Introduction to Mobile Robotics

Bayes Filter – Extended Kalman Filter

Daniel Büscher
Bayes Filter Reminder

\[ \text{bel}(x_t) = \eta \ p(z_t | x_t) \int p(x_t | u_t, x_{t-1}) \text{bel}(x_{t-1}) \, dx_{t-1} \]

- **Prediction**

\[ \overline{\text{bel}}(x_t) = \int p(x_t | u_t, x_{t-1}) \text{bel}(x_{t-1}) \, dx_{t-1} \]

- **Correction**

\[ \text{bel}(x_t) = \eta \ p(z_t | x_t) \overline{\text{bel}}(x_t) \]
Discrete Kalman Filter

Estimates the state $x$ of a discrete-time controlled process

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

with a measurement

$$z_t = C_t x_t + \delta_t$$
Nonlinear Dynamic Systems

- Most realistic robotic problems involve nonlinear functions

\[ x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \quad \Rightarrow \quad x_t = g(u_t, x_{t-1}) \]

\[ z_t = C_t x_t + \delta_t \quad \Rightarrow \quad z_t = h(x_t) \]
Linear Fn: Gaussian preserved
Non-Linear Function

Non-Gaussian!
Non-Gaussian Distributions

- Non-linear functions lead to non-Gaussian distributions
- The Kalman filter is not applicable anymore!

What can be done to resolve this?

Local linearization!
1. Linearization
EKF Linearization: First Order Taylor Expansion

- Prediction:
  \[ g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} (x_{t-1} - \mu_{t-1}) \]
  \[ = g(u_t, \mu_{t-1}) + G_t (x_{t-1} - \mu_{t-1}) \]

- Correction:
  \[ h(x_t) \approx h(\bar{\mu}_t) + \frac{\partial h(\bar{\mu}_t)}{\partial x_t} (x_t - \bar{\mu}_t) \]
  \[ = h(\bar{\mu}_t) + H_t (x_t - \bar{\mu}_t) \]

Jacobian matrices: linear functions!
Jacobian Matrix

- Non-square \( n \times m \) in general
- Given a vector-valued function

\[
f(x) = \begin{bmatrix}
f_1(x) \\
f_2(x) \\
\vdots \\
f_m(x)
\end{bmatrix}
\]

It is defined as

\[
F_x = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots & \frac{\partial f_1}{\partial x_n} \\
\frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots & \frac{\partial f_2}{\partial x_n} \\
\vdots & \vdots & \ldots & \vdots \\
\frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \ldots & \frac{\partial f_m}{\partial x_n}
\end{bmatrix}
\]
**Jacobian Matrix interpretation**

- It is the orientation of the tangent plane to the vector-valued function at a given point

- Generalizes the gradient of a scalar-valued function
Non-Linear Function
EKF Linearization (2)
EKF Linearization (3)
2. EKF Algorithm
Reminder: KF Algorithm

1. Algorithm **Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

   Prediction:
   2. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
   3. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q$

   Correction:
   4. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}$
   5. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \mu_t)$
   6. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

7. Return $\mu_t, \Sigma_t$
EKF Algorithm

1. Extended_Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

   Prediction:
   
   2. $\overline{\mu}_t = g(u_t, \mu_{t-1})$
   3. $\overline{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + Q_t$

   Correction:

   4. $K_t = \overline{\Sigma}_t H_t^T (H_t \overline{\Sigma}_t H_t^T + R_t)^{-1}$
   5. $\mu_t = \overline{\mu}_t + K_t (z_t - h(\overline{\mu}_t))$
   6. $\Sigma_t = (I - K_t H_t) \overline{\Sigma}_t$

   7. Return $\mu_t, \Sigma_t$

Kalman_filter:

   $\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t$
   $\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q$

   $K_t = \overline{\Sigma}_t C_t (C_t \overline{\Sigma}_t C_t^T + R_t)^{-1}$
   $\mu_t = \mu_t + K_t (z_t - C_t \mu_t)$
   $\Sigma_t = (I - K_t C_t) \overline{\Sigma}_t$

\[
G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} \quad H_t = \frac{\partial h(\overline{\mu}_t)}{\partial x_t}
\]
3. EKF Example
Example: EKF Localization

