Bayes Filter Reminder

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

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

### Gaussians

**Univariate**

- \( p(x) \sim N(\mu, \sigma^2) \):
  \[
  p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
  \]

**Multivariate**

- \( p(x) \sim N(\mu, \Sigma) \):
  \[
  p(x) = \frac{1}{(|2\pi\Sigma|)^{1/2}} e^{-\frac{1}{2} (x-\mu)^T \Sigma^{-1} (x-\mu)}
  \]

### Properties of Gaussians

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

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

Multivariate Gaussians

\[
\begin{align*}
X &\sim N(\mu, \Sigma) \\
Y &= AX + B \\
&\Rightarrow \quad Y \sim N(A\mu + B, A\Sigma A^T)
\end{align*}
\]

- 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.

Discrete Kalman Filter

Estimates the state \( x_t \) 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 \( z_t \).
- \( \epsilon_t \): 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

bel(x_t) = \begin{align*}
\bar{x}_t &= a_t \mu_{t-1} + b_t \mu_t \\
\sigma_{t}^2 &= a_t^2 \sigma_{t-1}^2 + \sigma_{w,t}^2 \\
\Sigma_t &= A_t \Sigma_{t-1} A_t^T + R
\end{align*}

with 
K_t = \frac{\sigma_{t}^2}{\sigma_{t}^2 + \sigma_{w,t}^2}

\mu_t = \bar{x}_t + K_t (z_t - \bar{x}_t)
\sigma_{t}^2 = (1 - K_t) \sigma_{t}^2

\Sigma_t = (I - K_t \Sigma_t) \Sigma_t

Kalman Filter Updates in 1D

Kalman Filter Updates
**Linear Gaussian Systems: Initialization**

• Initial belief is normally distributed:

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

**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:

\[ z_t = C x_t + \delta_t \]
\[ p(z_t | x_t) = N(z_t; C x_t, Q_t) \]

\[ \text{bel}(x_t) = \eta p(z_t | x_t) \text{bel}(x_t) \]
\[ \downarrow \]
\[ \sim N(z_t; C x_t, Q_t) \sim N(x_t; \mu_t, \Sigma_t) \]
Linear Gaussian Systems: Observations

\[
\begin{align*}
\text{bel}(x_i) &= \eta \ p(z_i | x_i) \\
&= \mathcal{N}(z_i; C_i x_i, Q_i) \\
&= \mathcal{N}(x_i; \mu_i, \Sigma_i)
\end{align*}
\]

\[
\begin{align*}
\text{bel}(x_i) &= \eta \exp \left\{ -\frac{1}{2} (z_i - C_i x_i)^T Q_i^{-1} (z_i - C_i x_i) \right\} \\
&\times \exp \left\{ -\frac{1}{2} (x_i - \mu_i)^T \Sigma_i^{-1} (x_i - \mu_i) \right\}
\end{align*}
\]

\[
\begin{align*}
\text{bel}(x_i) &= \left( \mu_i = \mu_i + K_i (z_i - C_i \mu_i) \right) \\
\Sigma_i &= (I - K_i C_i) \Sigma_i \\
\text{with } K_i &= \Sigma_i C_i (C_i \Sigma_i C_i^T + Q_i)^{-1}
\end{align*}
\]

Kalman Filter Algorithm

1. Algorithm Kalman_filter(\(\mu_{-1}, \Sigma_{-1}, u_t, z_t\)):
2. Prediction:
3. \(\tilde{\mu}_t = A \mu_{-1} + B u_t\)
4. \(\tilde{\Sigma}_t = A \Sigma_{-1} A^T + R_t\)
5. Correction:
6. \(K_t = \Sigma_t C_t (C_t \Sigma_t C_t^T + Q_t)^{-1}\)
7. \(\mu_t = \tilde{\mu}_t + K_t (z_t - C_t \tilde{\mu}_t)\)
8. \(\Sigma_t = (I - K_t C_t) \Sigma_t\)
9. Return \(\mu_t, \Sigma_t\)

