CSE-571
AI-based Mobile Robotics

Mobile Robot Localization

Kalman filters
Discrete
Particle filters

Localization

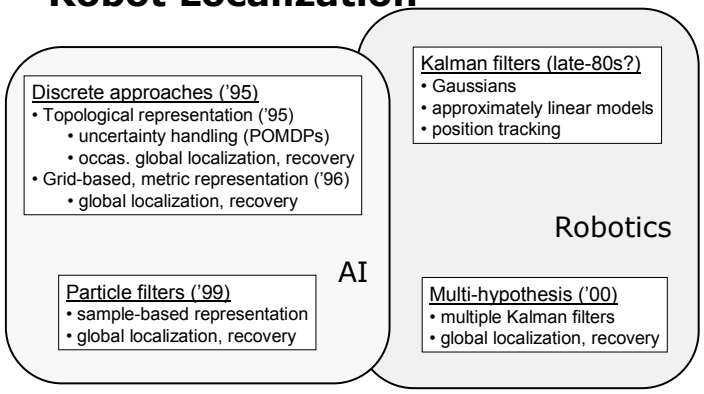
"Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities." [Cox '91]

- **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)

Bayes Filter Algorithm

1. Algorithm **Bayes_filter**(*Bel(x),d*):
2. $\eta=0$
3. If *d* is a perceptual data item *z* then
4. For all *x* do
5. $Bel'(x) = P(z \mid x) Bel(x)$
6. $\eta = \eta + Bel'(x)$
7. For all *x* do
8. $Bel'(x) = \eta^{-1} Bel'(x)$
9. Else if *d* is an action data item *u* then
10. For all *x* do
11. $Bel'(x) = \int P(x \mid u, x') \; Bel(x') \; dx'$
12. Return *Bel'(x)*

Representations for Bayesian Robot Localization

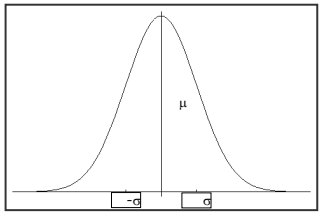


Gaussians

$$p(x) \sim N(\mu,\sigma^2):$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

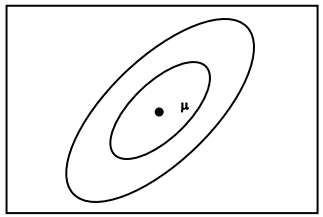
Univariate



$$p(\mathbf{x}) \sim N(\boldsymbol{\mu},\boldsymbol{\Sigma}):$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

Multivariate



Properties of Gaussians

$$\left.\begin{array}{l} X \sim N(\mu,\sigma^2) \\ Y = aX + b \end{array}\right\} \Rightarrow \quad Y \sim N(a\mu + b, a^2\sigma^2)$$

$$\left.\begin{array}{l} X_1 \sim N(\mu_1,{\sigma_1}^2) \\ X_2 \sim N(\mu_2,{\sigma_2}^2) \end{array}\right\} \Rightarrow \begin{array}{c} X_1 + X_2 \sim N\left(\mu_1+\mu_2, {\sigma_1}^2+{\sigma_2}^2\right) \\ X_1 \cdot X_2 \sim N\left(\frac{{\sigma_2}^2}{{\sigma_1}^2+{\sigma_2}^2}\mu_1 + \frac{{\sigma_1}^2}{{\sigma_1}^2+{\sigma_2}^2}\mu_2,\ \frac{1}{{\sigma_1}^{-2}+{\sigma_2}^{-2}}\right) \end{array}$$

Same holds for multivariate Gaussians

- We stay in the "Gaussian world" as long as we start with Gaussians and perform only linear transformations.

Observations are linear in state with independent noise $\varepsilon_{\sigma^2_{obs}} = N\left(0,\sigma^2_{obs}\right)$

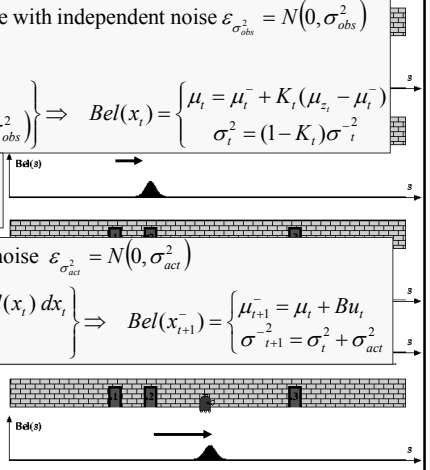
$$\left.\begin{array}{l} Bel(x_t) = P(z_t \mid x_t) Bel(x_t^-) \\ Z_t = CX_t + \varepsilon_{\sigma^2_{obs}}, Z_t \sim N\left(\mu_{z_t},\sigma^2_{obs}\right) \end{array}\right\} \Rightarrow \quad Bel(x_t) = \left\{\begin{array}{l} \mu_t = \mu_t^- + K_t(\mu_{z_t} - \mu_t^-) \\ \sigma_t^2 = (1-K_t)\sigma_t^{-2} \end{array}\right.$$

