Bayes Filter Reminder

- **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**
  \[
  \text{bel}(x_t) = \eta \, p(z_t \mid x_t) \overline{\text{bel}}(x_t)
  \]

**Properties of Gaussians**

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

$$z_t = C_t x_t + \delta_t$$

- $A_t$ 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 $z_t$.
- $\epsilon_t$ 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

\[ \begin{align*}
\bar{x}_t &= \bar{x}_{t-1} + K_t (z_t - \bar{x}_{t-1}) \\
\text{with} & \quad K_t = \frac{\tilde{\Sigma}_t}{\tilde{\Sigma}_t + \sigma_{\text{noise}}^2}
\end{align*} \]

\[ \begin{align*}
\bar{x}_t &= \bar{x}_{t-1} + K_t (z_t - \bar{x}_{t-1}) \\
\Sigma_t &= (I - K_tC_t)\Sigma_{t-1}
\end{align*} \]

with \( K_t = \Sigma_t C_t^T (C_t \Sigma_t C_t^T + Q)^{-1} \)
Linear Gaussian Systems: Initialization

- Initial belief is normally distributed:

\[ \text{bel}(x_0) = \mathcal{N}(x_0; \mu_0, \Sigma_0) \]

Linear Gaussian Systems: Dynamics

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

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

\[
p(x_t | u_{t-1}, x_{t-1}) = \mathcal{N}(x_t; A_t x_{t-1} + B_t u_t, R_t)
\]

\[
\text{bel}(x_t) = \int p(x_t | u_{t-1}, x_{t-1}) \, \text{bel}(x_{t-1}) \, dx_{t-1}
\]

\[
\downarrow \quad \downarrow
\]

\[
\sim \mathcal{N}(x_t; A_t x_{t-1} + B_t u_t, R_t) \sim \mathcal{N}(x_{t-1}; \mu_{t-1}, \Sigma_{t-1})
\]

Linear Gaussian Systems: Observations

- Observations are linear function of state plus additive noise:

\[
z_t = C_t x_t + \delta_t
\]

\[
p(z_t | x_t) = \mathcal{N}(z_t; C_t x_t, Q_t)
\]

\[
\text{bel}(x_t) = \eta \int p(z_t | x_t) \, \text{bel}(x_t) \, dx_t
\]

\[
\downarrow \quad \downarrow
\]

\[
\sim \mathcal{N}(z_t; C_t x_t, Q_t) \sim \mathcal{N}(x_t; \bar{\mu}_t, \bar{\Sigma}_t)
\]
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**!

**Linear Gaussian Systems: Observations**

\[
\begin{align*}
\text{bel}(x_i) &= \eta \ p(z_i|x_i) \ 	ext{bel}(x_i) \\
&- N(z_i; C, \Sigma) \\
&- N(x_i; \mu_i, \Sigma_i)
\end{align*}
\]

\[
\begin{align*}
\text{bel}(x_i) &= \eta \exp\left(-\frac{1}{2} (z_i - C, x_i)'^T \Sigma_i^{-1} (z_i - C, x_i)\right) \\
&\quad \exp\left(-\frac{1}{2} (x_i - \mu_i)'^T \Sigma_i^{-1} (x_i - \mu_i)\right)
\end{align*}
\]

\[
\begin{align*}
\mu_i &= \bar{\mu} + K_i (z_i - C, \bar{\mu}) \\
\Sigma_i &= (I - K_i C) \Sigma_i
\end{align*}
\]

\[
K_i = \Sigma_i C_i (C_i \Sigma_i C_i'^T + Q)^{-1}
\]