The Prediction-Correction-Cycle

The Prediction-Correction-Cycle
The Prediction-Correction-Cycle

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

\[ x_t = g(u_t, x_{t-1}) \]

\[ z_t = h(x_t) \]
Linearity Assumption Revisited

Non-linear Function

EKF Linearization (1)

EKF Linearization (2)
**EKF Linearization (3)**

- **Prediction:**
  \[ g(u, x_{i-1}) = g(u, \mu_{i-1}) + \frac{\partial g(u, \mu_{i-1})}{\partial x_{i-1}} (x_{i-1} - \mu_{i-1}) \]
  \[ g(u, x_{i-1}) = g(u, \mu_{i-1}) + G_i (x_{i-1} - \mu_{i-1}) \]

- **Correction:**
  \[ h(x_i) = h(\bar{x}_i) + \frac{\partial h(\bar{x}_i)}{\partial x_i} (x_i - \bar{x}_i) \]
  \[ h(x_i) = h(\bar{x}_i) + H_i (x_i - \bar{x}_i) \]

---

**EKF Algorithm**

1. **Extended_Kalman_filter**\((\mu_{i-1}, \Sigma_{i-1}, u_i, z_i)\):
2. Prediction:
   \[ \bar{\mu}_i = A \mu_{i-1} + B u_i \]
   \[ \bar{\Sigma}_i = A \Sigma_{i-1} A^T + R_i \]
3. Correction:
   \[ K_i = \Sigma_i H_i^T (H_i \Sigma_i H_i^T + Q_i)^{-1} \]
   \[ \mu_i = \bar{\mu}_i + K_i (z_i - h(\bar{x}_i)) \]
   \[ \Sigma_i = (I - K_i H_i) \Sigma_i \]
4. Return \(\mu_i, \Sigma_i\)

**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** \((\mu_{t-1}, \Sigma_{t-1}, U_t, \hat{z}_i)\):

   **Prediction:**
   \[
   \begin{align*}
   &1. \quad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial \mu_{t-1}} = \begin{pmatrix}
   \frac{\partial g}{\partial \mu_{t-1}} & \frac{\partial g}{\partial \mu_{t-1}} & \frac{\partial g}{\partial \mu_{t-1}} \\
   \frac{\partial g}{\partial \theta_{t-1}} & \frac{\partial g}{\partial \theta_{t-1}} & \frac{\partial g}{\partial \theta_{t-1}} \\
   \frac{\partial g}{\partial \sigma_{t-1}} & \frac{\partial g}{\partial \sigma_{t-1}} & \frac{\partial g}{\partial \sigma_{t-1}}
   \end{pmatrix} \text{ Jacobian of } g \text{ w.r.t location} \\
   &2. \quad V_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t} = \begin{pmatrix}
   \frac{\partial g}{\partial u_t} \\
   \frac{\partial g}{\partial u_t} \\
   \frac{\partial g}{\partial u_t}
   \end{pmatrix} \text{ Jacobian of } g \text{ w.r.t control} \\
   &3. \quad M_t = \begin{pmatrix}
   \alpha_x u_t + \alpha_y v_t & 0 \\
   0 & \alpha_x u_t + \alpha_y v_t
   \end{pmatrix} \text{ Motion noise} \\
   &4. \quad \bar{\mu}_t = g(u_t, \mu_{t-1}) \quad \text{Predicted mean} \\
   &5. \quad \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T \quad \text{Predicted covariance}
   \end{align*}
   \]

2. **Correction:**

   \[
   \begin{align*}
   &1. \quad H_t = \frac{\partial h(\mu_{t-1})}{\partial \mu_{t-1}} = \begin{pmatrix}
   \frac{\partial h}{\partial \mu_{t-1}} \\
   \frac{\partial h}{\partial \mu_{t-1}} \\
   \frac{\partial h}{\partial \mu_{t-1}}
   \end{pmatrix} \text{ Jacobian of } h \text{ w.r.t location} \\
   &2. \quad \hat{z}_t = \frac{\partial h(\mu_{t-1})}{\partial \mu_{t-1}} = \begin{pmatrix}
   \frac{\partial h}{\partial \mu_{t-1}} \\
   \frac{\partial h}{\partial \mu_{t-1}} \\
   \frac{\partial h}{\partial \mu_{t-1}}
   \end{pmatrix} \text{ Predicted measurement mean} \\
   &3. \quad Q = \begin{pmatrix}
   \sigma^2_z & 0 \\
   0 & \sigma^2_z
   \end{pmatrix} \quad \text{Pred. measurement covariance} \\
   &4. \quad S_t = H_t \bar{\Sigma}_t H_t^T + Q_t \quad \text{Kalman gain} \\
   &5. \quad K_t = \bar{\Sigma}_t H_t^T S_t^{-1} \quad \text{Updated mean} \\
   &6. \quad \bar{\mu}_t = \hat{\mu}_t + K_t (z_t - \hat{z}_t) \quad \text{Updated covariance}
   \end{align*}
   \]
**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)**

