Introduction to Mobile Robotics

Bayes Filter – Extended Kalman Filter

Wolfram Burgard, Cyrill Stachniss, Maren Bennewitz, Kai Arras
Bayes Filter Reminder

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

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

- **Correction**
  \[ \text{bel}(x_t) = \eta \ p(z_t \mid x_t) \overline{\text{bel}}(x_t) \]
Discrete Kalman Filter

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

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

with a measurement

$$z_t = C_t x_t + \delta_t$$
Components of a Kalman Filter

- $A_t$: Matrix (nxn) that describes how the state evolves from $t-1$ to $t$ without controls or noise.

- $B_t$: Matrix (nxl) that describes how the control $u_t$ changes the state from $t-1$ to $t$.

- $C_t$: Matrix (kxn) that describes how to map the state $x_t$ to an observation $z_t$.

- $\varepsilon_t$: Random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance $Q_t$ and $R_t$ respectively.
Kalman Filter Update Example

prediction

measurement

It's a weighted mean!
Kalman Filter Update Example

prediction

measurement

correction
Kalman Filter Algorithm

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

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

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

9. Return $\mu_t$, $\Sigma_t$
Nonlinear Dynamic Systems

- Most realistic robotic problems involve nonlinear functions

\[ x_t = A_t x_{t-1} + B_t u_t + \epsilon_t \]

\[ z_t = C_t x_t + \delta_t \]

\[ x_t = g(u_t, x_{t-1}) \]

\[ z_t = h(x_t) \]
Linearity Assumption Revisited

$p(y) = N(y; \mu_y + b, \sigma^2 \sigma_y^2)$

Mean of $p(y)$

$y = a \times + b$

Mean $\mu$

$p(x) = N(x; \mu, \sigma^2)$

Mean of $p(x)$
Non-Linear Function

Non-Gaussian!
Non-Gaussian Distributions

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

What can be done to resolve this?
Non-Gaussian Distributions

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

What can be done to resolve this?

Local linearization!
EKF Linearization: First Order Taylor Expansion

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

- **Correction:**
  \[ h(x_t) \approx h(\mu_t) + \frac{\partial h(\mu_t)}{\partial x_t} (x_t - \mu_t) \]
  \[ h(x_t) \approx h(\mu_t) + H_t (x_t - \mu_t) \]
Reminder: Jacobian Matrix

- It is a **non-square matrix** $n \times m$ in general

- Given a vector-valued function

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

- The **Jacobian matrix** is defined as

  $$F_x = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$
Reminder: Jacobian Matrix

- 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.
EKF Linearization: First Order Taylor Expansion

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

- **Correction:**
  \[
h(x_t) \approx h(\bar{\mu}_t) + \frac{\partial h(\bar{\mu}_t)}{\partial x_t} (x_t - \bar{\mu}_t)
  \]
  \[
h(x_t) \approx h(\bar{\mu}_t) + H_t (x_t - \bar{\mu}_t)
  \]
  Linear function!
Linearity Assumption Revisited
Non-Linear Function
EKF Linearization (1)
EKF Linearization (2)
EKF Linearization (3)
EKF Algorithm

1. **Extended_Kalman_filter**\((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\):
   
   2. **Prediction**:
      
      3. \(\bar{\mu}_t = g(u_t, \mu_{t-1})\) \[\bar{\mu}_t = A_t\mu_{t-1} + B_tu_t\]
      
      4. \(\bar{\Sigma}_t = G_t\Sigma_{t-1}G_t^T + Q_t\) \[\bar{\Sigma}_t = A_t\Sigma_{t-1}A_t^T + Q_t\]
      
   5. **Correction**:
      
      6. \(K_t = \bar{\Sigma}_tH_t^T (H_t\bar{\Sigma}_tH_t^T + R_t)^{-1}\) \[K_t = \bar{\Sigma}_tC_t^T (C_t\bar{\Sigma}_tC_t^T + R_t)^{-1}\]
      
      7. \(\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))\) \[\mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t)\]
      
      8. \(\Sigma_t = (I - K_tC_t)\bar{\Sigma}_t\) \[\Sigma_t = (I - K_tC_t)\bar{\Sigma}_t\]
      
   9. **Return** \(\mu_t, \Sigma_t\)

\[
H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \quad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}
\]
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. \(G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}\) = \begin{bmatrix}
\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{bmatrix}\) Jacobian of \(g\) w.r.t location

3. \(V_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t}\) = \begin{bmatrix}
\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{bmatrix}\) Jacobian of \(g\) w.r.t control

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

5. \(\bar{\mu}_t = g(u_t, \mu_{t-1})\) Predicted mean

6. \(\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t Q_t V_t^T\) Predicted covariance (\(V\) maps \(Q\) into state space)
1. **EKF_localization** \( (\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m) \):

   **Correction:**

   2. \[ \hat{z}_t = \left( \sqrt{(m_x - \bar{\mu}_{t,x})^2 + (m_y - \bar{\mu}_{t,y})^2} \right) \text{ atan} 2 \left( m_y - \bar{\mu}_{t,y}, m_x - \bar{\mu}_{t,x} \right) \] Predicted measurement mean (depends on observation type)

3. \[ H_t = \frac{\partial h(\bar{\mu}_t,m)}{\partial x} = \left( \begin{array}{ccc} \frac{\partial r_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,y}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,\theta}} \\ \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,y}} & \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,\theta}} \end{array} \right) \] Jacobian of \( h \) w.r.t location

4. \[ R_t = \begin{pmatrix} \sigma^2_r & 0 \\ 0 & \sigma^2_r \end{pmatrix} \]

5. \[ S_t = H_t \bar{\Sigma}_t H_t^T + R_t \] Innovation covariance

6. \[ K_t = \bar{\Sigma}_t H_t^T S_t^{-1} \] Kalman gain

7. \[ \mu_t = \bar{\mu}_t + K_t (z_t - \hat{z}_t) \] Updated mean

8. \[ \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \] Updated covariance
EKF Prediction Step Examples
EKF Observation Prediction Step
EKF Correction Step
Estimation Sequence (1)
Estimation Sequence (2)
Comparison to GroundTruth
Extended Kalman Filter

Summary

- Ad-hoc solution to deal with non-linearities
- Performs local linearizations in each step
- Works well in practice for moderate non-linearities
- Example: landmark localization
- There exists better ways for dealing with non-linearities such as the unscented Kalman filter called UKF