**Kalman Filter Algorithm**

1. Algorithm **Kalman_filter**($\mu_{i-1}, \Sigma_{i-1}, u_i, z_i$):
2. **Prediction**:
3. \[ \bar{\mu}_i = A \mu_{i-1} + B u_i \]
4. \[ \bar{\Sigma}_i = A \Sigma_{i-1} A' + R \]
5. **Correction**:
6. \[ K_i = \Sigma_i C_i (C_i \Sigma_i C_i'^T + Q)^{-1} \]
7. \[ \mu_i = \mu_{i-1} + K_i (z_i - C, \mu_{i-1}) \]
8. \[ \Sigma_i = (I - K_i C) \Sigma_i \]
9. **Return** \( \mu_i, \Sigma_i \)

**EXTENDED KALMAN FILTER**

Going non-linear
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 (2)

EKF Linearization (3)

EKF Linearization: First Order Taylor Series Expansion

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

- **Correction:**
  \[
  h(x_t) = h(\bar{\mu}) + \frac{\partial h(\bar{\mu})}{\partial x} (x_{t} - \bar{\mu})
  \]
  \[
  h(x_t) = h(\bar{\mu}) + H_{x} (x_{t} - \bar{\mu})
  \]

EKF Algorithm

1. **Extended Kalman filter** \((\mu_{t-1}, x_{t-1}, u_t, z_t):\)
2. Prediction:
   \[
   \bar{\mu}_{t} = g(u_t, \mu_{t-1})
   \]
   \[
   \bar{\mu}_{t} = A_{t} \mu_{t-1} + B_{t} u_t
   \]
   \[
   \bar{x}_{t} = G_{x} \Sigma_{t-1} G_{x}^{T} + R_t
   \]
   \[
   \bar{x}_{t} = A_{t} \Sigma_{t-1} A_{t}^{T} + R_t
   \]
3. Correction:
   \[
   K_t = \Sigma_{t} H_{x}^{T} (H_{x} \Sigma_{t} H_{x}^{T} + Q)^{-1}
   \]
   \[
   K_t = \Sigma_{t} C_{t}^{T} (C_{t} \Sigma_{t} C_{t}^{T} + Q)^{-1}
   \]
   \[
   \mu_{t} = \mu_{t} + K_t (z_t - h(\bar{\mu}))
   \]
   \[
   \mu_{t} = \mu_{t} + K_t (z_t - C_{t} \mu_{t})
   \]
   \[
   \Sigma_{t} = (I - K_t H_{x}) \Sigma_{t}
   \]
   \[
   \Sigma_{t} = (I - K_t C_{t}) \Sigma_{t}
   \]
4. Return \(\mu_t, \Sigma_t\)

\[
H = \frac{\partial h(\bar{\mu})}{\partial x_t}
\]
\[
G_{x} = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}
\]
**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)

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

   **Prediction:**
   3. $a = \frac{\partial g(u, \mu)}{\partial u} = \begin{pmatrix} \frac{\partial a}{\partial u} \frac{\partial a}{\partial v} \frac{\partial a}{\partial \theta} \end{pmatrix}$
      Jacobian of $g$ w.r.t location
   5. $v = \frac{\partial g(u, \mu)}{\partial \Sigma} = \begin{pmatrix} \frac{\partial v}{\partial \Sigma} \frac{\partial v}{\partial u} \frac{\partial v}{\partial \theta} \end{pmatrix}$
      Jacobian of $g$ w.r.t control
   6. $M = \begin{pmatrix} a_w^2 + a_w^2 & 0 & a_w^2 + a_w^2 \end{pmatrix}$
      Motion noise
   7. $\tilde{\mu} = g(u, \mu_{t-1})$
   8. $\tilde{\Sigma} = G\Sigma_u G_t + V_M V_t$
      Predicted mean
      Predicted covariance

