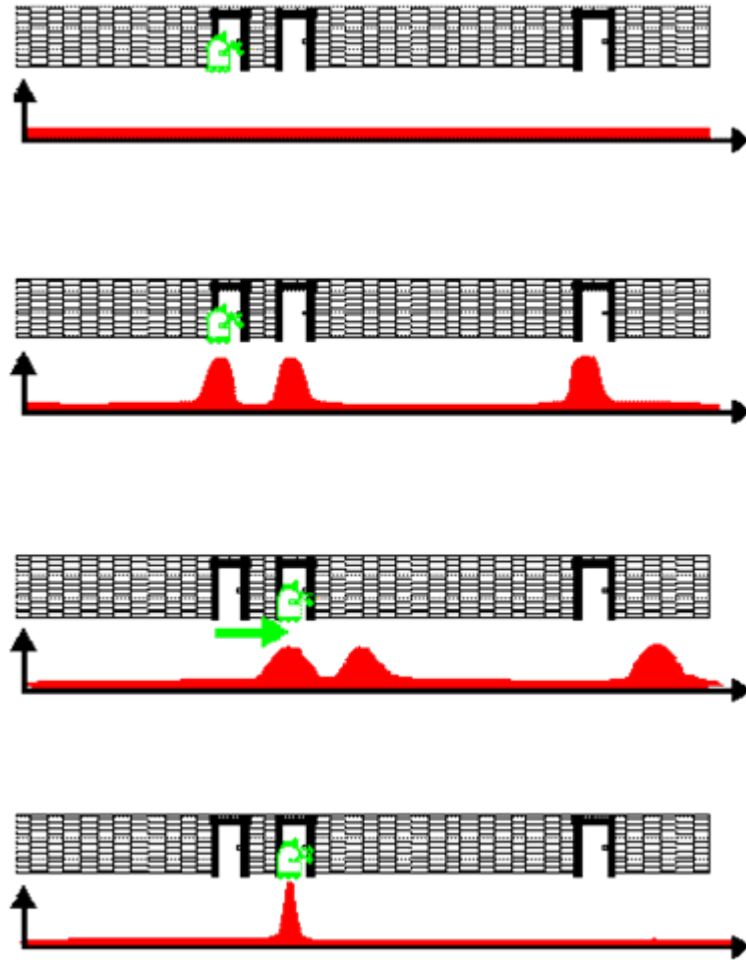


Probabilistic Robotics: Extended Kalman Filters

Arnoud Visser
Universiteit van Amsterdam

Slide credits: Sebastian Thrun, Alex Teichman, Wolfram Burgard, Dieter Fox, Cyrill Stachniss, Giorgio Grisetti, Maren Bennewitz, Christian Plagemann, Dirk Haehnel, Mike Montemerlo, Nick Roy, Kai Arras, Patrick Pfaff and others

Bayes Filters in Localization



$$Bel(x_t) = \eta P(z_t | x_t) \int P(x_t | u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

Bayes Filter Reminder

- Prediction

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

- Measurement Update

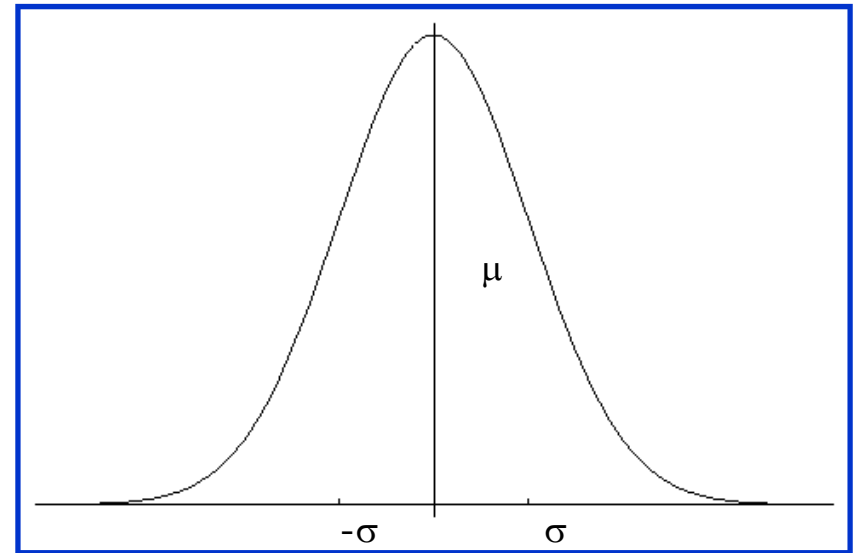
$$bel(x_t) = \eta p(z_t | x_t) \overline{bel}(x_t)$$

