Introduction to Mobile Robotics

Bayes Filter – Kalman Filter

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

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

1. Algorithm **Bayes_filter** (Bel(x), d):
2. \( \eta = 0 \)
3. If \( d \) is a perceptual data item \( z \) then
4. For all \( x \) do
5. \( Bel'(x) = P(z \mid x) Bel(x) \)
6. \( \eta = \eta + Bel'(x) \)
7. For all \( x \) do
8. \( Bel'(x) = \eta^{-1} Bel'(x) \)
9. Else if \( d \) is an action data item \( u \) then
10. For all \( x \) do
11. \( Bel'(x) = \int P(x \mid u, x') Bel(x') \, dx' \)
12. Return \( Bel'(x) \)
Kalman Filter

- Bayes filter with **Gaussians**
- Developed in the late 1950's
- Most relevant Bayes filter variant in practice
- Applications range from economics, weather forecasting, satellite navigation to robotics and many more.

- The Kalman filter "algorithm" is a couple of **matrix multiplications**!
Gaussians

Univariate

\[ p(x) \sim N(\mu, \sigma^2): \]
\[ p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}} \]

Multivariate

\[ p(x) \sim N(\mu, \Sigma): \]
\[ p(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2} (x-\mu)' \Sigma^{-1} (x-\mu)} \]
Gaussians

1D

2D

\[ C = \begin{bmatrix} 0.020 & 0.013 \\ 0.013 & 0.020 \end{bmatrix} \]

\[ \lambda_1 = 0.007 \]

\[ \lambda_2 = 0.033 \]

\[ \rho = \sigma_{XY} / \sigma_X \sigma_Y = 0.673 \]
Properties of Gaussians

- Univariate case

\[
\begin{align*}
X & \sim N(\mu, \sigma^2) \\
Y & = aX + b \\
\end{align*} \implies Y & \sim N(a\mu + b, a^2\sigma^2)
\]

\[
\begin{align*}
X_1 & \sim N(\mu_1, \sigma_1^2) \\
X_2 & \sim N(\mu_2, \sigma_2^2) \\
\end{align*} \implies p(X_1) \cdot p(X_2) & \sim N\left(\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \mu_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \mu_2, \frac{1}{\sigma_1^{-2} + \sigma_2^{-2}} \right)
\]
Properties of Gaussians

- Multivariate case

\[ X \sim N(\mu, \Sigma) \]
\[ Y = AX + B \]
\[ \Rightarrow \quad Y \sim N(A\mu + B, A\Sigma A^T) \]

\[ X_1 \sim N(\mu_1, \Sigma_1) \]
\[ X_2 \sim N(\mu_2, \Sigma_2) \]
\[ \Rightarrow \quad p(X_1) \cdot p(X_2) \sim N\left( \frac{\Sigma_2}{\Sigma_1 + \Sigma_2} \mu_1 + \frac{\Sigma_1}{\Sigma_1 + \Sigma_2} \mu_2, \frac{1}{\Sigma_1^{-1} + \Sigma_2^{-1}} \right) \]

(where division "−" denotes matrix inversion)

- We **stay Gaussian** as long as we start with Gaussians and perform only **linear transformations**
Discrete Kalman Filter

Estimates the state $x$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

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

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 $t$ to $t-1$ without controls or noise.
- $B_t$: Matrix (nxl) that describes how the control $u_t$ changes the state from $t$ to $t-1$.
- $C_t$: Matrix (kxn) that describes how to map the state $x_t$ to an observation $z_t$.
- $\epsilon_t$: 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.
Bayes Filter Reminder