**Landmark-based Localization**

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

   **Prediction:**
   $\theta = \mu_{t-1, \theta}$
   $G_t = \begin{pmatrix} 1 & 0 & -\frac{\beta_t}{\omega_t} \cos \theta + \frac{\beta_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ 0 & 1 & -\frac{\beta_t}{\omega_t} \sin \theta + \frac{\beta_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \end{pmatrix}$
   $V_t = \begin{pmatrix} \frac{\omega_t}{\beta_t} \cos \theta \cos(\theta + \omega_t \Delta t) & \frac{\omega_t}{\beta_t} \cos \theta \sin(\theta + \omega_t \Delta t) \\ \frac{\omega_t}{\beta_t} \cos(\theta + \omega_t \Delta t) & \frac{\omega_t}{\beta_t} \sin(\theta + \omega_t \Delta t) \end{pmatrix}$
   $M_t = \begin{pmatrix} \alpha_1 \omega_t^2 + \alpha_2 \omega_t^2 & 0 & 0 & \alpha_3 \omega_t^2 + \alpha_4 \omega_t^2 \\ 0 & \alpha_1 \omega_t^2 + \alpha_2 \omega_t^2 & \omega_t \Delta t \\ 0 & \alpha_3 \omega_t^2 + \alpha_4 \omega_t^2 & \omega_t \Delta t \end{pmatrix}$
   $\tilde{\mu} = \mu_{t-1} + \begin{pmatrix} \frac{\beta_t}{\omega_t} \cos \theta - \frac{\beta_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \frac{\beta_t}{\omega_t} \sin \theta - \frac{\beta_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \end{pmatrix}$
   $\tilde{\Sigma} = G\Sigma_u G_t' + V_M V_t'$
      Predicted covariance
1. EKF_localization \((\mu_{t-1}, \Sigma_{t-1}, u_t)\)

Correction:

3. \(z_t = \left[ \sqrt{\alpha_t^2 - \beta_t^2} \cos \beta_t - \frac{\alpha_t}{\sqrt{\alpha_t^2 - \beta_t^2}} \sin \beta_t \right] \) Predicted measurement mean

5. \(H_t = \frac{\partial \phi(\mu_t, m_t)}{\partial x_t} = \begin{bmatrix} \frac{\partial \phi}{\partial x} & \frac{\partial \phi}{\partial y} & \frac{\partial \phi}{\partial \theta} \end{bmatrix} \) Jacobian of \(h\) w.r.t. location

6. \(\alpha = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \)

7. \(S_t = H_t \Sigma_t H_t^T + Q_t\) Pred. measurement covariance

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

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

10. \(\Sigma_t = (I - K_t H_t) \Sigma_t\) Updated covariance
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!
Going unscented

**UNSCENTED KALMAN FILTER**

Linearization via Unscented Transform

UKF Sigma-Point Estimate (2)

UKF Sigma-Point Estimate (3)
### Unscented Transform

**Sigma points**

\[
\chi^i = \mu, \quad w^2_i = \frac{\lambda}{n + \lambda}, \quad w^0_i = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta)
\]

\[
\chi^i = \mu \pm \sqrt{(n + \lambda)\Sigma}
\]

\[
w^0_i = w^i_2 = \frac{1}{2(n + \lambda)} \quad \text{for } i = 1, \ldots, 2n
\]

Pass sigma points through nonlinear function

\[
\psi^i = g(\chi^i)
\]

Recover mean and covariance

\[
\mu = \sum_{i=1}^{2n} w_i^0 \psi^i
\]

\[
\Sigma = \sum_{i=1}^{2n} w^0_i (\psi^i - \mu)(\psi^i - \mu)^T
\]