- EKF localization with landmarks (point features)
1. **EKF_localization** \( (\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m) \):

Prediction:

2. \( G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial \mu_{t-1}} = \begin{pmatrix} \frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial \theta'}{\partial \mu_{t-1,x}} & \frac{\partial \theta'}{\partial \mu_{t-1,y}} & \frac{\partial \theta'}{\partial \mu_{t-1,\theta}} \end{pmatrix} \) \quad \text{Jacobian of } g \text{ w.r.t location}

3. \( V_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t} = \begin{pmatrix} \frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial \omega_t} \\ \frac{\partial y'}{\partial v_t} & \frac{\partial y'}{\partial \omega_t} \\ \frac{\partial \theta'}{\partial v_t} & \frac{\partial \theta'}{\partial \omega_t} \end{pmatrix} \) \quad \text{Jacobian of } g \text{ w.r.t control}

4. \( Q_t = \begin{pmatrix} (\alpha_1 |v_t| + \alpha_2 |\omega_t|)^2 & 0 \\ 0 & (\alpha_3 |v_t| + \alpha_4 |\omega_t|)^2 \end{pmatrix} \) \quad \text{Motion noise}

5. \( \mu_t = g(u_t, \mu_{t-1}) \) \quad \text{Predicted mean}

6. \( \Sigma_t = G_t \Sigma_{t-1} G_t^T + V_t Q_t V_t^T \) \quad \text{Predicted covariance (}V\text{ maps } Q\text{ into state space)}
EKF Prediction Step

Small Q

Medium Q

Medium Q

Large Q
Correction:  

(EKF_localization continued)

7. \( \hat{z}_t = \begin{pmatrix} \sqrt{(m_x - \bar{\mu}_{t,x})^2 + (m_y - \bar{\mu}_{t,y})^2} \\ \text{atan} 2 \left( m_y - \bar{\mu}_{t,y}, m_x - \bar{\mu}_{t,x} - \bar{\mu}_{t,0} \right) \end{pmatrix} \) 

Predicted measurement mean \( h(\mu) = z = (r, \Phi)^T \)

8. \( \mathbf{H}_t = \frac{\partial h(\bar{\mu}_t, \mathbf{m})}{\partial \mathbf{x}_t} = \begin{pmatrix} \frac{\partial r_i}{\partial \mu_{t,x}} & \frac{\partial r_i}{\partial \mu_{t,y}} & \frac{\partial r_i}{\partial \mu_{t,0}} \\ \frac{\partial \phi_i}{\partial \mu_{t,x}} & \frac{\partial \phi_i}{\partial \mu_{t,y}} & \frac{\partial \phi_i}{\partial \mu_{t,0}} \end{pmatrix} \) 

Jacobian of \( h \) w.r.t location

9. \( \mathbf{R}_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix} \) 

Measurement noise

10. \( \mathbf{S}_t = \mathbf{H}_t \Sigma_t \mathbf{H}_t^T + \mathbf{R}_t \) 

Innovation covariance

11. \( \mathbf{K}_t = \Sigma_t \mathbf{H}_t^T \mathbf{S}_t^{-1} \) 

Kalman gain

12. \( \mu_t = \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - \hat{\mathbf{z}}_t) \) 

Updated mean

13. \( \Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \Sigma_t \) 

Updated covariance
EKF Observation Prediction

Example 1

Example 2
EKF Correction Step

Example 1

Example 2
Localization Sequence

- Odometry only
- EKF Corrected
- True motion
The Extended Kalman Filter is an ad-hoc solution to deal with non-linearities. It performs local linearization in each step and works well in practice for moderate non-linearities (example: landmark localization). It is optimal if the measurement and the motion model are both linear (reduces to the KF). There exist better ways for dealing with non-linearities, such as the unscented Kalman filter.