CSE-571
Probabilistic Robotics

Kalman Filters

Dieter Fox
Bayes Filter Reminder

- **Prediction**

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

- **Correction**

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

Univariate

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

Multivariate

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

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

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

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

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

- We stay in the “Gaussian world” as long as we start with Gaussians and perform only linear transformations.
Discrete Kalman Filter

Estimates the state $x$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

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
Kalman Filter Updates in 1D

\[
bel(x_t) = \begin{cases} 
\mu_t = \bar{\mu}_t + K_t (z_t - \bar{\mu}_t) \\
\sigma_t^2 = (1 - K_t) \bar{\sigma}_t^2 
\end{cases}
\]

with \( K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \sigma_{obs,t}^2} \)

\[
bel(x_t) = \begin{cases} 
\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) \\
\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t 
\end{cases}
\]

with \( K_t = \frac{\bar{\Sigma}_t}{\bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}} \)
Kalman Filter Updates in 1D

$$\text{bel}(x_t) = \begin{cases} 
\overline{\mu}_t = a_t \mu_{t-1} + b_t u_t \\
\overline{\sigma}^2_t = a_t^2 \sigma^2_t + \sigma^2_{act,t}
\end{cases}$$

$$\text{bel}(x_t) = \begin{cases} 
\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t \\
\Sigma_t = A_t \Sigma_{t-1} A^T_t + R_t
\end{cases}$$
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:

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

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

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

\[ \sim N(x_t; A_t x_{t-1} + B_t u_t, R_t) \quad \sim N(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}) \]
Linear Gaussian Systems: Dynamics

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

\[
\downarrow
\]

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

\[
\downarrow
\]

\[
\overline{\text{bel}}(x_t) = \eta \int \exp \left\{ -\frac{1}{2} (x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t) \right\} \exp \left\{ -\frac{1}{2} (x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - \mu_{t-1}) \right\} \, dx_{t-1}
\]

\[
\overline{\text{bel}}(x_t) = \begin{cases} 
\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t \\
\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t
\end{cases}
\]
Linear Gaussian Systems: Observations

- Observations are linear function of state plus additive noise:

\[ z_t = C_t x_t + \delta_t \]

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

\[ bel(x_t) = \eta \ p(z_t \mid x_t) \]

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

\[ bel(x_t) \]

\[ \sim N(x_t; \mu_t, \Sigma_t) \]
Linear Gaussian Systems: Observations

\[
\begin{aligned}
bel(x_t) &= \eta \ p(z_t \mid x_t) \\
&\downarrow \\
&\sim N(z_t; C_t x_t, Q_t) \\
&\downarrow \\
&\sim N(x_t; \mu_t, \Sigma_t) \\
&\downarrow \\
bel(x_t) &= \eta \exp \left\{ -\frac{1}{2} (z_t - C_t x_t)^T Q_t^{-1} (z_t - C_t x_t) \right\} \exp \left\{ -\frac{1}{2} (x_t - \mu_t)^T \Sigma_t^{-1} (x_t - \mu_t) \right\}
\end{aligned}
\]

\[
\begin{aligned}
bel(x_t) &= \mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) \\
\Sigma_t &= (I - K_tC_t) \bar{\Sigma}_t \\
&\text{with} \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}
\end{aligned}
\]
Kalman Filter Algorithm

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

2. Prediction:

3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$

4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. Correction:

6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$

7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$

8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

9. Return $\mu_t, \Sigma_t$
The Prediction-Correction-Cycle

\[ \begin{align*}
\overline{bel}(x_t) &= \begin{cases} 
\bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\
\bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2
\end{cases} \\
\overline{bel}(x_t) &= \begin{cases} 
\mu_t = A_t \mu_{t-1} + B_t u_t \\
\Sigma_t = A_t \Sigma_{t-1} A_t^T + R_t
\end{cases}
\end{align*} \]
The Prediction-Correction-Cycle

\[ \text{bel}(x_i) = \left\{ \begin{array}{l} \mu_i = \bar{\mu}_i + K_t(z_t - \bar{\mu}_i), \\ \sigma_i^2 = (1 - K_t)\tilde{\sigma}_i^2 \\ \end{array} \right. \]

\[ K_t = \frac{\tilde{\sigma}_i^2}{\sigma_i^2 + \sigma_{\text{obs}}^2} \]

\[ \text{bel}(x_i) = \left\{ \begin{array}{l} \mu_i = \bar{\mu}_i + K_t(z_t - C_i\bar{\mu}_i), \\ \Sigma_i = (I - K_tC_i)\tilde{\Sigma}_i \\ \end{array} \right. \]

\[ K_t = \frac{\tilde{\Sigma}_i C_i^T (C_i \Sigma_i C_i^T + Q_t)^{-1}}{\sigma^2} \]
The Prediction-Correction-Cycle

\[ bel(x_t) = \left\{ \begin{array}{l}
\mu_t = \overline{\mu}_t + K_t(z_t - \overline{\mu}_t) \\
\sigma^2_t = (1 - K_t)\overline{\sigma}^2_t
\end{array} \right. \quad , \quad K_t = \frac{\overline{\sigma}^2_t}{\overline{\sigma}^2_t + \sigma^2_{\text{obs},t}} \]

