**Bayes Filter Reminder**

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

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

**Properties of Gaussians**

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

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

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

- 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.

**Components of a Kalman Filter**

- **A**: Matrix (nxn) that describes how the state evolves from \( t-1 \) to \( t \) without controls or noise.
- **B**: Matrix (nxl) that describes how the control \( u \) changes the state from \( t \) to \( t-1 \).
- **C**: 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.

**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

\[
z_t = C_t x_t + \delta_t
\]

**Kalman Filter Updates in 1D**
Kalman Filter Updates in 1D

\[
\begin{align*}
\text{bel}(x_t) &= \begin{cases} 
\mu_t = \mu_t + K_t (z_t - \mu_t) \\
\sigma_t^2 = (1 - K_t) \sigma_t^2
\end{cases} \quad \text{with} \quad K_t = \frac{\sigma_t^2}{\sigma_t^2 + \sigma_{w,t}} \\
\text{bel}(x_t) &= \begin{cases} 
\mu_t = \mu_t + K_t (C_t \mu_t - z_t) \\
\Sigma_t = (I - K_t C_t) \Sigma_t
\end{cases} \quad \text{with} \quad K_t = \Sigma_t C_t^T (C_t \Sigma_t C_t^T + Q_t)^{-1}
\end{align*}
\]

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:

\[
x_t = A x_{t-1} + B u_t + \epsilon_t
\]

\[
p(x_t | u_t, x_{t-1}) = N(x_t; A x_{t-1} + B u_t, R_t)
\]

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

\[
\downarrow
\]

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

Linear Gaussian Systems: Observations

- Observations are linear function of state plus additive noise:

\[
z_t = C_t x_t + \delta_t
\]

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

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

\[
\downarrow
\]

\[
\sim N(z_t; C_t x_t, Q_t) \sim N(x_t; \mu_t, \Sigma_t)
\]

Linear Gaussian Systems: Dynamics

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

\[
\downarrow \quad \downarrow
\]

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

\[
\text{bel}(x_t) = \eta \int \exp \left\{-\frac{1}{2} (x_t - A x_{t-1} - B u_t)^T R_t^{-1} (x_t - A x_{t-1} - B u_t) \right\} \text{bel}(x_{t-1}) dx_{t-1}
\]

\[
\sim N(x_t; \mu_t, \Sigma_t)
\]

\[
\text{bel}(x_t) = \eta \exp \left\{-\frac{1}{2} (z_t - C_t x_t)^T Q_t^{-1} (z_t - C_t x_t) \right\} \text{bel}(x_{t-1}) dx_{t-1}
\]

\[
\sim N(z_t; C_t x_t, Q_t)
\]

\[
\text{bel}(x_t) = \left\{ \begin{array}{l}
\mu_t = \mu_t + K_t (z_t - C_t x_t) \\
\Sigma_t = (I - K_t C_t) \Sigma_t
\end{array} \right\}
\]

with

\[
K_t = \Sigma_t C_t^T (C_t \Sigma_t C_t^T + Q_t)^{-1}
\]
Kalman Filter Algorithm

1. Algorithm Kalman_filter(μ₀, Σ₀, uᵣ, vᵣ):

2. Prediction:
   \[ \hat{\mu}_t = A\mu_{t-1} + Bu \]
   \[ \Sigma_t = A\Sigma_{t-1}A^T + R_t \]

3. Correction:
   \[ K_t = \Sigma_t C_t^T (C_t \Sigma_t C_t^T + Q_t)^{-1} \]
   \[ \mu_t = \hat{\mu}_t + K_t (z_t - C_t \hat{\mu}_t) \]
   \[ \Sigma_t = (I - K_t C_t) \Sigma_t \]

4. 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**!

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

**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)

**EKF Algorithm**

1. \( \text{Extended Kalman_filter}(u_{t-1}, \Sigma_{t-1}, u_t, z_t) \):
   \[ \tilde{\mu}_t = A \mu_{t-1} + B u_t \]
   \[ \tilde{\Sigma}_t = A \Sigma_{t-1} A^T + R \]

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

3. Correction:
   \[ K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \]
   \[ \mu_t = \bar{\mu}_t + K_t (z_t - H_t \bar{\mu}_t) \]
   \[ \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \]

4. Return \( \mu, \Sigma \):
   \[ H_t = \frac{\partial h(\bar{x}_t)}{\partial x_t}, \quad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} \]

**Landmark-based Localization**
1. **EKF_localization** \((\bar{\mu}, \Sigma) \rightarrow (\hat{\mu}, \hat{\Sigma})\):

**Prediction:**

\[
\dot{\hat{\mu}} = \hat{\mu}_0 + H \hat{\Sigma}_0 \dot{\mu},
\]

\[
\dot{\hat{\Sigma}} = \Sigma_0 + H \Sigma_0 \dot{\mu} + \dot{\mu}^T H^T \Sigma_0 + Q.
\]

3. \( G = \frac{\partial (\bar{\mu}, \Sigma)}{\partial \mu} \) Jacobian of \( g \) w.r.t location

5. \( V = \frac{\partial (g, \mu)}{\partial \mu} \) Jacobian of \( g \) w.r.t control

6. \( M = \left[ \begin{array}{cc} \alpha \sigma^2 & 0 \\ 0 & \alpha \sigma^2 \end{array} \right] \) Motion noise

