Extended Kalman Filter SLAM

July 18, 2025

Introduction - Recursive Bayes Filters and Probability Model

- Simultaneous Localization and Mapping (SLAM): Robot constructs a map while determining its location.
- Recursive Bayes filter estimates belief $bel(x_t)$ over state x_t given controls $u_{1:t}$ and observations $z_{1:t}$.
- Belief update:

$$bel(x_t) = \eta \, p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) \int p(x_t \mid x_{t-1}, u_t) \, bel(x_{t-1}) \, dx_{t-1}$$

- Terms:
  - η: Normalization constant.
  - $p(z_t \mid x_t, z_{1:t-1}, u_{1:t})$: Measurement model.
  - $p(x_t \mid x_{t-1}, u_t)$: Motion model.
  - $bel(x_{t-1})$: Prior belief.
- For SLAM, x_t includes robot pose and landmarks; simplifies to Markov model:

Kalman Filter

$$bel(x_t) = \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t)\, bel(x_{t-1})\, dx_{t-1}$$

- Estimates state $\boldsymbol{x}_t$ of a linear dynamical system with Gaussian noise.
- Recursive update: Prediction and correction (measurement update).
- **Motion model:**

$$\boldsymbol{x}_t = A_{t-1}\boldsymbol{x}_{t-1} + B_{t-1}\boldsymbol{u}_{t-1} + \boldsymbol{w}_{t-1}, \quad \boldsymbol{w}_{t-1} \sim \mathcal{N}(0, R_{t-1})$$

- **Measurement model:**

$$\boldsymbol{z}_t = C_t\boldsymbol{x}_t + \boldsymbol{v}_t, \quad \boldsymbol{v}_t \sim \mathcal{N}(0, Q_t)$$

Kalman Filter - Prediction Step

- Prior (predicted) belief:

$$\hat{\boldsymbol{x}}_t = A_{t-1}\hat{\boldsymbol{x}}_{t-1} + B_{t-1}\boldsymbol{u}_{t-1} \tag{1}$$

$$P_t^- = A_{t-1}P_{t-1}A_{t-1}^\top + R_{t-1} \tag{2}$$

Kalman Filter - Measurement Update (Correction)

- Compute Kalman gain:

$$K_t = P_t^- C_t^\top (C_t P_t^- C_t^\top + Q_t)^{-1}$$

- Corrected belief:

$$\hat{x}_t \leftarrow \hat{x}_t^- + K_t(z_t - C_t \hat{x}_t^-) \tag{3}$$

$$P_t \leftarrow (I - K_t C_t) P_t^- \tag{4}$$

Extended Kalman Filter (EKF)

- Handles nonlinear systems by linearizing models with first-order Taylor expansion.

$$g(x) \approx g(\hat{x}) + G(x - \hat{x}) \tag{5}$$
$$h(x) \approx h(\hat{x}) + H(x - \hat{x}) \tag{6}$$

where $G = \frac{\partial g}{\partial x}\Big|_{\hat{x}}$, $H = \frac{\partial h}{\partial x}\Big|_{\hat{x}}$.

EKF - Non-linear Models

- **Motion model:**

$$\boldsymbol{x}_t = g(\boldsymbol{x}_{t-1}, \boldsymbol{u}_{t-1}) + \boldsymbol{w}_{t-1}, \quad \boldsymbol{w}_{t-1} \sim \mathcal{N}(0, R_t)$$

- **Measurement model:**

$$\boldsymbol{z}_t = h(\boldsymbol{x}_t) + \boldsymbol{v}_t, \quad \boldsymbol{v}_t \sim \mathcal{N}(0, Q_t)$$

EKF - State Vector and Covariance

- State vector μ_t includes robot pose and landmarks:

$$\mu_t = \begin{bmatrix} \mu_{t,x} \\ \mu_{t,y} \\ \mu_{t,\theta} \\ \mu_{j,x} \\ \mu_{j,y} \\ \vdots \end{bmatrix}, \quad \Sigma_t = \mathrm{Cov}[\mu_t]$$

EKF - Motion Model and Jacobian G_t

- Motion model for differential drive robot ($u_t = (v, \omega)$):

$$g(\mu_{t-1}, u_t) = \begin{bmatrix} \mu_{t-1,x} - \frac{v}{\omega}\sin(\mu_{t-1,\theta}) + \frac{v}{\omega}\sin(\mu_{t-1,\theta} + \omega\Delta t) \\ \mu_{t-1,y} + \frac{v}{\omega}\cos(\mu_{t-1,\theta}) - \frac{v}{\omega}\cos(\mu_{t-1,\theta} + \omega\Delta t) \\ \mu_{t-1,\theta} + \omega\Delta t \end{bmatrix}$$

- Jacobian G_t:

$$G_t = I + F_x^\top G_t^x F_x$$

where

$$G_t^x = \begin{bmatrix} 1 & 0 & -\frac{v}{\omega}\cos(\mu_{t-1,\theta}) + \frac{v}{\omega}\cos(\mu_{t-1,\theta} + \omega\Delta t) \\ 0 & 1 & -\frac{v}{\omega}\sin(\mu_{t-1,\theta}) + \frac{v}{\omega}\sin(\mu_{t-1,\theta} + \omega\Delta t) \\ 0 & 0 & 1 \end{bmatrix}$$