\[ bel(x_t) = \left\{ \begin{array}{l}
\mu_t = \overline{\mu}_t + K_t(z_t - C_t\overline{\mu}_t) \\
\Sigma_t = (I - K_tC_t)\overline{\Sigma}_t
\end{array} \right. \quad , \quad K_t = \overline{\Sigma}_tC_t^T(C_t\overline{\Sigma}_tC_t^T + Q_t)^{-1} \]

\[ \overline{bel}(x_t) = \left\{ \begin{array}{l}
\overline{\mu}_t = a_t\mu_{t-1} + b_tu_t \\
\overline{\sigma}^2_t = a_t^2\sigma^2_t + \sigma^2_{\text{act},t}
\end{array} \right. \]

\[ \overline{bel}(x_t) = \left\{ \begin{array}{l}
\overline{\mu}_t = A_t\mu_{t-1} + B_tu_t \\
\overline{\Sigma}_t = A_t\Sigma_{t-1}A_t^T + R_t
\end{array} \right. \]
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**!
Going non-linear

EXTENDED KALMAN FILTER
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

- Grey shaded area: $p(y)$
- Blue line: Gaussian of $p(y)$
- Cross: Mean of $p(y)$

- Blue line: Function $g(\mu)$
- Triangle: Mean $\mu$
- Circle: $g(\mu)$

- Grey shaded area: $p(\mu)$
- Cross: Mean $\mu$
EKF Linearization (1)
EKF Linearization (2)
EKF Linearization (3)
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:**

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

\[
h(x_t) \approx h(\bar{\mu}_t) + H_t (x_t - \bar{\mu}_t)
\]
EKF Algorithm

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

2. Prediction:

3. $\overline{\mu}_t = g(u_t, \mu_{t-1})$

4. $\overline{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$

3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$

4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. Correction:

6. $K_t = \overline{\Sigma}_t H_t^T (H_t \overline{\Sigma}_t H_t^T + Q_t)^{-1}$

7. $\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$

8. $\Sigma_t = (I - K_t H_t) \overline{\Sigma}_t$

9. $\Sigma_t = (I - K_t C_t) \overline{\Sigma}_t$

9. Return $\mu_t, \Sigma_t$

$$H_t = \frac{\partial h(\overline{\mu}_t)}{\partial x_t} \quad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$$
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)
Landmark-based Localization
1. **EKF_localization** \((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\):

**Prediction:**

\[
G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{pmatrix}
\frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \mu_{t-1,\theta}} \\
\frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \mu_{t-1,\theta}} \\
\frac{\partial \theta'}{\partial \mu_{t-1,x}} & \frac{\partial \theta'}{\partial \mu_{t-1,y}} & \frac{\partial \theta'}{\partial \mu_{t-1,\theta}}
\end{pmatrix}
\]

Jacobian of \(g\) w.r.t location

\[
V_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t} = \begin{pmatrix}
\frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial \omega_t} \\
\frac{\partial y'}{\partial v_t} & \frac{\partial y'}{\partial \omega_t} \\
\frac{\partial \theta'}{\partial v_t} & \frac{\partial \theta'}{\partial \omega_t}
\end{pmatrix}
\]

Jacobian of \(g\) w.r.t control

\[
M_t = \begin{pmatrix}
\alpha_1 v_t^2 + \alpha_2 \omega_t^2 & 0 \\
0 & \alpha_3 v_t^2 + \alpha_4 \omega_t^2
\end{pmatrix}
\]

Motion noise

\[
\bar{\mu}_t = g(u_t, \mu_{t-1})
\]

Predicted mean

\[
\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T
\]

Predicted covariance
EKF Prediction Step
1. **EKF_localization** \((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\):

   **Correction:**

   2. \[
   \hat{z}_t = \left( \sqrt{(m_x - \mu_{t,x})^2 + (m_y - \mu_{t,y})^2} \right) \text{ atan} \left( \frac{m_y - \mu_{t,y}}{m_x - \mu_{t,x}} \right) - \mu_{t, \theta}
   \]

   Predicted measurement mean

   3. \[
   H_t = \frac{\partial h(\mu_t, m)}{\partial x_i} = \begin{bmatrix}
   \frac{\partial r_i}{\partial \mu_{t,x}} & \frac{\partial r_i}{\partial \mu_{t,y}} & \frac{\partial r_i}{\partial \mu_{t,\theta}} \\
   \frac{\partial \phi_t}{\partial \mu_{t,x}} & \frac{\partial \phi_t}{\partial \mu_{t,y}} & \frac{\partial \phi_t}{\partial \mu_{t,\theta}}
   \end{bmatrix}
   \]

   Jacobian of \(h\) w.r.t location

   4. \[
   Q_t = \begin{bmatrix}
   \sigma_r^2 & 0 \\
   0 & \sigma_r^2
   \end{bmatrix}
   \]

   Pred. measurement covariance

   5. \[
   S_t = H_t \Sigma_t H_t^T + Q_t
   \]

   Pred. measurement covariance

   6. \[
   K_t = \Sigma_t H_t^T S_t^{-1}
   \]