7. \( \bar{\mu}_t = g(\mu_t, \mu_{t-1}) \) Predicted mean

8. \( \Sigma_t = G \Sigma_{t-1} G^T + V M V^T \) Predicted covariance

---

1. **EKF_localization** \((\bar{\mu}, \Sigma) \rightarrow (\hat{\mu}, \hat{\Sigma})\):

**Correction:**

\[
\bar{\Sigma}_t = \Sigma_t + \Sigma_t \bar{H}_t^T \Sigma_t^{-1} \Sigma_t \bar{H}_t.
\]

3. \( \bar{z}_t = \left( \begin{array}{c} \bar{z}_t^T \\ \bar{z}_t^T \end{array} \right) \) Predicted measurement mean

4. \( \bar{y}_t = \frac{\partial (\bar{z}_t, \Sigma)}{\partial \mu} = \left( \begin{array}{c} \frac{\partial \bar{y}_t}{\partial \bar{y}_t^T} \frac{\partial \Sigma_t}{\partial \Sigma_t^T} \frac{\partial \Sigma_t}{\partial \Sigma_t^T} \end{array} \right) \) Jacobian of \( h \) w.r.t location

6. \( Q = \left[ \begin{array}{cc} \sigma^2 & 0 \\ 0 & \sigma^2 \end{array} \right] \) 

7. \( S_t = H \Sigma_t H^T + Q \) Pred. measurement covariance

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

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

10. \( \hat{\Sigma}_t = (I - K_t H) \Sigma_t \) Updated covariance
EKF 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!

**UNSCENTED KALMAN FILTER**

**Linearization via Unscented Transform**

- EKF
- UKF

**UKF Sigma-Point Estimate (2)**

- EKF
- UKF
**UKF Sigma-Point Estimate (3)**

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

**Unscented Transform**

Sigma points

\[ \chi^0 = \mu \]

Weights

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

Pass sigma points through nonlinear function

\[ \psi_i = g(\chi') \]

Recover mean and covariance

\[
\begin{align*}
\mu' &= \sum_{i=1}^{2n} w_i^i \psi_i \\
\Sigma' &= \sum_{i=1}^{2n} w_i^i (\psi_i' - \mu)(\psi_i' - \mu)^T
\end{align*}
\]
**UKF Correction Step**

**UKF_correct** $(\mu, \Sigma, u, z_t)$:

**Correction:**
- $Z_i = h(\hat{X}_i) + \chi_i$: Measurement sigma points
- $\tilde{z}_i = \sum_{i=0}^{N} w^i_i (Z_{i,j} - \hat{z}_i) (Z_{i,j} - \hat{z}_i)^T$: Predicted measurement mean
- $S_i = \sum_{i=0}^{N} w^i_i (Z_{i,j} - \hat{z}_i)(Z_{i,j} - \hat{z}_i)^T$: Pred. measurement covariance
- $\Sigma_i = \sum_{i=0}^{N} w^i_i (\chi_{i,j} - \tilde{z}_i)(\chi_{i,j} - \tilde{z}_i)^T$: Cross-covariance
- $K_i = \Sigma_i^{-1} S_i$: Kalman gain
- $\mu_i = \tilde{z}_i + K_i (z_i - \hat{z}_i)$: Updated mean
- $\Sigma_i = \Sigma_i - K_i S_i K_i^T$: Updated covariance

**UKF_predict** $(\mu, \Sigma, u, z_t)$:

**Prediction:**
- $\mathbf{M}_t = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$: Motion noise
- $\mathbf{Q}_t = \begin{pmatrix} \sigma_t^2 & 0 \\ 0 & \sigma_t^2 \end{pmatrix}$: Measurement noise
- $\mu^c_t = \begin{pmatrix} \mu_t \\ (0,0)^T \end{pmatrix}$: Augmented state mean
- $\Sigma^c_t = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$: Augmented covariance
- $X_{i,t}^c = \left(\mu^c_{i} \quad \mu^c_{i} + \mathbf{Q}_t \sqrt{\Sigma^c_{i-1}} \quad \mu^c_{i} - \mathbf{Q}_t \sqrt{\Sigma^c_{i-1}} \right)$: Sigma points
- $\mathbf{Z}_t = \mathbf{g}(u_t + \chi_{t,1}^c, \chi_{t,-1}^c)$: Prediction of sigma points
- $\mu_i = \sum_{i=0}^{N} w^i_i X_{i,t}^c$: Predicted mean
- $\Sigma_i = \sum_{i=0}^{N} w^i_i (\chi_{i,t} - \mu_i)(\chi_{i,t} - \mu_i)^T$: Predicted covariance

**Estimation Sequence**

- EKF
- PF
- UKF
**Estimation Sequence**

**Prediction Quality**

**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)

Courtesy of K. Arras
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:
  \[ H_i = \{ \hat{s}_i, \Sigma_i, P(H_i) \} \]
- 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.
  \[ C_j = \{ z_j, R_j \} \]

MHT: Implemented System (2)

Courtesy of P. Jensfelt and S. Kristensen [Jensfelt et al. '00]
Projects

- Groups of 1-3, with 1 and 3 being exceptions
- This week, find a partner
- Next Mon/Tue meet w/ Arun for initial discussion
- Next Thu/Fri meet w/ me to finalize things
- Keep project blog, midterm meeting
- Poster / demo session 12/14 2:30 – 4:20pm