Recommended reading

- Sec 4.9.1 in Dudek and Jenkin
- Chapter 3.3 in Probabilistic Robotics for EKF
- Chapter 7.4 in Probabilistic Robotics for EKF-Localization
- Chapter 10.2 in Probabilistic Robotics for EKF-SLAM
Kalman Filter: an instance of Bayes’ Filter

Linear observations with Gaussian noise

Kalman Filter:
- Linear dynamics with Gaussian noise
- Initial belief is Gaussian
- Assumptions guarantee that if the prior belief before the prediction step is Gaussian, and the posterior belief (after the update step) will be Gaussian.

How do we linearize?

- Using the first order Taylor expansion around the mean of the previous update step’s state estimate:

      \[ \begin{align*}
      x_{t+1} &= f(x_t, u_t) + w_t \\
      &\approx f(\mu_t, \nu_t) + \frac{\partial f}{\partial x}(\mu_t, \nu_t)(x_t - \mu_t) + w_t \\
      &= F_t x_t + (f(\mu_t, \nu_t) - F_t \mu_t) + w_t \\
      &= F_t x_t + \hat{u}_t + w_t
      \end{align*} \]

Recall how to compute the Jacobian matrix. For example, if

      \[ f(x_1, x_2, u) = [x_1 + x_2^2, x_2 + 3u, x_1^2 - u^2] \in \mathbb{R}^3 \]

then the Jacobian of \( f \) with respect to \((x_1, x_2)\) at \((\mu_1, \mu_2)\) is

      \[ \frac{\partial f}{\partial x_{1:2}}(\mu_1, \mu_2) = \begin{bmatrix}
      1 & 2 \mu_2 \\
      0 & 1 \\
      \end{bmatrix} \]

Extended Kalman Filter: an instance of Bayes’ Filter

How do we linearize?

- Using the first order Taylor expansion around the mean of the previous prediction step’s state estimate:

      \[ \begin{align*}
      x_{t+1} &= h(x_{t+1}) + n_{t+1} \\
      &\approx h(\mu_{t+1}) + \frac{\partial h}{\partial x}(\mu_{t+1})(x_{t+1} - \mu_{t+1}) + n_{t+1} \\
      &= h(\mu_{t+1}) + H_{t+1} x_{t+1} - H_{t+1} \mu_{t+1} + n_{t+1} \\
      &= H_{t+1} x_{t+1} + \hat{v}_{t+1} + n_{t+1}
      \end{align*} \]

Recall how to compute the Jacobian matrix. For example, if

      \[ h(\mu_1, \mu_2) = [\mu_1 + x_1^2, \mu_2 + 3u, \mu_1^2 - u^2] \in \mathbb{R}^3 \]

then the Jacobian of \( h \) with respect to \((x_1, x_2)\) at \((\mu_1, \mu_2)\) is

      \[ \frac{\partial h}{\partial x}(\mu_1, \mu_2) = \begin{bmatrix}
      2 \mu_1 & 2 \mu_2 \\
      0 & 1 \\
      \end{bmatrix} \]
EKF in N dimensions

Init
\[ \mathbf{w}(x_i) \sim \mathcal{N}(\mathbf{m}_i, \Sigma_i) \]

Prediction Step
\[
\begin{align*}
\mathbf{u}_{i+1} &= f(\mathbf{x}_i, \mathbf{u}_i) \\
\Sigma_{i+1|i} &= P_i \Sigma_{i|i} P_i^T + \mathbf{Q} \mathbf{Q}^T
\end{align*}
\]

Update Step
Received measurement \( \mathbf{z}_{i+1} \) but expected to receive \( \mu_{i+1} \)
Prediction residual is a Gaussian random variable \( \delta \mathbf{z} \sim \mathcal{N}(\delta \mathbf{x}_{i+1} - \mu_{i+1}, \Sigma_{i+1}) \)
where the covariance of the residual is
\[
\Sigma_{i+1} = H_i \Sigma_{i|i} H_i^T + R
\]
Kalman Gain (optimal correction factor):
\[
K_{i+1} = \Sigma_{i+1} \Sigma_{i+1|i}^{-1}
\]
\[
\begin{align*}
\mu_{i+1|i+1} &= \mu_{i+1|i} + K_{i+1}(\delta \mathbf{z}_{i+1} - \mu_{i+1}) \\
\Sigma_{i+1|i+1} &= \Sigma_{i+1|i} - K_{i+1} H_i \Sigma_{i+1|i}
\end{align*}
\]

EKF Summary

- **Efficient**: Polynomial in measurement dimensionality \( k \) and state dimensionality \( n \): \( O(k^{2.376} + n^2) \)
- **Not optimal** (unlike the Kalman Filter for linear systems)
- **Can diverge** if nonlinearities are large
- **Works surprisingly well** even when all assumptions are violated

Example #1: EKF-Localization

“Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities.” [Cox ’91]

- **Given**
  - Map of the environment.
  - Sequence of sensor measurements.
- **Wanted**
  - Estimate of the robot’s position.
- **Problem classes**
  - Position tracking
  - Global localization
  - Kidnapped robot problem (recovery)

Landmark-based Localization

Landmarks, whose position \([p_x, p_y]\) in the world is known.

Each robot measures its range and bearing from each landmark to localize itself.

State of a robot:
\[
x_t = \begin{bmatrix}
p_x(t) \\
p_y(t) \\
\theta(t)
\end{bmatrix}
\]
Landmark-based Localization

Measurement at time $t$, $z_t = \begin{bmatrix} z_1(t) \\ \vdots \\ z_m(t) \end{bmatrix}$ is a variable-sized vector, depending on the landmarks that are visible at time $t$.

Each measurement is a 2D vector, containing range and bearing from the robot to a landmark.

$z_t^{(i)} = h_t(x_t) = \begin{bmatrix} r(t) \\ \theta(t) \end{bmatrix}$

Estimation Sequence (1)

Estimation Sequence (2)

Comparison to true trajectory
Example #2: EKF-SLAM

A robot is exploring an unknown, static environment.

Given:
• The robot’s controls
• Observations of nearby features

Estimate:
• Map of features
• Path of the robot

Why is SLAM a hard problem?

SLAM: robot path and map are both unknown

Robot path error correlates errors in the map

Why is SLAM a hard problem?

• In the real world, the mapping between observations and landmarks is unknown
• Picking wrong data associations can have catastrophic consequences
• Pose error correlates data associations
EKF-SLAM: the map is part of the state!

• Map with N landmarks: $(3+2N)$-dimensional Gaussian

$$\text{Bel}(x, m_i) = \left( \begin{array}{cccc}
\sigma^2_1 & \sigma_{12} & \cdots & \sigma_{1N} \\
\sigma_{12} & \sigma^2_2 & \cdots & \sigma_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{1N} & \sigma_{2N} & \cdots & \sigma^2_N 
\end{array} \right)$$

• Can handle hundreds of dimensions
Properties of EKF-SLAM (Linear Case) [Dissanayake et al., 2001]

**Theorem:**
The determinant of any sub-matrix of the map covariance matrix decreases monotonically as successive observations are made.

**Theorem:**
In the limit the landmark estimates become fully correlated