Kalman Filter

Zhang Hongxin
zhx@cad.zju.edu.cn

State Key Lab of CAD&CG, ZJU
2007-03-22
Rudolf Emil Kalman

- Born 1930 in Hungary
- BS and MS from MIT
- PhD 1957 from Columbia
- Filter developed in 1960-61
- Now retired
What is a Kalman Filter?

- Just some applied math.
- A linear system: $f(a+b) = f(a) + f(b)$.
- Noisy data in :: hopefully less noisy out.
- But delay is the price for filtering...
- Pure KF does not even adapt to the data.

- An “optimal recursive data processing algorithm”
What is it used for?

- Tracking missiles
- Tracking heads/hands/drumsticks
- Extracting lip motion from video
- Fitting Bezier patches to point data
- Economics
- Navigation
A really simple example
The Process to be Estimated

- Discrete-time controlled process
  - State estimation:
    \[ x_k = A x_{k-1} + B u_{k-1} + w_{k-1} \]
    \[ x_k \in \mathbb{R}^n \]
  - Measurement:
    \[ z_k = H x_{k-1} + v_{k-1} \]
    \[ z_k \in \mathbb{R}^m \]
  - Process noise covariance: \( Q \)
    \[ p(w) \sim N(0, Q) \]
  - Measurement noise covariance: \( R \)
    \[ p(v) \sim N(0, R) \]
The computational Origins of the Filters

- **Priori** state estimation error at step $k$
  \[ e_k^- := x_k - \hat{x}_k^- \quad P_k^- = E[e_k^- e_k^-^T] \]

- **Posteriori** estimation error
  \[ e_k := x_k - \hat{x}_k \quad P_k = E[e_k e_k^T] \]

- **Posteriori** as a linear combination of a **Priori**

\[
\begin{align*}
x_k &= Ax_{k-1} + Bu_{k-1} + w_{k-1} \\
z_k &= Hx_{k-1} + v_{k-1} \\
\hat{x}_k &= \hat{x}_k^- + K(z_k - H\hat{x}_k^-)
\end{align*}
\]

Measurement *innovation* or *residual*
The computational Origins of the Filters

\[ \hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \]

- The *gain* or *blending factor* that minimizes the a posteriori error covariance \( P_k = E[e_k e_k^T] \)

\[ K_k = \frac{P_k^- H^T}{H P_k^- H^T + R} \]

\[ \lim_{R \to 0} K_k = H^{-1} \quad \lim_{P_k^- \to 0} K_k = 0 \]
The Probabilistic Origins of the Filter

\[ E[x_k] = \hat{x}_k \]
\[ E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] = P_k \]

- The a posteriori state estimate \( \hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \) reflects the mean of the state distribution.
- The a posteriori state estimate error covariance \( P_k = E[e_k e_k^T] \) reflects the variance of the state distribution.

\[ p(x_k \mid z_k) \sim N(E[x_k], E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]) \]
\[ = N(x_k, P_k) \]
The Discrete Kalman Filter Algorithm

- **Time update equations**

\[
\hat{x}^-_k = A\hat{x}_k + Bu_{k-1} \\
P^-_k = AP_k A^T + Q
\]

- **Measurement update equations**

\[
K_k = \frac{P^-_k H^T}{HP^-_k H^T + R} \\
\hat{x}_k = \hat{x}^-_k + K_k (z_k - H\hat{x}^-_k) \\
P_k = (I - K_k H)P^-_k
\]
Filter Parameters and Tuning

- The measurement noise covariance $R$ is usually measured prior to operation of the filter.

- $Q$ and $R$ are generally constants during filtering. Superior filter performance can be obtained by tuning them, referred to as system identification.
Example: 2D Position-Only

- Apparatus: 2D Tablet
Process Model

\[
\begin{bmatrix}
  x_k \\
  y_k
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 \\
  0 & 1
\end{bmatrix}
\begin{bmatrix}
  x_{k-1} \\
  y_{k-1}
\end{bmatrix}
+\begin{bmatrix}
  \sim x_{k-1} \\
  \sim y_{k-1}
\end{bmatrix}
\]

State \( k \) \quad State \ transition \quad State \( k-1 \) \quad Noise

\[
x_k = Ax_{k-1} + w_{k-1}
\]
Measurement Model

\[
\begin{bmatrix}
    u_k \\
    v_k 
\end{bmatrix} =
\begin{bmatrix}
    h_x & 0 \\
    0 & h_y 
\end{bmatrix} \begin{bmatrix}
    x_k \\
    y_k 
\end{bmatrix} + \begin{bmatrix}
    \sim u_k \\
    \sim v_k 
\end{bmatrix}
\]

Measurement $k$  Measurement matrix  State $k$  Noise

\[
\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k
\]
Preparation

\[
A = \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\]

State Transition

\[
Q = E \left\{ w \cdot w^T \right\} = \begin{bmatrix}
Q_{xx} & 0 \\
0 & Q_{yy}
\end{bmatrix}
\]

Process

\[
R = E \left\{ v \cdot v^T \right\} = \begin{bmatrix}
R_{xx} & 0 \\
0 & R_{yy}
\end{bmatrix}
\]

Measurement

Noise Covariance

Noise Covariance
Initialization

\[ x_0 = H z_0 \]

\[ P = \begin{bmatrix} \varepsilon & 0 \\ 0 & \varepsilon \end{bmatrix} \]
Predict

\[ x_k^- = A x_{k-1} \]

\[ P_k^- = A P_{k-1} A^T + Q \]

transition  uncertainty
\[ K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \]
\[ \hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \]
\[ P_k = (I - K_k H) P_k^- \]
Results: XY Track
Y Track: Moving then Still

```
<table>
<thead>
<tr>
<th>Time [seconds]</th>
<th>y [meters]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0</td>
</tr>
<tr>
<td>5</td>
<td>0.8</td>
</tr>
<tr>
<td>10</td>
<td>0.6</td>
</tr>
<tr>
<td>15</td>
<td>0.4</td>
</tr>
<tr>
<td>20</td>
<td>0.2</td>
</tr>
<tr>
<td>25</td>
<td>0.1</td>
</tr>
<tr>
<td>30</td>
<td>0.0</td>
</tr>
</tbody>
</table>
```

**Truth**

**Estimate**
Motion-Dependent Performance

significant latency when moving...

...relatively smooth when not
The **Extended Kalman Filter**

- **Nonlinear Process (Model)**
  - Process dynamics: \( A \) becomes \( a(x) \)
  - Measurement: \( H \) becomes \( h(x) \)

\[
x_k = a(x_{k-1})x_{k-1} + Bu_{k-1} + w_{k-1}
\]

\[
z_k = h(x_{k-1})x_{k-1} + v_{k-1}
\]

- **Filter Reformulation**
  - Use functions instead of matrices
  - Use Jacobians to project forward, and to relate measurement to state
Jacobian?

- Partial derivative of measurement with respect to state

- If measurement is a vector of length $M$ and state has length $N$
  - Jacobian of measurement function will be $M \times N$ matrix of numbers (not equations)

- Evaluating $h(x)$ and $Jacobian(h(x))$ at the same time mostly only cost a little additional computing time.
New Approaches

- Several extensions are available that work better than the EKF in some circumstances
Summary

- A set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process.

- Minimizes the mean of the squared error

- Powerful:
  - supports estimations of past, present, and even future states,
  - can do so even when the precise nature of the modeled system is unknown
The End of Kalman Filter
Before the end of this course

- Many techniques I cannot mention yet:
  - Neural network
  - Graphical model
  - Genetic methods
  - ...

- It is just a beginning …