Lecture 9
The Extended Kalman filter

• Nonlinear filtering
• Extended Kalman filter
• Linearization and random variables
Nonlinear filtering

- nonlinear Markov model:

\[ x_{t+1} = f(x_t, w_t), \quad y_t = g(x_t, v_t) \]

- \( f \) is (possibly nonlinear) dynamics function
- \( g \) is (possibly nonlinear) measurement or output function
- \( w_0, w_1, \ldots, v_0, v_1, \ldots \) are independent
- even if \( w, v \) Gaussian, \( x \) and \( y \) need not be

- nonlinear filtering problem: find, e.g.,

\[ \hat{x}_{t|t-1} = \mathbb{E}(x_t | y_0, \ldots, y_{t-1}), \quad \hat{x}_{t|t} = \mathbb{E}(x_t | y_0, \ldots, y_t) \]

- general nonlinear filtering solution involves a PDE, and is not practical
Extended Kalman filter

- extended Kalman filter (EKF) is *heuristic* for nonlinear filtering problem
- often works well (when tuned properly), but sometimes not
- widely used in practice
- based on
  - linearizing dynamics and output functions at current estimate
  - propagating an approximation of the conditional expectation and covariance
Linearization and random variables

- consider $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$

- suppose $\mathbb{E} x = \bar{x}$, $\mathbb{E}(x - \bar{x})(x - \bar{x})^T = \Sigma_x$, and $y = \phi(x)$

- if $\Sigma_x$ is small, $\phi$ is not too nonlinear,

$$ y \approx \tilde{y} = \phi(\bar{x}) + D\phi(\bar{x})(x - \bar{x}) $$

- gives approximation for mean and covariance of nonlinear function of random variable:

$$ \bar{y} \approx \phi(\bar{x}), \quad \Sigma_y \approx D\phi(\bar{x})\Sigma_x D\phi(\bar{x})^T $$

- if $\Sigma_x$ is not small compared to ‘curvature’ of $\phi$, these estimates are poor
• a good estimate can be found by Monte Carlo simulation:

\[
\bar{y} \approx \bar{y}_{mc} = \frac{1}{N} \sum_{i=1}^{N} \phi(x^{(i)})
\]

\[
\Sigma_y \approx \frac{1}{N} \sum_{i=1}^{N} \left( \phi(x^{(i)}) - \bar{y}_{mc} \right) \left( \phi(x^{(i)}) - \bar{y}_{mc} \right)^T
\]

where \(x^{(1)}, \ldots, x^{(N)}\) are samples from the distribution of \(x\), and \(N\) is large

• another method: use Monte Carlo formulas, with a small number of nonrandom samples chosen as ‘typical’, e.g., the 90% confidence ellipsoid semi-axis endpoints

\[
x^{(i)} = \bar{x} \pm \beta v_i, \quad \Sigma_x = V \Lambda V^T
\]

The Extended Kalman filter
Example

\[ x \sim \mathcal{N}(0, 1), \quad y = \exp(x) \]

(for this case we can compute mean and variance of \( y \) exactly)

<table>
<thead>
<tr>
<th></th>
<th>( \bar{y} )</th>
<th>( \sigma_y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact values</td>
<td>( e^{1/2} = 1.649 )</td>
<td>( \sqrt{e^2 - e} = 2.161 )</td>
</tr>
<tr>
<td>linearization</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Monte Carlo (( N = 10 ))</td>
<td>1.385</td>
<td>1.068</td>
</tr>
<tr>
<td>Monte Carlo (( N = 100 ))</td>
<td>1.430</td>
<td>1.776</td>
</tr>
<tr>
<td>Sigma points (( x = \bar{x}, \quad \bar{x} \pm 1.5\sigma_x ))</td>
<td>1.902</td>
<td>2.268</td>
</tr>
</tbody>
</table>
Extended Kalman filter

- **initialization**: \( \hat{x}_{0|-1} = \bar{x}_0 \), \( \Sigma(0|-1) = \Sigma_0 \)

- **measurement update**
  
  - linearize output function at \( x = \hat{x}_{t|t-1} \):

    \[
    C = \frac{\partial g}{\partial x}(\hat{x}_{t|t-1}, 0)
    \]

    \[
    V = \frac{\partial g}{\partial v}(\hat{x}_{t|t-1}, 0) \Sigma_v \frac{\partial g}{\partial v}(\hat{x}_{t|t-1}, 0)^T
    \]

  - measurement update based on linearization

    \[
    \hat{x}_{t|t} = \hat{x}_{t|t-1} + \Sigma_{t|t-1} C^T \left( C \Sigma_{t|t-1} C^T + V \right)^{-1} \ldots (y_t - g(\hat{x}_{t|t-1}, 0))
    \]

    \[
    \Sigma_{t|t} = \Sigma_{t|t-1} - \Sigma_{t|t-1} C^T \left( C \Sigma_{t|t-1} C^T + V \right)^{-1} C \Sigma_{t|t-1}
    \]
\textbullet{} \textit{time update}

- linearize dynamics function at $x = \hat{x}_{t|t}$:

$$A = \frac{\partial f}{\partial x}(\hat{x}_{t|t}, 0)$$

$$W = \frac{\partial f}{\partial w}(\hat{x}_{t|t}, 0)\Sigma_w \frac{\partial f}{\partial w}(\hat{x}_{t|t}, 0)^T$$

- time update based on linearization

$$\hat{x}_{t+1|t} = f(\hat{x}_{t|t}, 0), \quad \Sigma_{t+1|t} = A\Sigma_{t|t}A^T + W$$

\textbullet{} replacing linearization with Monte Carlo yields \textit{particle filter}

\textbullet{} replacing linearization with sigma-point estimates yields \textit{unscented Kalman filter} (UKF)
Example

• $p_t, u_t \in \mathbb{R}^2$ are position and velocity of vehicle, with $(p_0, u_0) \sim \mathcal{N}(0, I)$

• vehicle dynamics:

\[
p_{t+1} = p_t + 0.1 u_t, \quad u_{t+1} = \begin{bmatrix} 0.85 & 0.15 \\ -0.1 & 0.85 \end{bmatrix} u_t + w_t
\]

$w_t$ are IID $\mathcal{N}(0, I)$

• measurements: noisy measurements of distance to 9 points $p_i \in \mathbb{R}^2$

\[
(y_t)_i = \|p_t - p_i\| + (v_t)_i, \quad i = 1, \ldots, 9,
\]

$(v_t)_i$ are IID $\mathcal{N}(0, 0.3^2)$
EKF results

- EKF initialized with $\hat{x}_{0|1} = 0$, $\Sigma(0|1) = I$, where $x = (p, u)$
- $p_i$ shown as stars; $p_t$ as dotted curve; $\hat{p}_{t|t}$ as solid curve
Current position estimation error

\[ \| \hat{p}_t \| - p_t \| \text{ versus } t \]
Current position estimation predicted error

\[(\Sigma(t|t)_{11} + \Sigma(t|t)_{22})^{1/2}\] versus \(t\)