CSE-P590a
Robotics

Kalman Filters

Dieter Fox
Bayes Filter Reminder

- **Prediction**

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

- **Correction**

\[
bel(x_t) = \eta \, p(z_t \mid x_t) \, \overline{bel}(x_t)
\]
Properties of Gaussians

\[ X \sim N(\mu, \sigma^2) \]
\[ Y = aX + b \]
\[ \implies Y \sim N(a\mu + b, a^2\sigma^2) \]
Properties of Gaussians

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

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

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

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

- Marginalization and conditioning in Gaussians results in Gaussians
- We stay in the “Gaussian world” 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 $t-1$ to $t$ 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 $z_t$.

- $\varepsilon_t$: Random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance $R_t$ and $Q_t$ respectively.
Kalman Filter Updates in 1D
Kalman Filter Updates in 1D

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

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

\[
\bar{\mu}_t = a_t \mu_{t-1} + b_t u_t
\]
\[
\bar{\sigma}_t^2 = a^2_t \sigma_{t-1}^2 + \sigma^2_{act,t}
\]

\[
\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t
\]
\[
\bar{\Sigma}_t = A_t \Sigma_{t-1} A^T_t + R_t
\]
Kalman Filter Updates
Kalman Filter Algorithm

1. Algorithm \textbf{Kalman\_filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t):

2. Prediction:
3. \quad \mu_t = A_t \mu_{t-1} + B_t u_t
4. \quad \Sigma_t = A_t \Sigma_{t-1} A_t^T + R_t

5. Correction:
6. \quad K_t = \Sigma_t C_t^T (C_t \Sigma_t C_t^T + Q_t)^{-1}
7. \quad \mu_t = \mu_t + K_t (z_t - C_t \mu_t)
8. \quad \Sigma_t = (I - K_t C_t) \Sigma_t
9. Return $\mu_t, \Sigma_t$
Kalman Filter Summary

- **Highly efficient**: Polynomial in measurement dimensionality $k$ and state dimensionality $n$:
  \[ O(k^{2.376} + n^2) \]

- **Optimal for linear Gaussian systems**!

- Most robotics systems are **nonlinear**!
Going non-linear

EXTENDED KALMAN FILTER
Linearity Assumption Revisited
Non-linear Function
EKF Linearization (1)
EKF Linearization (2)
EKF Linearization (3)
EKF Algorithm

1. \textbf{Extended Kalman filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)

2. Prediction:
   \[
   \bar{\mu}_t = g(u_t, \mu_{t-1}) \quad \quad \quad \quad \quad \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t
   \]
   \[
   \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t \quad \quad \quad \quad \quad \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t
   \]

3. Correction:
   \[
   K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \quad \quad \quad \quad \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}
   \]
   \[
   \mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t)) \quad \quad \quad \quad \quad \mu_t = \bar{\mu}_t + K_t (z_t - C_t \mu_t)
   \]
   \[
   \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \quad \quad \quad \quad \quad \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t
   \]
   \[
   H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \quad \quad \quad \quad \quad \quad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}
   \]

9. Return $\mu_t, \Sigma_t$
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
1. **EKF_localization** \((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\):

   **Prediction:**

   \[
   G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{pmatrix}
   \frac{\partial x'}{\partial x_{t-1}} & \frac{\partial x'}{\partial y_{t-1}} & \frac{\partial x'}{\partial \theta'_{t-1}} \\
   \frac{\partial y'}{\partial x_{t-1}} & \frac{\partial y'}{\partial y_{t-1}} & \frac{\partial y'}{\partial \theta'_{t-1}} \\
   \frac{\partial \theta'}{\partial x_{t-1}} & \frac{\partial \theta'}{\partial y_{t-1}} & \frac{\partial \theta'}{\partial \theta'_{t-1}} \\
   \frac{\partial \mu_{t-1,x}}{\partial x_{t-1}} & \frac{\partial \mu_{t-1,y}}{\partial x_{t-1}} & \frac{\partial \mu_{t-1,\theta}}{\partial x_{t-1}} \\
   \frac{\partial \mu_{t-1,x}}{\partial y_{t-1}} & \frac{\partial \mu_{t-1,y}}{\partial y_{t-1}} & \frac{\partial \mu_{t-1,\theta}}{\partial y_{t-1}} \\
   \frac{\partial \mu_{t-1,x}}{\partial \theta'_{t-1}} & \frac{\partial \mu_{t-1,y}}{\partial \theta'_{t-1}} & \frac{\partial \mu_{t-1,\theta}}{\partial \theta'_{t-1}} \\
   \end{pmatrix}
   \]