EKF - Process Noise Covariance R_t

- Process noise covariance:

$$R_t = F_x^\top Q_t F_x$$

where

$$Q_t = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & \sigma_\theta \end{bmatrix}$$

EKF - Measurement Model and Jacobian H_t^i

- Expected measurement for landmark j:

$$h(\mu_t) = \begin{bmatrix} \sqrt{(\mu_{j,x} - \mu_{t,x})^2 + (\mu_{j,y} - \mu_{t,y})^2} \\ \arctan 2(\mu_{j,y} - \mu_{t,y}, \mu_{j,x} - \mu_{t,x}) - \mu_{t,\theta} \end{bmatrix}$$

- Define $\delta = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix}$, $q = \delta_x^2 + \delta_y^2$. Jacobian:

$$H_t^i = \frac{1}{\sqrt{q}} \begin{bmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{bmatrix} F_{x,j}$$

EKF - Kalman Gain K_t^i

- Kalman gain:

$$K_t^i = \Sigma_t (H_t^i)^\top \left[(H_t^i \Sigma_t (H_t^i)^\top + Q_t) \right]^{-1}$$

EKF SLAM Algorithm - Part 1

- **EKF_SLAM(known_correspondences($z_{1:t}, z_{1:t}, u_{1:t}, z_{1:t}$)):**

1. $$F_x = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \end{bmatrix}$$
2. $$\mu_t = \mu_{t-1} + F_x^\top \begin{bmatrix} -\frac{v}{\omega}\sin(\mu_{t-1,\theta}) + \frac{v}{\omega}\sin(\mu_{t-1,\theta} + \omega\Delta t) \\ \frac{v}{\omega}\cos(\mu_{t-1,\theta}) - \frac{v}{\omega}\cos(\mu_{t-1,\theta} + \omega\Delta t) \\ \omega\Delta t \end{bmatrix}$$
3. $$\mu_t = \mu_{t-1} + F_x^\top \left(\begin{bmatrix} -\frac{v}{\omega}\sin(\mu_{t-1,\theta}) + \frac{v}{\omega}\sin(\mu_{t-1,\theta} + \omega\Delta t) \\ \frac{v}{\omega}\cos(\mu_{t-1,\theta}) - \frac{v}{\omega}\cos(\mu_{t-1,\theta} + \omega\Delta t) \\ \omega\Delta t \end{bmatrix} \right)$$
4. $$G_t = I + F_x^\top \begin{bmatrix} 0 & 0 & \frac{v}{\omega}\cos(\mu_{t-1,\theta}) - \frac{v}{\omega}\cos(\mu_{t-1,\theta} + \omega\Delta t) \\ 0 & 0 & \frac{v}{\omega}\sin(\mu_{t-1,\theta}) - \frac{v}{\omega}\sin(\mu_{t-1,\theta} + \omega\Delta t) \\ 0 & 0 & 0 \end{bmatrix} F_x$$
5. $$\Sigma_t = G_t \Sigma_{t-1} G_t^\top + F_x^\top \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & \sigma_\theta \end{bmatrix} F_x$$

EKF SLAM Algorithm - Part 2

7. **for all observed features** $z = (r^i, \phi^i, j)^\top$ **do**
8. **if landmark** j **never seen before**
$$\begin{bmatrix} \mu_{j,x} \\ \mu_{j,y} \end{bmatrix} = \begin{bmatrix} \mu_{t,x} \\ \mu_{t,y} \end{bmatrix} + \begin{bmatrix} r^i \cos(\phi^i + \mu_{t,\theta}) \\ r^i \sin(\phi^i + \mu_{t,\theta}) \end{bmatrix}$$
9. **endif**
10. $\delta_x = \mu_{j,x} - \mu_{t,x}, \quad \delta_y = \mu_{j,y} - \mu_{t,y}$
11. $q = \delta_x^2 + \delta_y^2$
12. $\hat{z}^i = \begin{bmatrix} \sqrt{q} \\ \operatorname{arctan2}(\delta_y, \delta_x) - \mu_{t,\theta} \end{bmatrix}$
13. $H_t^i = \frac{1}{\sqrt{q}} \begin{bmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{bmatrix} F_{x,j}$
14. $K_t^i = \Sigma_t (H_t^i)^\top \left[H_t^i \Sigma_t (H_t^i)^\top + \begin{bmatrix} \sigma_r & 0 \\ 0 & \sigma_\phi \end{bmatrix} \right]^{-1}$
15. $\mu_t = \mu_t + K_t^i (z^i - \hat{z}^i)$
16. $\Sigma_t = (I - K_t^i H_t^i) \Sigma_t$
17. **endfor**

Conclusion

- **Benefits of EKF**:
  - Handles nonlinear models via linearization, enabling real-time estimation.
  - Efficiently fuses noisy sensor data with motion predictions.
  - Reduces uncertainty and maintains consistent maps.
  - Recursive updates suit resource-constrained systems.
- **Uses of EKF**:
  - Indoor robot navigation with odometry and range sensors.
  - Autonomous vehicle localization with GPS and IMUs.
  - Underwater robotics mapping with sonar.
- Performance depends on accurate linearization and initial conditions.