   Kalman gain

   7. \[
   \mu_t = \bar{\mu}_t + K_t (z_t - \hat{z}_t)
   \]

   Updated mean

   8. \[
   \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t
   \]

   Updated covariance
EKF Observation Prediction Step
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!
Going unscented

UNSCENTED KALMAN FILTER
Linearization via Unscented Transform
UKF Sigma-Point Estimate (2)

EKF

UKF
UKF Sigma-Point Estimate (3)

EKF

UKF
Unscented Transform

Sigma points

\[ \chi^0 = \mu \]
\[ \chi^i = \mu \pm \left( \sqrt{(n+\lambda)\Sigma} \right)_i \]

Weights

\[ w^0_m = \frac{\lambda}{n+\lambda} \]
\[ w^0_c = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \]
\[ w^i_m = w^i_c = \frac{1}{2(n+\lambda)} \] for \( i = 1, \ldots, 2n \)

Pass sigma points through nonlinear function

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

Recover mean and covariance

\[ \mu' = \sum_{i=0}^{2n} w^i_m \psi^i \]
\[ \Sigma' = \sum_{i=0}^{2n} w^i_c (\psi^i - \mu)(\psi^i - \mu)^T \]
UKF\_predict ( \( \mu_{t-1}, \Sigma_{t-1}, u_t, z_t \) ):

**Prediction:**

\[
M_t = \begin{pmatrix}
(\alpha_1 \mid v_t \mid + \alpha_2 \mid \omega_t \mid)^2 & 0 \\
0 & (\alpha_3 \mid v_t \mid + \alpha_4 \mid \omega_t \mid)^2 \\
\end{pmatrix}
\]

Motion noise

\[
Q_t = \begin{pmatrix}
\sigma_r^2 & 0 \\
0 & \sigma_r^2 \\
\end{pmatrix}
\]

Measurement noise

\[
\mu_{t-1}^a = \begin{pmatrix}
\mu_{t-1}^T \\
(0 0)^T \\
(0 0)^T \\
\end{pmatrix}
\]

Augmented state mean

\[
\Sigma_{t-1}^a = \begin{pmatrix}
\Sigma_{t-1} & 0 & 0 \\
0 & M_t & 0 \\
0 & 0 & Q_t \\
\end{pmatrix}
\]

Augmented covariance

\[
\chi_{t-1}^a = \begin{pmatrix}
\mu_{t-1}^a \\
\mu_{t-1}^a + \gamma \sqrt{\Sigma_{t-1}^a} \\
\mu_{t-1}^a - \gamma \sqrt{\Sigma_{t-1}^a} \\
\end{pmatrix}
\]

Sigma points

\[
\overline{\chi}_t^x = g(u_t + \chi_t^u, \chi_{t-1}^x)
\]

Prediction of sigma points

\[
\overline{\mu}_t = \sum_{i=0}^{2L} w_m^i \chi_{i,t}^x
\]

Predicted mean

\[
\overline{\Sigma}_t = \sum_{i=0}^{2L} w_c^i \left( \chi_{i,t}^x - \overline{\mu}_t \right) \left( \chi_{i,t}^x - \overline{\mu}_t \right)^T
\]

Predicted covariance
**UKF_correct** $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

**Correction:**

$$\bar{Z}_t = h(\chi^x_t) + \chi^z_t$$  \hspace{1cm} \text{Measurement sigma points} \\

$$\hat{Z}_t = \sum_{i=0}^{2L} w^i_m \bar{Z}_{i,t}$$  \hspace{1cm} \text{Predicted measurement mean} \\

$$S_t = \sum_{i=0}^{2L} w^i_c \left( \bar{Z}_{i,t} - \hat{Z}_t \right) \left( \bar{Z}_{i,t} - \hat{Z}_t \right)^T$$  \hspace{1cm} \text{Pred. measurement covariance} \\

$$\Sigma^{x,z}_t = \sum_{i=0}^{2L} w^i_c \left( \bar{\chi}^x_{i,t} - \bar{\mu}_t \right) \left( \bar{Z}_{i,t} - \hat{Z}_t \right)^T$$  \hspace{1cm} \text{Cross-covariance} \\

$$K_t = \Sigma^{x,z}_t S_t^{-1}$$  \hspace{1cm} \text{Kalman gain} \\

$$\mu_t = \bar{\mu}_t + K_t (z_t - \hat{Z}_t)$$  \hspace{1cm} \text{Updated mean} \\

$$\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$$  \hspace{1cm} \text{Updated covariance}
UKF Prediction Step
UKF Observation Prediction Step
UKF Correction Step
EKF Correction Step
Estimation Sequence

EKF                    PF                    UKF
Estimation Sequence

EKF

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)

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

Robot view

Pose candidates

Sensor data

Feature extraction

Generate pose candidates

MATCH existing? NO

Update hypothesis

Creative feature? YES

Create hypothesis

Courtesy of P. Jensfelt and S. Kristensen
MHT: Implemented System (3)
Example run

Map and trajectory

# hypotheses vs. time

Courtesy of P. Jensfelt and S. Kristensen