Gaussians

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

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

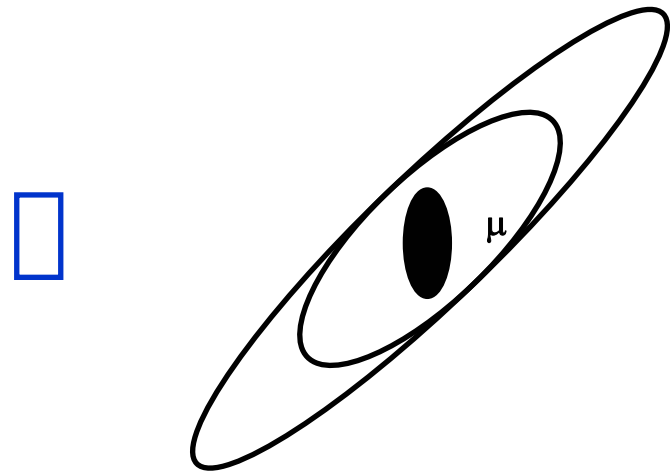
Univariate



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

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

Multivariate



Properties of Gaussians

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

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

Multivariate Gaussians

$$\left. \begin{array}{l} X \sim N(\mu, \Sigma) \\ Y = AX + B \end{array} \right\} \Rightarrow Y \sim N(A\mu + B, A\Sigma A^T)$$

$$\left. \begin{array}{l} X_1 \sim N(\mu_1, \Sigma_1) \\ X_2 \sim N(\mu_2, \Sigma_2) \end{array} \right\} \Rightarrow p(X_1) \cdot p(X_2) \sim N\left(\frac{\Sigma_1^{-1}}{\Sigma_1^{-1} + \Sigma_2^{-1}}\mu_1 + \frac{\Sigma_2^{-1}}{\Sigma_1^{-1} + \Sigma_2^{-1}}\mu_2, \frac{1}{\Sigma_1^{-1} + \Sigma_2^{-1}}\right)$$

- 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

$$z_t = C_t x_t + \delta_t$$

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:

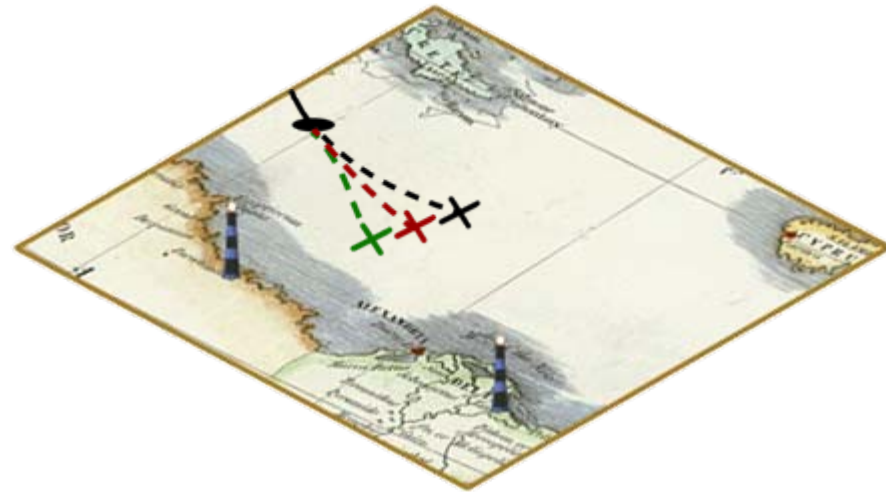
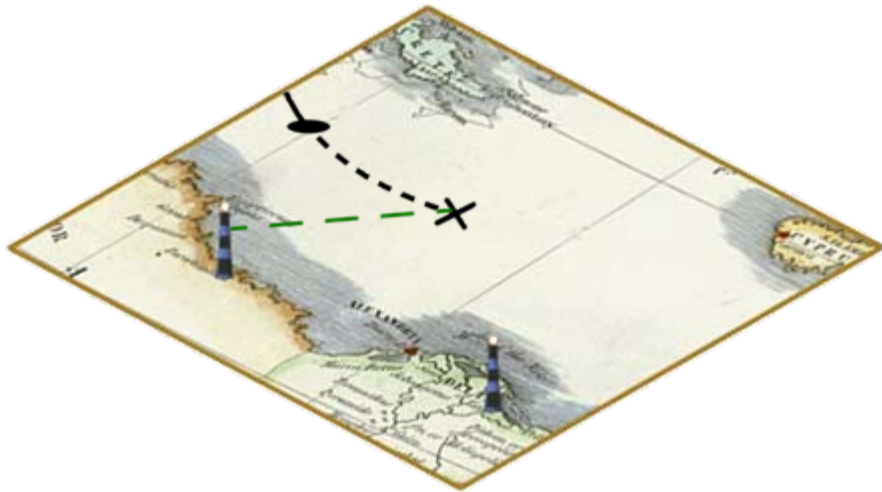
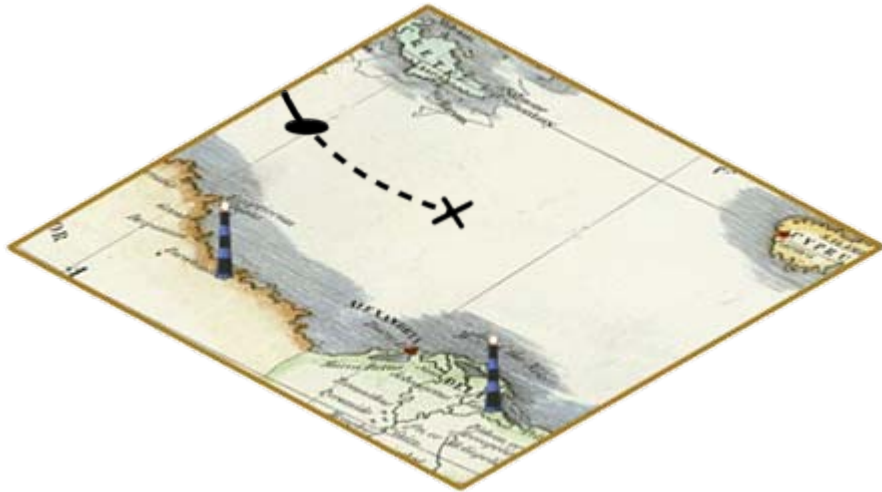
6.
$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