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

   \[
   V_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t} = \begin{pmatrix}
   \frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial \omega_t} \\
   \frac{\partial v'}{\partial v_t} & \frac{\partial v'}{\partial \omega_t} \\
   \frac{\partial v'}{\partial \theta'} & \frac{\partial v'}{\partial \theta'} \\
   \frac{\partial \mu_{t-1,x}}{\partial v_t} & \frac{\partial \mu_{t-1,y}}{\partial v_t} & \frac{\partial \mu_{t-1,\theta}}{\partial v_t} \\
   \frac{\partial \mu_{t-1,x}}{\partial \omega_t} & \frac{\partial \mu_{t-1,y}}{\partial \omega_t} & \frac{\partial \mu_{t-1,\theta}}{\partial \omega_t} \\
   \end{pmatrix}
   \]

   Jacobian of \(g\) w.r.t control

   \[
   M_t = \begin{pmatrix}
   \alpha_1 v_t^2 + \alpha_2 \omega_t^2 & 0 \\
   0 & \alpha_3 v_t^2 + \alpha_4 \omega_t^2 \\
   \end{pmatrix}
   \]

   Motion noise

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

   Predicted mean

   \[
   \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T
   \]

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

**Prediction:**

\[
\begin{align*}
\theta &= \mu_{t-1, \theta} \\
G_t &= \begin{pmatrix}
1 & 0 & -\frac{v_t}{\omega_t} \cos \theta + \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\
0 & 1 & -\frac{v_t}{\omega_t} \sin \theta + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\
0 & 0 & 1
\end{pmatrix} \\
V_t &= \begin{pmatrix}
\frac{-\sin \theta + \sin(\theta + \omega_t \Delta t)}{\omega_t} & \frac{v_t (\sin \theta - \sin(\theta + \omega_t \Delta t))}{\omega_t^2} + \frac{v_t \cos(\theta + \omega_t \Delta t) \Delta t}{\omega_t} \\
\frac{\cos \theta - \cos(\theta + \omega_t \Delta t)}{\omega_t} & \frac{-v_t (\cos \theta - \cos(\theta + \omega_t \Delta t))}{\omega_t^2} + \frac{v_t \sin(\theta + \omega_t \Delta t) \Delta t}{\omega_t} \\
0 & \Delta t
\end{pmatrix} \\
M_t &= \begin{pmatrix}
\alpha_1 v_t^2 + \alpha_2 \omega_t^2 & 0 \\
0 & \alpha_3 v_t^2 + \alpha_4 \omega_t^2
\end{pmatrix} \\
\bar{\mu}_t &= \mu_{t-1} + \begin{pmatrix}
-\frac{v_t}{\omega_t} \sin \theta + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\
\frac{v_t}{\omega_t} \cos \theta - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\
\omega_t \Delta t
\end{pmatrix}
\end{align*}
\]

6. $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T$  
   **Predicted covariance**
EKF Prediction Step
1. **EKF_localization** \((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\):

**Correction:**

3. \[ \hat{z}_t = \left( \frac{\sqrt{(m_x - \mu_{t,x})^2 + (m_y - \mu_{t,y})^2}}{\tan(2(m_y - \mu_{t,y}, m_x - \mu_{t,x}) - \mu_{t,\theta})} \right) \] \( \text{Predicted measurement mean} \)

5. \[ H_t = \frac{\partial h(\mu_t, m)}{\partial x_t} = \begin{pmatrix} \frac{\partial r_i}{\partial \mu_{t,x}} & \frac{\partial r_i}{\partial \mu_{t,y}} & \frac{\partial r_i}{\partial \mu_{t,\theta}} \\ \frac{\partial \phi_i}{\partial \mu_{t,x}} & \frac{\partial \phi_i}{\partial \mu_{t,y}} & \frac{\partial \phi_i}{\partial \mu_{t,\theta}} \end{pmatrix} \] \( \text{Jacobian of } h \text{ w.r.t. location} \)

6. \[ Q_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix} \]

7. \[ S_t = H_t \Sigma_t H_t^T + Q_t \] \( \text{Pred. measurement covariance} \)

8. \[ K_t = \Sigma_t H_t^T S_t^{-1} \] \( \text{Kalman gain} \)

9. \[ \mu_t = \mu_t + K_t (z_t - \hat{z}_t) \] \( \text{Updated mean} \)

10. \[ \Sigma_t = (I - K_t H_t) \Sigma_t \] \( \text{Updated covariance} \)
EKF Observation Prediction / Correction Step
Estimation Sequence (1)
Estimation Sequence (2)
Comparison to GroundTruth
EKF Summary

• **Highly efficient**: Polynomial in measurement dimensionality $k$ and state dimensionality $n$:
  
  \[ O(k^{2.376} + n^2) \]

• **Not optimal**!
• Can **diverge** if nonlinearities are large!
• Works surprisingly well even when all assumptions are violated!
Multi-hypothesis Tracking