- **Prediction**

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

- **Correction**

\[ bel(x_t) = \eta \, p(z_t \mid x_t) \, \overline{bel}(x_t) \]
Kalman Filter Updates in 1D

prediction

measurement

correction

It's a weighted mean!
Kalman Filter Updates in 1D

How to get the blue one?
→ Kalman correction step

\[
\text{bel}(x_t) = \begin{cases}
\mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\
\sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2
\end{cases}
\quad \text{with} \quad K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \sigma_{obs,t}^2}
\]

\[
\text{bel}(x_t) = \begin{cases}
\mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t) \\
\Sigma_t = (I - K_tC_t)\bar{\Sigma}_t
\end{cases}
\quad \text{with} \quad K_t = \frac{\bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}}{\bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}}
\]
Kalman Filter Updates in 1D

\[ \overline{bel}(x_t) = \begin{cases} \overline{\mu}_t = a_t u_{t-1} + b_t u_t \\ \overline{\sigma}^2_t = a_t^2 \sigma_t^2 + \sigma^2_{act,t} \end{cases} \]

\[ \overline{bel}(x_t) = \begin{cases} \overline{\mu}_t = A_t x_{t-1} + B_t u_t \\ \overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t \end{cases} \]

How to get the magenta one?
→ State prediction step
Kalman Filter Updates
Linear Gaussian Systems: Initialization

- Initial belief is normally distributed:

\[ bel(x_0) = N(x_0; \mu_0, \Sigma_0) \]
Linear Gaussian Systems: Dynamics

- Dynamics are linear function of state and control plus additive noise:

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

\[ p(x_t \mid u_t, x_{t-1}) = N(x_t; A_t x_{t-1} + B_t u_t, Q_t) \]

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

\[ \downarrow \]

\[ \sim N(x_t; A_t x_{t-1} + B_t u_t, Q_t) \sim N(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}) \]
Linear Gaussian Systems: Dynamics

\[
\text{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) \quad \text{bel}(x_{t-1}) \, dx_{t-1}
\]
\[
\downarrow \quad \downarrow
\]
\[
\sim N(x_t; A_t x_{t-1} + B_t u_t, Q_t) \quad \sim N(x_{t-1}; u_{t-1}, \Sigma_{t-1})
\]
\[
\downarrow
\]
\[
\text{bel}(x_t) = \eta \int \exp \left\{ -\frac{1}{2} (x_t - A_t x_{t-1} - B_t u_t)^T Q_t^{-1} (x_t - A_t x_{t-1} - B_t u_t) \right\} \exp \left\{ -\frac{1}{2} (x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - \mu_{t-1}) \right\} \, dx_{t-1}
\]
\[
\text{bel}(x_t) = \begin{cases}
\bar{\mu}_t = A_t u_{t-1} + B_t u_t \\
\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t
\end{cases}
\]
Linear Gaussian Systems: Observations

- Observations are linear function of state plus additive noise:

\[ z_t = C_t x_t + \delta_t \]

\[ p(z_t \mid x_t) = N(z_t; C_t x_t, R_t) \]

\[ \text{bel}(x_t) = \eta \quad p(z_t \mid x_t) \quad \text{bel}(x_t) \]

\[ \sim N(z_t; C_t x_t, R_t) \quad \sim N(x_t; \mu_t, \Sigma_t) \]
Linear Gaussian Systems: Observations

\begin{align*}
\text{bel}(x_t) &= \eta \ p(z_t | x_t) \quad \text{bel}(x_t) \\
\quad & \downarrow \quad \downarrow \\
\sim N(z_t; C_t x_t, R_t) & \sim N(x_t; \mu_t, \Sigma_t) \\
\downarrow \\
\text{bel}(x_t) &= \eta \exp\left\{-\frac{1}{2} (z_t - C_t x_t)^T R_t^{-1} (z_t - C_t x_t) \right\} \exp\left\{-\frac{1}{2} (x_t - \mu_t)^T \Sigma_t^{-1} (x_t - \mu_t) \right\} \\
\text{bel}(x_t) &= \begin{cases} 
\mu_t = \bar{\mu}_t + K_t (z_t - C_t \mu_t) \\
\Sigma_t = (I - K_t C_t) \Sigma_t 
\end{cases} \quad \text{with} \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}
\end{align*}
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. \( \mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) \)
8. \( \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \)
9. Return \( \mu_t, \Sigma_t \)
Kalman Filter Algorithm
Kalman Filter Algorithm