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 μ_t, Σ_t

Kalman Filter Algorithm



Kalman Filter Algorithm

- Prediction

$$\hat{\mathbf{x}}(k+1|k) = f(\hat{\mathbf{x}}(k|k), \mathbf{u}(k+1))$$

$$\mathbf{P}(k+1|k) = \nabla f_x \mathbf{P}(k|k) \nabla f_x^T + \nabla f_u \mathbf{U}(k+1) \nabla f_u^T$$

- Observation

$$z(k+1), \mathbf{R}(k+1)$$

- Matching

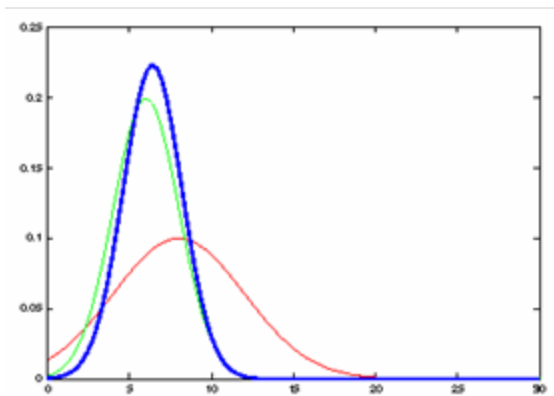
$$\mathbf{v}_{ij}(k+1) = z_i(k+1) - h(\hat{\mathbf{x}}(k+1|k), m_j)$$

- Correction

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\mathbf{v}(k+1)$$

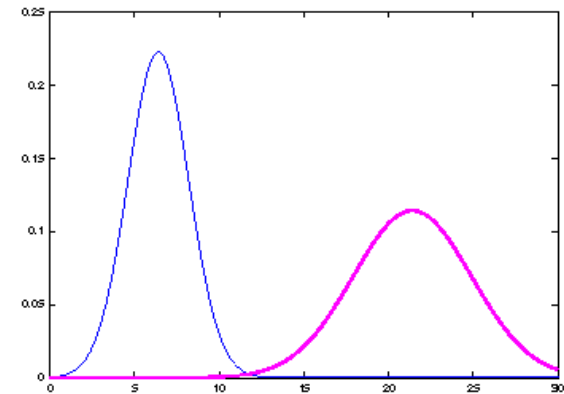
$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1)$$

The Prediction-Correction-Cycle

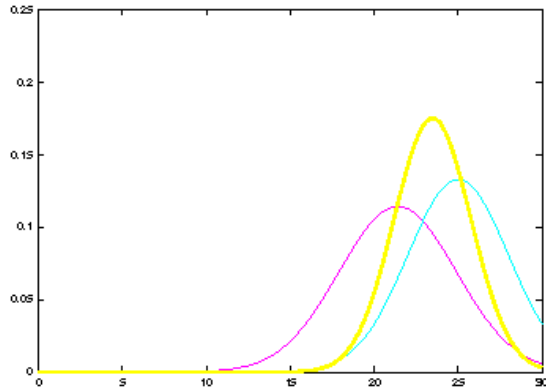


$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_{t-1}^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

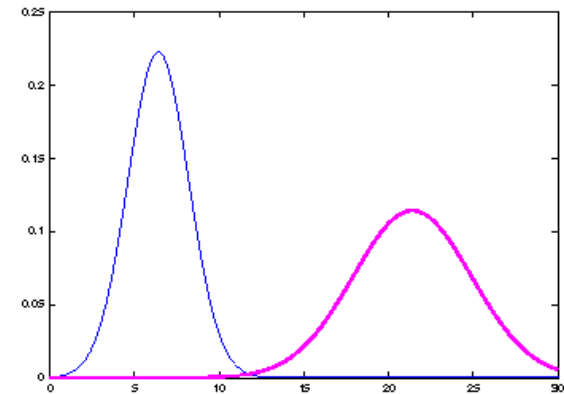


The Prediction-Correction-Cycle



$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2, K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2} \end{cases}$$

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t) \\ \Sigma_t = (I - K_tC_t)\bar{\Sigma}_t, K_t = \bar{\Sigma}_tC_t^T(C_t\bar{\Sigma}_tC_t^T + Q_t)^{-1} \end{cases}$$