Kalman gain $K_t = \dfrac{\sigma_t^{-2}}{\sigma_t^{-2} + \sigma^2_{obs}}$

Linear dynamics with white noise $\varepsilon_{\sigma^2_{act}} = N\left(0,\sigma^2_{act}\right)$

$$\left.\begin{array}{l} Bel(x_{t+1}^-) = \int P(x_{t+1}^- \mid u_t, x_t) Bel(x_t)\, dx_t \\ X_{t+1}^- = AX_t + Bu_t + \varepsilon_{\sigma^2_{act}} \end{array}\right\} \Rightarrow \quad Bel(x_{t+1}^-) = \left\{\begin{array}{l} \mu_{t+1}^- = \mu_t + Bu_t \\ \sigma_{t+1}^{-2} = \sigma_t^2 + \sigma^2_{act} \end{array}\right.$$

$$Bel(x_{t+2}^-) = \left\{\begin{array}{l} \mu_{t+2}^- = \mu_{t+1} + Bu_{t+1} \\ \sigma_{t+2}^{-2} = \sigma_{t+1}^2 + \sigma^2_{act} \end{array}\right.$$



Multivariate Observations

Observations are linear in state with independent noise $\varepsilon_{\sigma^2_{obs}} = N\left(0,\sigma^2_{obs}\right)$

$$\left.\begin{array}{l} Bel(x_t) = P(z_t \mid x_t) Bel(x_t^-) \\ Z_t = CX_t + \varepsilon_{\sigma^2_{obs}}, Z_t \sim N\left(\mu_{z_t},\sigma^2_{obs}\right) \end{array}\right\} \Rightarrow \quad Bel(x_t) = \left\{\begin{array}{l} \mu_t = \mu_t^- + K_t(\mu_{z_t} - \mu_t^-) \\ \sigma_t^2 = (1-K_t)\sigma_t^{-2} \end{array}\right.$$

Kalman gain $K_t = \dfrac{\sigma_t^{-2}}{\sigma_t^{-2} + \sigma^2_{obs}}$

Observations are linear in state with independent noise $\varepsilon_{\Sigma_{obs}} = N\left(0,\Sigma_{obs}\right)$

$$\left.\begin{array}{l} Bel(x_t) = P(z_t \mid x_t) Bel(x_t^-) \\ Z_t = CX_t + \varepsilon_{\Sigma_{obs}}, Z_t \sim N\left(\mu_{z_t},\Sigma_{obs}\right) \end{array}\right\} \Rightarrow \quad Bel(x_t) = \left\{\begin{array}{l} \mu_t = \mu_t^- + K_t(z_t - C\mu_t^-) \\ \Sigma_t = (I - K_tC)\Sigma_t^- \end{array}\right.$$

Kalman gain $K_t = \Sigma_t^- C^T \left(C\Sigma_t^- C^T + \Sigma_{obs}\right)^{-1}$

Multivariate Dynamics

Linear dynamics with white noise $\varepsilon_{\sigma^2_{act}} = N(0, \sigma^2_{act})$

$$\left.\begin{aligned} Bel(x^-_{t+1}) &= \int P(x^-_{t+1} \mid u_t, x_t) Bel(x_t)\, dx_t \\ X^-_{t+1} &= AX_t + Bu_t + \varepsilon_{\sigma^2_{act}} \end{aligned}\right\} \Rightarrow \quad Bel(x^-_{t+1}) = \begin{cases} \mu^-_{t+1} = \mu_t + Bu_t \\ \sigma^{-2}_{t+1} = \sigma^2_t + \sigma^2_{act} \end{cases}$$

Linear dynamics with white noise $\varepsilon_{\Sigma_{act}} = N(0, \Sigma_{act})$

$$\left.\begin{aligned} Bel(x^-_{t+1}) &= \int P(x^-_{t+1} \mid u_t, x_t) Bel(x_t)\, dx_t \\ X^-_{t+1} &= AX_t + Bu_t + \varepsilon_{\Sigma_{act}} \end{aligned}\right\} \Rightarrow \quad Bel(x^-_{t+1}) = \begin{cases} \mu^-_{t+1} = A\mu_t + Bu_t \\ \Sigma^-_{t+1} = A\Sigma_t A^T + \Sigma_{act} \end{cases}$$