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

**Unscented Transform**

Sigma points: $\chi^i = \mu$  
Weights: $w_0^i = \frac{\lambda}{n + \lambda}$  
$w_0^i = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta)$  
$w_i = \frac{1}{2(n + \lambda)}$ for $i = 1, \ldots, 2n$

Pass sigma points through nonlinear function $\psi^i = g(\chi^i)$

Recover mean and covariance  
$\mu' = \sum_{i=1}^{2n} w_i' \psi^i$  
$\Sigma' = \sum_{i=1}^{2n} w_i' (\psi^i - \mu')(\psi^i - \mu')^T$

**UKF Prediction Step**
UKF Observation Prediction Step

**UKF_predict** (μ_{t-1}, Σ_{t-1}, u_t, z_t):

Prediction:

\[ M_f = \begin{pmatrix} \alpha_i \xi_i | \alpha_i \xi_i | & 0 \\ 0 & 0 \end{pmatrix} \]

Motion noise

\[ Q_f = \begin{pmatrix} \xi_i^2 & 0 \\ 0 & \sigma_t^2 \end{pmatrix} \]

Measurement noise

\[ \mu_i = \begin{pmatrix} \mu_{x_i}^f \\ 0 \end{pmatrix} \]

Augmented state mean

\[ \Sigma_i = \begin{pmatrix} \Sigma_{x_i} & 0 \\ 0 & M_f \end{pmatrix} \]

Augmented covariance

\[ \chi_i = \begin{pmatrix} \mu_{x_i}^f + \sqrt{\Sigma_{x_i}} \xi_i \\ \mu_{x_i}^f - \sqrt{\Sigma_{x_i}} \xi_i \end{pmatrix} \]

Sigma points

\[ \bar{x}_f = g(\bar{x}_t + \chi_i, \chi_i^*) \]

Prediction of sigma points

\[ \bar{\mu} = \sum_{i=0}^{2^L} \omega_i \chi_i \]

Predicted mean

\[ \bar{\Sigma} = \sum_{i=0}^{2^L} \omega_i (\chi_i - \bar{\mu})(\chi_i - \bar{\mu})^T \]

Predicted covariance

UKF Correction Step

**UKF_correct** (μ_{t-1}, Σ_{t-1}, u_t, z_t):

Correction:

\[ Z_i = h(\bar{x}_f^i) + \chi_i^* \]

Measurement sigma points

\[ \hat{z}_i = \sum_{j=0}^{2^L} \omega_j Z_{ij} \]

Predicted measurement mean

\[ S_i = \sum_{j=0}^{2^L} \omega_j (Z_{ij} - \hat{z}_i)(Z_{ij} - \hat{z}_i)^T \]

Pred. measurement covariance

\[ \Sigma_{x_i} = \sum_{i=0}^{2^L} \omega_i \left( (Z_{ij} - \bar{\mu})S_{ij}^{-1} \right) \]

Cross-covariance

\[ K_i = \Sigma_{x_i} \Sigma_{z_i}^{-1} \]

Kalman gain

\[ \mu_t = \bar{\mu} + K_i (z_t - \hat{z}_t) \]

Updated mean

\[ \Sigma_t = \bar{\Sigma} - K_i S_i K_i^T \]

Updated covariance
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)

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:
  \[ H_i = \{ \hat{x}_i, \Sigma_i, P(H_i) \} \]

- 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]