- **Prediction**
  \[
  \hat{x}(k+1 | k) = f(\hat{x}(k | k), u(k + 1))
  \]
  \[
  P(k + 1 | k) = \nabla f_x P(k | k) \nabla f_x^T + \nabla f_u U(k + 1) \nabla f_u^T
  \]

- **Observation**
  \[
  z(k + 1), R(k + 1)
  \]

- **Matching**
  \[
  \tilde{v}_{i,j}(k + 1) = z_i(k + 1) - h(\hat{x}(k + 1 | k), m_j)
  \]

- **Correction**
  \[
  \hat{x}(k + 1 | k + 1) = \hat{x}(k + 1 | k) + W(k + 1) v(k + 1)
  \]
  \[
  P(k + 1 | k + 1) = P(k + 1 | k) - W(k + 1) S(k + 1) W^T(k + 1)
  \]
The Prediction-Correction-Cycle

\[
\text{bel}(x_t) = \begin{cases} 
\mu_t = a_t \mu_{t-1} + b_t u_t \\
\sigma_t^2 = a_t^2 \sigma_{t-1}^2 + \sigma_{\text{act},t}^2 
\end{cases}
\]

\[
\text{bel}(x_t) = \begin{cases} 
\mu_t = A_t \mu_{t-1} + B_t u_t \\
\Sigma_t = A_t \Sigma_{t-1} A_t^T + Q_t 
\end{cases}
\]
The Prediction-Correction-Cycle

\[ bel(x_t) = \begin{cases} 
\mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t), & K_t = \frac{\sigma_t^2}{\sigma_t^2 + \sigma_{\text{obs}}^2} \\
\sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2 
\end{cases} \]

\[ \Sigma_t = (I - K_t C_t) \Sigma_t \]

\[ K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1} \]
The Prediction-Correction-Cycle

\[
\text{bel}(x_t) = \begin{cases} 
\mu_t = \bar{\mu} + K_t(z_t - \bar{\mu}), \\
\sigma_t^2 = (1 - K_t)\sigma_t^2, \\
K_t = \frac{\sigma_t^2}{\sigma_t^2 + \sigma_{\text{obs}}^2} 
\end{cases}
\]

\[
\Sigma_t = (I - K_t C_t)\Sigma_t, \quad K_t = \Sigma_t C_t^T (C_t \Sigma_t C_t^T + R_t)^{-1}
\]

\[
\text{bel}(x_t) = \begin{cases} 
\bar{\mu}_t = a_t \mu_{t-1} + b_t u_t, \\
\sigma_t^2 = a_t^2 \sigma_t^2 + \sigma_{\text{act}}^2, 
\end{cases}
\]

\[
\Sigma_t = A_t \Sigma_{t-1} A_t^T + Q_t
\]
Kalman Filter Summary

- **Highly efficient**: Polynomial in the measurement dimensionality $k$ and state dimensionality $n$:

\[ O(k^{2.376} + n^2) \]

- **Optimal for linear Gaussian systems**!

- Most robotics systems are **nonlinear**!
Nonlinear Dynamic Systems

- Most realistic robotic problems involve nonlinear functions

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

\[ z_t = h(x_t) \]
EKF Linearization: First Order Taylor Series 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:

\[
h(x_t) \approx h(\bar{u}_t) + \frac{\partial h(\bar{u}_t)}{\partial x_t} (x_t - \bar{u}_t)
\]

\[
h(x_t) \approx h(\bar{u}_t) + H_t (x_t - \bar{u}_t)
\]
EKF Algorithm

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

2. **Prediction:**

3. \[ \bar{\mu}_t = g(u_t, \mu_{t-1}) \]

4. \[ \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + Q_t \]

5. **Correction:**

6. \[ K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1} \]

7. \[ \mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t)) \]

8. \[ \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \]

9. Return $\mu_t, \Sigma_t$

\[ H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \]

\[ G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} \]
Linearity Assumption Revisited
Non-linear Function
EKF Linearization (1)
EKF Linearization (2)
EKF Linearization (3)