Kalman Filter Algorithm

1. Algorithm **Kalman_filter**(<μ,Σ>, *d*):
2. If *d* is a perceptual data item *z* then
3. $K = \Sigma C^T \left(C\Sigma C^T + \Sigma_{obs}\right)^{-1}$
4. $\mu = \mu + K(z - C\mu)$
5. $\Sigma = (I - KC)\Sigma$
6. Else if *d* is an action data item *u* then
7. $\mu = A\mu + Bu$
8. $\Sigma = A\Sigma A^T + \Sigma_{act}$
9. Return <μ,Σ>

Limitations

- Very strong assumptions:
  - Linear state dynamics $X^-_{t+1} = AX_t + Bu_t + \Sigma_{act}$
  - Observations linear in state $Z_t = CX_t + \Sigma_{obs}$
- What can we do if system is not linear?
  - Non-linear state dynamics $X^-_{t+1} = f(X_t, u_t, \Sigma_{act})$
  - Non-linear observations $Z_t = c(X_t, \Sigma_{obs})$
- Linearize it!
  - Determine Jacobians of dynamics *f* and observation function *c* w.r.t the current state *x* and the noise.

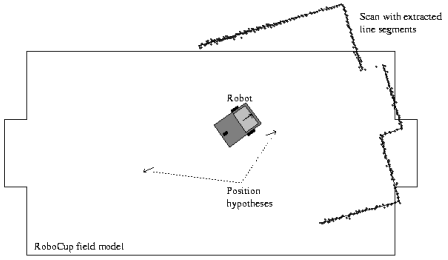
$$A_{ij} = \frac{\partial f_i}{\partial x_j}(x_t, u_t, 0) \qquad C_{ij} = \frac{\partial c_i}{\partial x_j}(x_t, 0) \qquad W_{ij} = \frac{\partial f_i}{\partial \Sigma_{act\,j}}(x_t, u_t, 0) \qquad V_{ij} = \frac{\partial c_i}{\partial \Sigma_{obs\,j}}(x_t, 0)$$

Extended Kalman Filter Algorithm

1. Algorithm **Extended_Kalman_filter**(<μ,Σ>, *d*):
2. If *d* is a perceptual data item *z* then
3. $K = \Sigma C^T \left(C\Sigma C^T + V\Sigma_{obs} V^T\right)^{-1}$ ⟵ $K = \Sigma C^T \left(C\Sigma C^T + \Sigma_{obs}\right)^{-1}$
4. $\mu = \mu + K\left(z - c(\mu, 0)\right)$ ⟵ $\mu = \mu + K(z - C\mu)$
5. $\Sigma = (I - KC)\Sigma$ ⟵ $\Sigma = (I - KC)\Sigma$
6. Else if *d* is an action data item *u* then
7. $\mu = f(\mu, u, 0)$ ⟵ $\mu = A\mu + Bu$
8. $\Sigma = A\Sigma A^T + W\Sigma_{act} W^T$ ⟵ $\Sigma = A\Sigma A^T + \Sigma_{act}$
9. Return <μ,Σ>

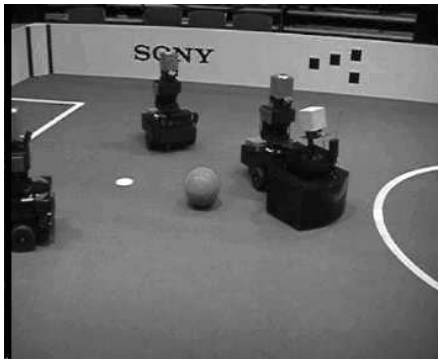
Kalman Filter Based Systems (1)

- [Gutmann et al. 96, 98]:
  - Match LRF scans against map
  - Highly successful in RoboCup mid-size league



Courtesy of S. Gutmann

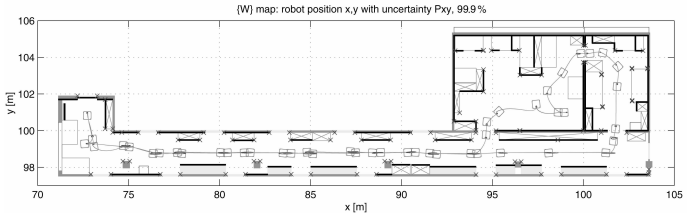
Kalman Filter Based Systems (1)



Courtesy of S. Gutmann

Kalman Filter Based Systems (2)