Correction

The Prediction-Correction-Cycle



$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2 \end{cases}, K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

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

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \end{cases}, K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$



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!**

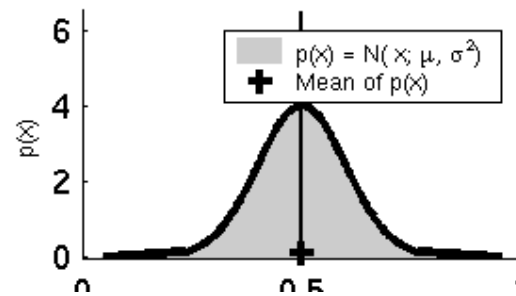
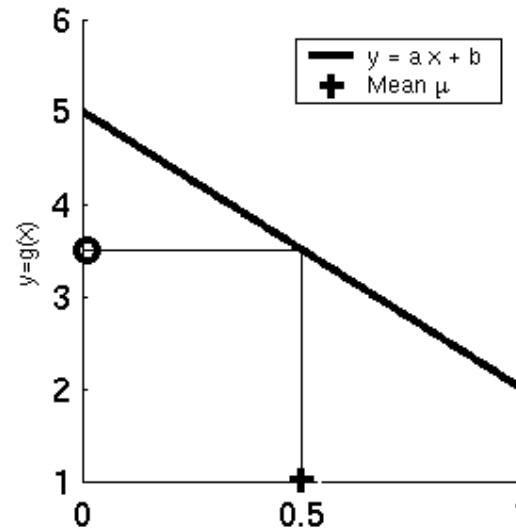
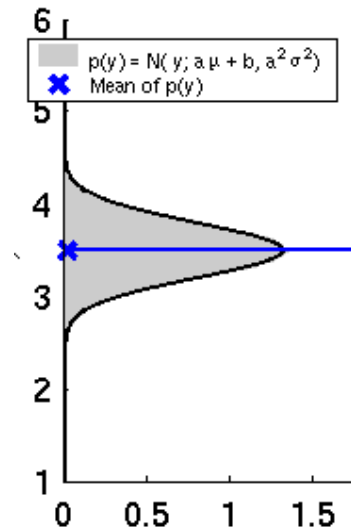
Nonlinear Dynamic Systems

- Most realistic robotic problems involve nonlinear functions

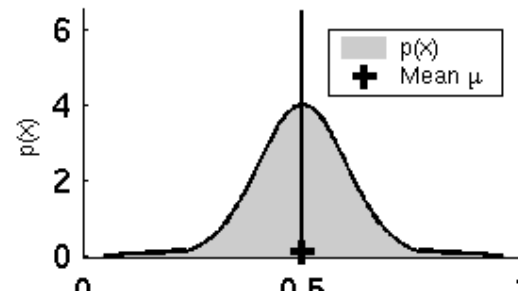
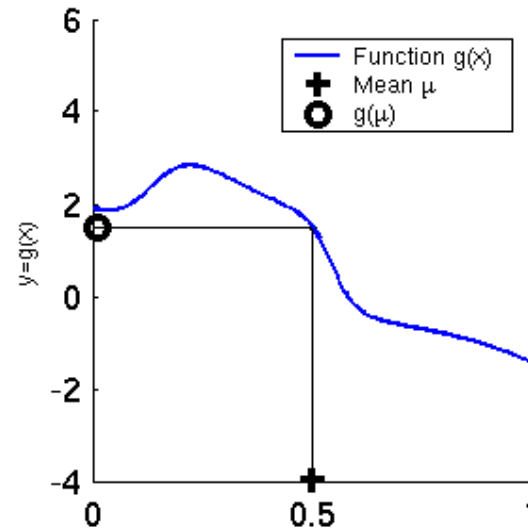
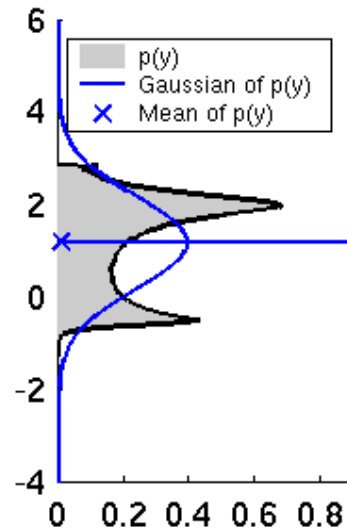
$$x_t = g(u_t, x_{t-1})$$

