-1}} & \frac{\partial g_\theta}{\partial \theta_{t-1}} \end{bmatrix}$$

Partial Derivatives

First Row: x-derivatives

$$\frac{\partial g_x}{\partial x_{t-1}} = 1$$

(Robot's motion in x-direction is independent of its previous location x_{t-1}.)

$$\frac{\partial g_x}{\partial y_{t-1}} = 0$$

(No direct dependency of x-motion on y_{t-1}.)

$$\frac{\partial g_x}{\partial \theta_{t-1}} = v_t \omega_t \left(-\cos(\theta_{t-1}) + \cos(\theta_{t-1} + \omega_t \Delta t)\right)$$

(This derivative represents the effect of orientation θ_{t-1} on the motion in x-direction. The term arises from the non-linear motion of the robot.)

Second Row: y-derivatives

$$\frac{\partial g_y}{\partial x_{t-1}} = 0$$

(No direct dependency of y-motion on x_{t-1}.)

$$\frac{\partial g_y}{\partial y_{t-1}} = 1$$

(Robot's motion in y-direction is independent of its previous location y_{t-1}.)

$$\frac{\partial g_y}{\partial \theta_{t-1}} = v_t \omega_t \left(-\sin(\theta_{t-1}) + \sin(\theta_{t-1} + \omega_t \Delta t)\right)$$

(This derivative represents the effect of orientation θ_{t-1} on the motion in y-direction. The term captures the influence of rotation.)

Third Row: θ-derivatives

$$\frac{\partial g_\theta}{\partial x_{t-1}} = 0$$

(No dependency of orientation θ_t on x_{t-1}.)

$$\frac{\partial g_\theta}{\partial y_{t-1}} = 0$$

(No dependency of orientation θ_t on y_{t-1}.)

$$\frac{\partial g_\theta}{\partial \theta_{t-1}} = 1$$

(Robot's orientation evolves linearly with respect to the previous orientation.)

Jacobian Matrix

Combining these partial derivatives, the Jacobian G_t is:

$$G_t = \begin{bmatrix} 1 & 0 & -\frac{v_t}{\omega_t}\left(\cos(\theta_{t-1}) - \cos(\theta_{t-1} + \omega_t \Delta t)\right) \\ 0 & 1 & -\frac{v_t}{\omega_t}\left(\sin(\theta_{t-1}) - \sin(\theta_{t-1} + \omega_t \Delta t)\right) \\ 0 & 0 & 1 \end{bmatrix}$$

2 Exercise: EKF Correction Step

Jacobian H_t of the Measurement Function

The sensor measures the range r and bearing ϕ to a landmark located at $m_j = (m_{j,x}, m_{j,y})$. The predicted measurement is given by:

$$\hat{z}_t = \begin{bmatrix} r \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(m_{j,x} - \mu_{t,x})^2 + (m_{j,y} - \mu_{t,y})^2} \\ \arctan 2(m_{j,y} - \mu_{t,y}, m_{j,x} - \mu_{t,x}) - \mu_{t,\theta} \end{bmatrix},$$

where:

- r: Range to the landmark,

- ϕ: Bearing to the landmark,
- $\mu_t = (\mu_{t,x}, \mu_{t,y}, \mu_{t,\theta})$: State of the robot.

The Jacobian H_t is the partial derivative of the measurement function h with respect to the robot state μ_t, given by:

$$H_t = \frac{\partial h}{\partial x_t} = \begin{bmatrix} \frac{\partial r}{\partial \mu_{t,x}} & \frac{\partial r}{\partial \mu_{t,y}} & \frac{\partial r}{\partial \mu_{t,\theta}} \\ \frac{\partial \phi}{\partial \mu_{t,x}} & \frac{\partial \phi}{\partial \mu_{t,y}} & \frac{\partial \phi}{\partial \mu_{t,\theta}} \end{bmatrix}.$$

—

Derivation of Jacobian Components

Range (r)

The range r is given by:

$$r = \sqrt{(m_{j,x} - \mu_{t,x})^2 + (m_{j,y} - \mu_{t,y})^2}.$$

The partial derivatives of r are:

$$\frac{\partial r}{\partial \mu_{t,x}} = -\frac{m_{j,x} - \mu_{t,x}}{\sqrt{(m_{j,x} - \mu_{t,x})^2 + (m_{j,y} - \mu_{t,y})^2}},$$

$$\frac{\partial r}{\partial \mu_{t,y}} = -\frac{m_{j,y} - \mu_{t,y}}{\sqrt{(m_{j,x} - \mu_{t,x})^2 + (m_{j,y} - \mu_{t,y})^2}},$$

$$\frac{\partial r}{\partial \mu_{t,\theta}} = 0.$$

—

Bearing (ϕ)

The bearing ϕ is given by:

$$\phi = \arctan 2(m_{j,y} - \mu_{t,y}, m_{j,x} - \mu_{t,x}) - \mu_{t,\theta}.$$

The partial derivatives of ϕ are:

$$\frac{\partial \phi}{\partial \mu_{t,x}} = \frac{m_{j,y} - \mu_{t,y}}{(m_{j,x} - \mu_{t,x})^2 + (m_{j,y} - \mu_{t,y})^2},$$

$$\frac{\partial \phi}{\partial \mu_{t,y}} = -\frac{m_{j,x} - \mu_{t,x}}{(m_{j,x} - \mu_{t,x})^2 + (m_{j,y} - \mu_{t,y})^2},$$

$$\frac{\partial \phi}{\partial \mu_{t,\theta}} = -1.$$

—

Jacobian Matrix

Substituting the partial derivatives into the Jacobian matrix H_t, we get:

$$H_t = \begin{bmatrix} \frac{\partial r}{\partial \mu_{t,x}} & \frac{\partial r}{\partial \mu_{t,y}} & \frac{\partial r}{\partial \mu_{t,\theta}} \\ \frac{\partial \phi}{\partial \mu_{t,x}} & \frac{\partial \phi}{\partial \mu_{t,y}} & \frac{\partial \phi}{\partial \mu_{t,\theta}} \end{bmatrix} = \begin{bmatrix} -\frac{m_{j,x}-\mu_{t,x}}{r} & -\frac{m_{j,y}-\mu_{t,y}}{r} & 0 \\ \frac{m_{j,y}-\mu_{t,y}}{r^2} & -\frac{m_{j,x}-\mu_{t,x}}{r^2} & -1 \end{bmatrix},$$

where:

$$r = \sqrt{(m_{j,x} - \mu_{t,x})^2 + (m_{j,y} - \mu_{t,y})^2}.$$

Reference

- S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, 2006. Page 203 to 210.
- Canvas Course materials on Extended Kalman Filters.