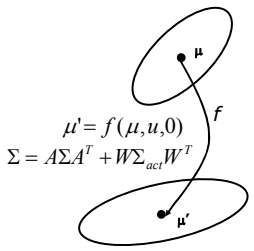
- [Arras et al. 98]:
  - Laser range-finder and vision
  - High precision (<1cm accuracy)

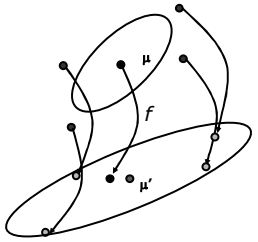


Courtesy of K. Arras

Unscented Kalman Filter

- Instead of linearizing, pass several points from the Gaussian through the non-linear transformation and re-compute a new Gaussian.
- Better performance (theory and practice).

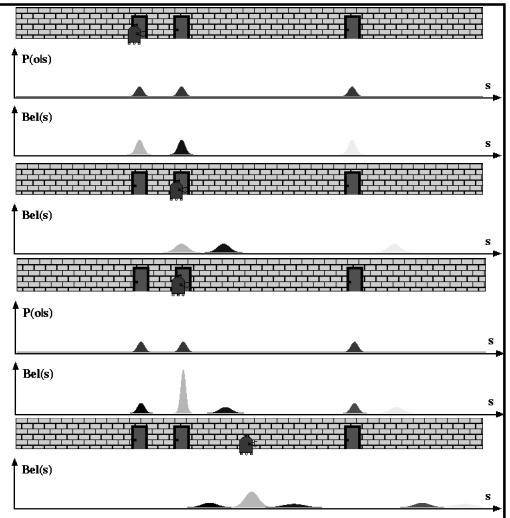




Localization Algorithms - Comparison

	Kalman filter				
Sensors	Gaussian				
Posterior	Gaussian				
Efficiency (memory)	++				
Efficiency (time)	++				
Implementation	+				
Accuracy	++				
Robustness	-				
Global localization	No				

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

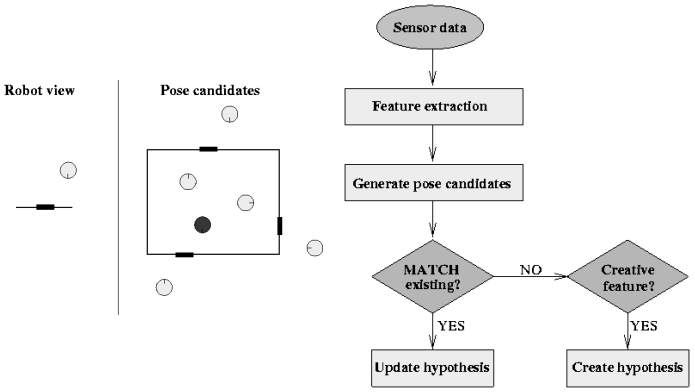
$$P(H_i \mid s) = \frac{P(s \mid H_i) P(H_i)}{P(s)}$$

- Hypotheses with low probability are deleted.
- New candidates are extracted from LRF scans.

$$C_j = \{z_j, R_j\}$$

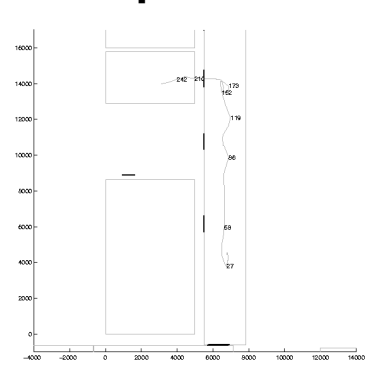
[Jensfelt et al. '00]

MHT: Implemented System (2)

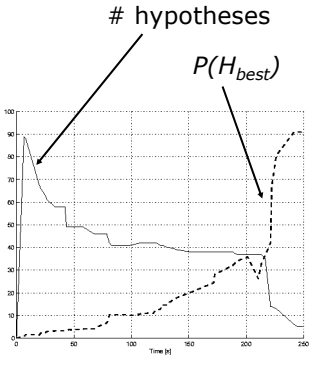


Courtesy of P. Jensfelt and S. Kristensen

MHT: Implemented System (3) Example run



Map and trajectory



#hypotheses vs. time

Courtesy of P. Jensfelt and S. Kristensen

Localization Algorithms - Comparison

	Kalman filter	Multi-hypothesis tracking			
Sensors	Gaussian	Gaussian			
Posterior	Gaussian	Multi-modal			
Efficiency (memory)	++	++			
Efficiency (time)	++	++			
Implementation	+	o			
Accuracy	++	++			
Robustness	-	+			
Global localization	No	Yes			