$$z_t = h(x_t)$$

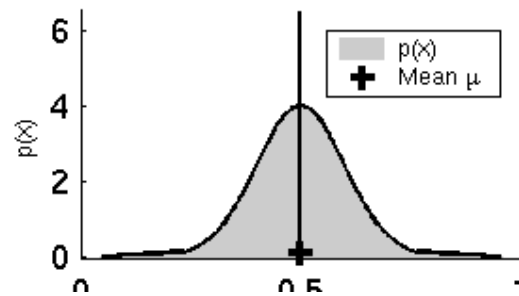
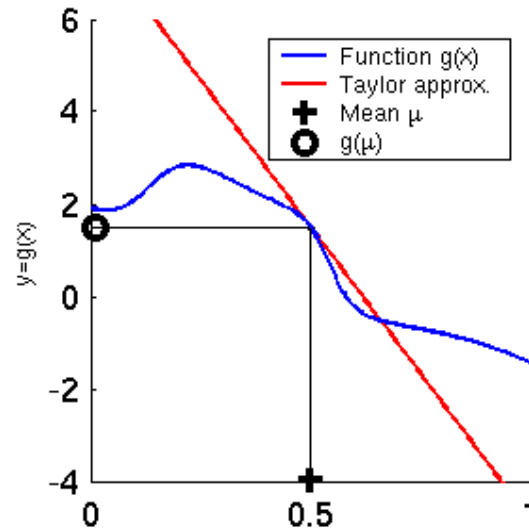
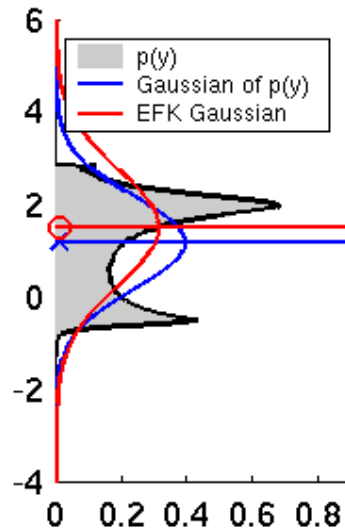
Linearity Assumption Revisited



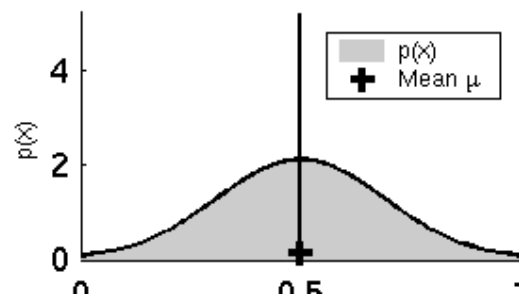
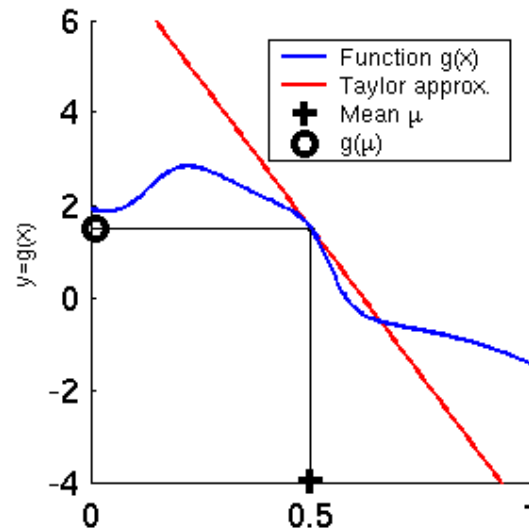
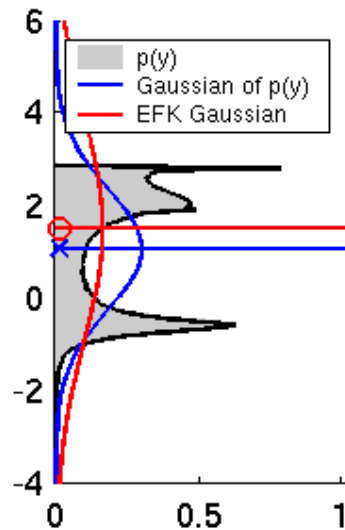
Non-linear Function



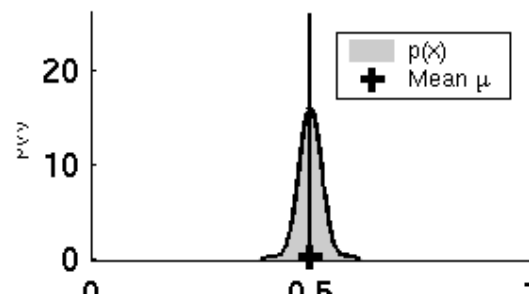
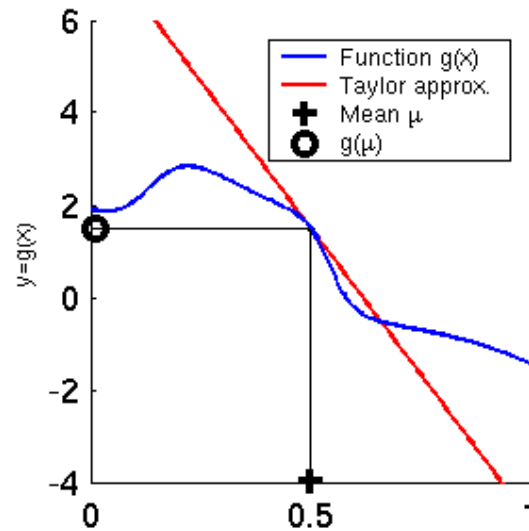
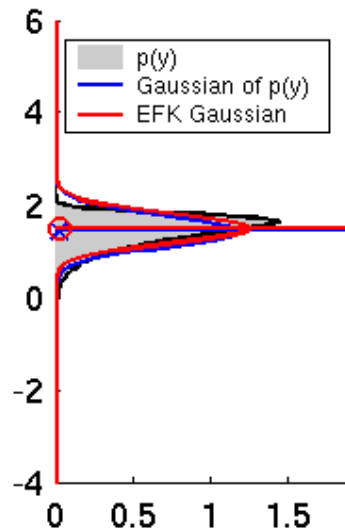
EKF Linearization (1)