---

### UKF Prediction Step

**UKF Observation Prediction Step**

**UKF Correction Step**
### UKF_predict\((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\):

<table>
<thead>
<tr>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction</td>
<td>(M = \begin{bmatrix} 0 &amp;</td>
</tr>
<tr>
<td></td>
<td>(Q = \begin{bmatrix} \sigma_w^2 &amp; 0 \ 0 &amp; \sigma_w^2 \end{bmatrix} ) Measurement noise</td>
</tr>
<tr>
<td>(\mu_t^*)</td>
<td>(\begin{bmatrix} \mu_t^* \ 0 \end{bmatrix} ) Augmented state mean</td>
</tr>
<tr>
<td>(\Sigma_t^*)</td>
<td>(\begin{bmatrix} 0 &amp; 0 \ 0 &amp; 0 \end{bmatrix} ) Augmented covariance</td>
</tr>
<tr>
<td>Sigma points</td>
<td>(X_t^* = [\mu_t^<em>, \mu_t^</em> + \sqrt{\Sigma_t^<em>} \mu_t^</em>, \mu_t^* - \sqrt{\Sigma_t^<em>} \mu_t^</em>] ) Prediction of sigma points</td>
</tr>
<tr>
<td>Predicted mean</td>
<td>(\Sigma = \sum_{i=1}^{2K} w^<em><em>i (X</em>{t,i}^</em> - \mu_t^<em>) (X_{t,i}^</em> - \mu_t^*)^T ) Predicted covariance</td>
</tr>
</tbody>
</table>

### UKF_correct\((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\):

<table>
<thead>
<tr>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correction</td>
<td>(Z_t = h(\bar{X}_t) + \chi_t^* ) Measurement sigma points</td>
</tr>
<tr>
<td></td>
<td>(\bar{z}<em>i = \sum</em>{j=1}^{2K} w^*<em>j Z</em>{t,j} ) Predicted measurement mean</td>
</tr>
<tr>
<td>(S_t)</td>
<td>(\sum_{i=1}^{2K} w^<em><em>i (\chi</em>{t,i}^</em> - \bar{z}<em>i) (\chi</em>{t,i}^* - \bar{z}_i)^T ) Pred. measurement covariance</td>
</tr>
<tr>
<td>Cross-covariance</td>
<td>(K_t = \Sigma_t^* S_t^{-1} ) Kalman gain</td>
</tr>
<tr>
<td>Updated mean</td>
<td>(\mu_t = \bar{\mu}_t + K_t (z_t - \hat{z}_t) ) Updated mean</td>
</tr>
<tr>
<td>Updated covariance</td>
<td>(\Sigma_t = \Sigma_t - K_t S_t K_t^T ) Updated covariance</td>
</tr>
</tbody>
</table>

### Estimation Sequence

- **EKF**
- **PF**
- **UKF**
**Prediction Quality**

- **EKF**
- **UKF**

**UKF Summary**
- **Highly efficient**: Same complexity as EKF, with a constant factor slower in typical practical applications
- **Better linearization than EKF**: Accurate in first two terms of Taylor expansion (EKF only first term)
- **Derivative-free**: No Jacobians needed
- **Still not optimal!**

**Kalman Filter-based System**

- [Arras et al. 98]:
  - Laser range-finder and vision
  - High precision (<1cm accuracy)

**Multi-hypothesis Tracking**
**Localization With MHT**

- Belief is represented by multiple hypotheses
- Each hypothesis is tracked by a Kalman filter

**Additional problems:**
- **Data association**: Which observation corresponds to which hypothesis?
- **Hypothesis management**: When to add / delete hypotheses?
- Huge body of literature on target tracking, motion correspondence etc.

**MHT: Implemented System (1)**

- Hypotheses are extracted from LRF scans
- Each hypothesis has probability of being the correct one:
  \[ H_i = \{ x_i, \Sigma_i, P(H_i) \} \]
- Hypothesis probability is computed using Bayes' rule
  \[ P(H_i | s) = \frac{P(s | H_i) P(H_i)}{P(s)} \]
- Hypotheses with low probability are deleted.
- New candidates are extracted from LRF scans.

\[ C_j = \{ z_j, R_j \} \]  

[Jensfelt et al. '00]

**MHT: Implemented System (2)**

**Example run**

- # hypotheses
- \( P(H_{best}) \)

**Map and trajectory**  
# hypotheses vs. time

Courtesy of P. Jensfelt and S. Kristensen