EKF Linearization (2)



EKF Linearization (3)



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. $\bar{\mu}_t = g(u_t, \mu_{t-1})$ \longleftarrow $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$ \longleftarrow $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. Correction:

6. $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$ \longleftarrow $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$ \longleftarrow $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
8. $\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$ \longleftarrow $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

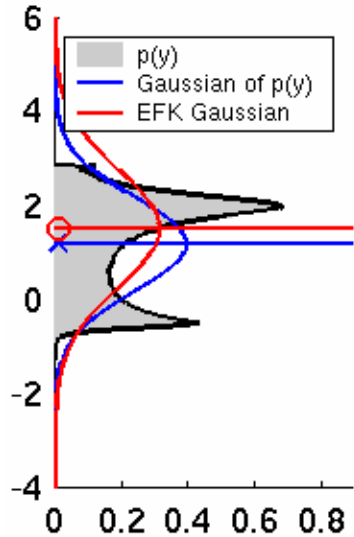
9. **Return** μ_t, Σ_t

$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \quad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$$

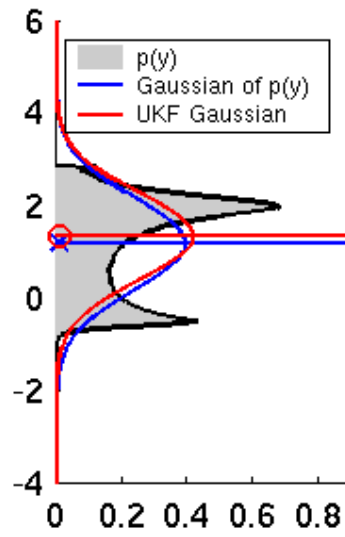
EKF Summary

- **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!

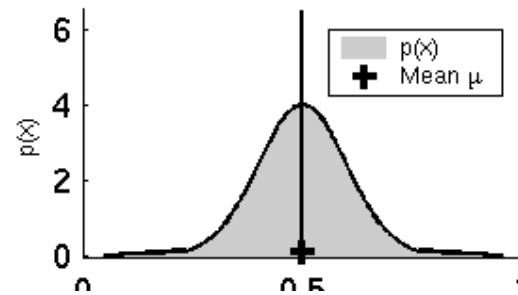
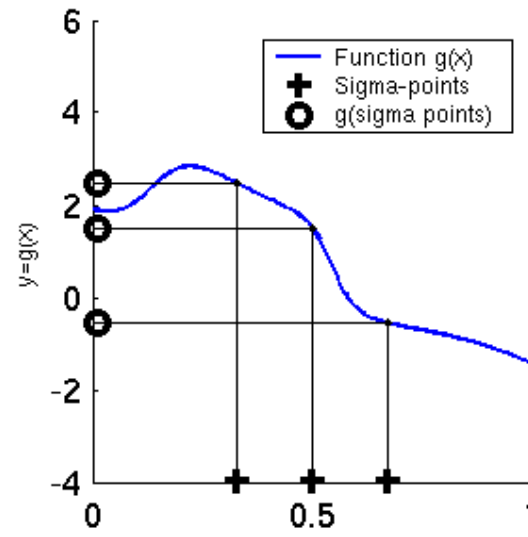
Linearization via Unscented Transform



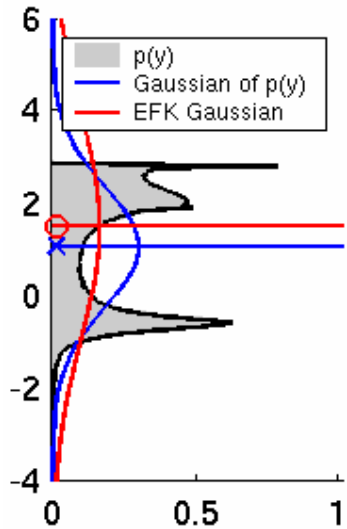
EKF



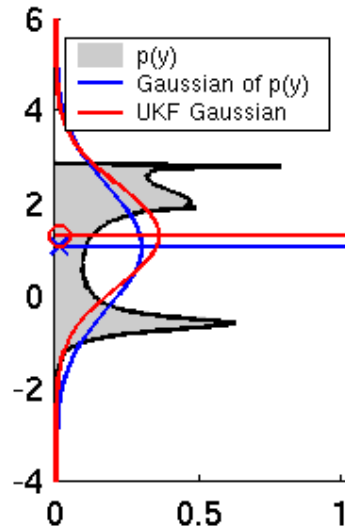
UKF



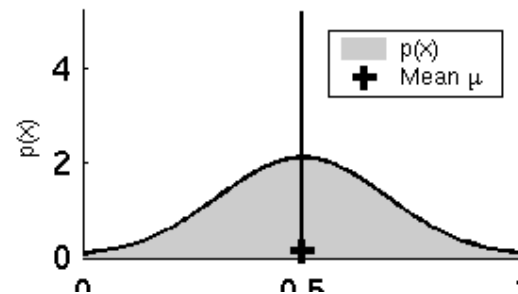
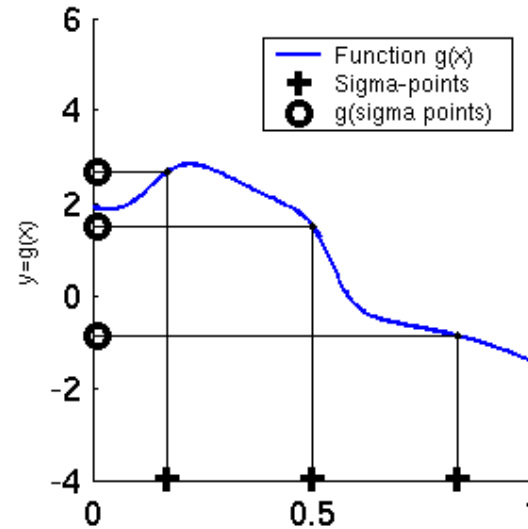
UKF Sigma-Point Estimate (2)



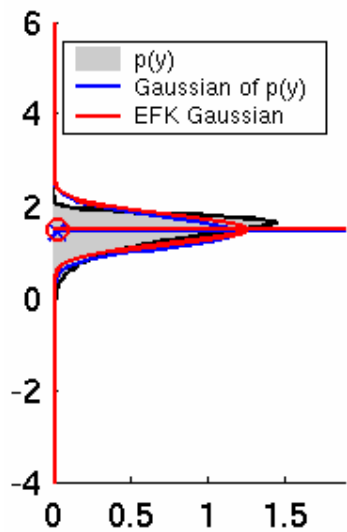
EKF



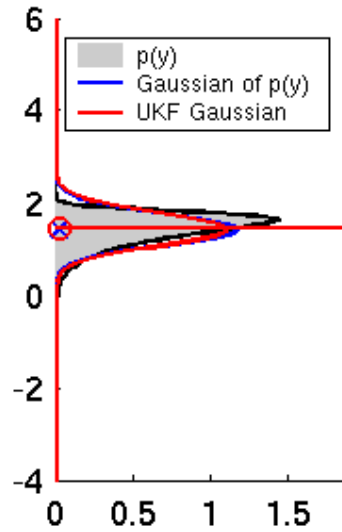
UKF



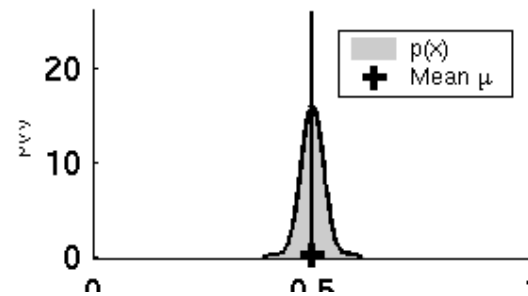
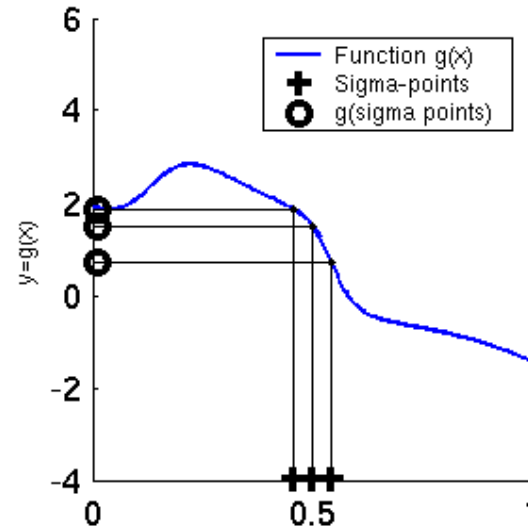
UKF Sigma-Point Estimate (3)



EKF



UKF



Unscented Transform

Sigma points

$$\chi^0 = \mu$$

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

Weights

$$w_m^0 = \frac{\lambda}{n + \lambda} \quad w_c^0 = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta)$$

$$w_m^i = w_c^i = \frac{1}{2(n + \lambda)} \quad \text{for } i = 1, \dots, 2n$$

Pass sigma points through nonlinear function

$$\psi^i = g(\chi^i)$$

Recover mean and covariance

$$\mu' = \sum_{i=0}^{2n} w_m^i \psi^i$$

$$\Sigma' = \sum_{i=0}^{2n} w_c^i (\psi^i - \mu)(\psi^i - \mu)^T$$

UKF_localization ($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m$):

Prediction:

$$M_t = \begin{pmatrix} (\alpha_1 |v_t| + \alpha_2 |\omega_t|)^2 & 0 \\ 0 & (\alpha_3 |v_t| + \alpha_4 |\omega_t|)^2 \end{pmatrix} \quad \text{Motion noise}$$

$$Q_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{pmatrix} \quad \text{Measurement noise}$$

$$\mu_{t-1}^a = \begin{pmatrix} \mu_{t-1}^T & (0\ 0)^T & (0\ 0)^T \end{pmatrix} \quad \text{Augmented state mean}$$

$$\Sigma_{t-1}^a = \begin{pmatrix} \Sigma_{t-1} & 0 & 0 \\ 0 & M_t & 0 \\ 0 & 0 & Q_t \end{pmatrix} \quad \text{Augmented covariance}$$

$$\chi_{t-1}^a = \left(\mu_{t-1}^a \quad \mu_{t-1}^a + \gamma \sqrt{\Sigma_{t-1}^a} \quad \mu_{t-1}^a - \gamma \sqrt{\Sigma_{t-1}^a} \right) \quad \text{Sigma points}$$

$$\bar{\chi}_t^x = g(u_t + \chi_t^u, \chi_{t-1}^x) \quad \text{Prediction of sigma points}$$

$$\bar{\mu}_t = \sum_{i=0}^{2L} w_m^i \chi_{i,t}^x \quad \text{Predicted mean}$$

$$\bar{\Sigma}_t = \sum_{i=0}^{2L} w_c^i (\chi_{i,t}^x - \bar{\mu}_t)(\chi_{i,t}^x - \bar{\mu}_t)^T \quad \text{Predicted covariance}$$

UKF_localization ($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m$):

Correction:

$$\bar{Z}_t = h(\chi_t^x) + \chi_t^z$$

Measurement sigma points

$$\hat{z}_t = \sum_{i=0}^{2L} w_m^i \bar{Z}_{i,t}$$

Predicted measurement mean

$$S_t = \sum_{i=0}^{2L} w_c^i (\bar{Z}_{i,t} - \hat{z}_t)(\bar{Z}_{i,t} - \hat{z}_t)^T$$

Pred. measurement covariance

$$\Sigma_t^{x,z} = \sum_{i=0}^{2L} w_c^i (\bar{\chi}_{i,t}^x - \bar{\mu}_t)(\bar{Z}_{i,t} - \hat{z}_t)^T$$

Cross-covariance

$$K_t = \Sigma_t^{x,z} S_t^{-1}$$

Kalman gain

$$\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$$

Updated mean

$$\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$$

Updated covariance

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

Correction:

2. $\hat{z}_t = \begin{pmatrix} \sqrt{(m_x - \bar{\mu}_{t,x})^2 + (m_y - \bar{\mu}_{t,y})^2} \\ \text{atan2}(m_y - \bar{\mu}_{t,y}, m_x - \bar{\mu}_{t,x}) - \bar{\mu}_{t,\theta} \end{pmatrix}$ Predicted measurement mean

3. $H_t = \frac{\partial h(\bar{\mu}_t, m)}{\partial x_t} = \begin{pmatrix} \frac{\partial r_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,y}} & \frac{\partial r_t}{\partial \bar{\mu}_{t,\theta}} \\ \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,x}} & \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,y}} & \frac{\partial \varphi_t}{\partial \bar{\mu}_{t,\theta}} \end{pmatrix}$ Jacobian of h w.r.t location

4. $Q_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{pmatrix}$

5. $S_t = H_t \bar{\Sigma}_t H_t^T + Q_t$ Pred. measurement covariance

6. $K_t = \bar{\Sigma}_t H_t^T S_t^{-1}$ 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

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!**

Information Filter

- **Gaussian Filter**: Same assumptions as KF and its derivatives EKF and UKF
- **Dual representation**: the inverse of the covariance matrix Σ is maintained; the *information* or *precision matrix* Ω
- **Dual complexity**: easier correction step, more complex prediction step
- **Perfect for uncertain worlds**: $\Omega = 0$

Information Filter

- Gaussian Distribution:

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right\}$$

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}x^T \Sigma^{-1}x + \frac{2}{2}x^T \Sigma^{-1}\mu - \frac{1}{2}\mu^T \Sigma^{-1}\mu\right\}$$

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{\frac{1}{2}\mu^T \Sigma^{-1}\mu\right\} \cdot \exp\left\{-\frac{1}{2}x^T \Sigma^{-1}x + \frac{2}{2}x^T \Sigma^{-1}\mu\right\}$$

$$p(x) = \eta \cdot \exp\left\{-\frac{1}{2}x^T \Sigma^{-1}x + x^T \Sigma^{-1}\mu\right\}$$

$$p(x) = \eta \cdot \exp\left\{-\frac{1}{2}x^T \Omega x + x^T \xi\right\}$$

with *information matrix* $\Omega = \Sigma^{-1}$
and *information vector* $\xi = \Sigma^{-1}\mu$

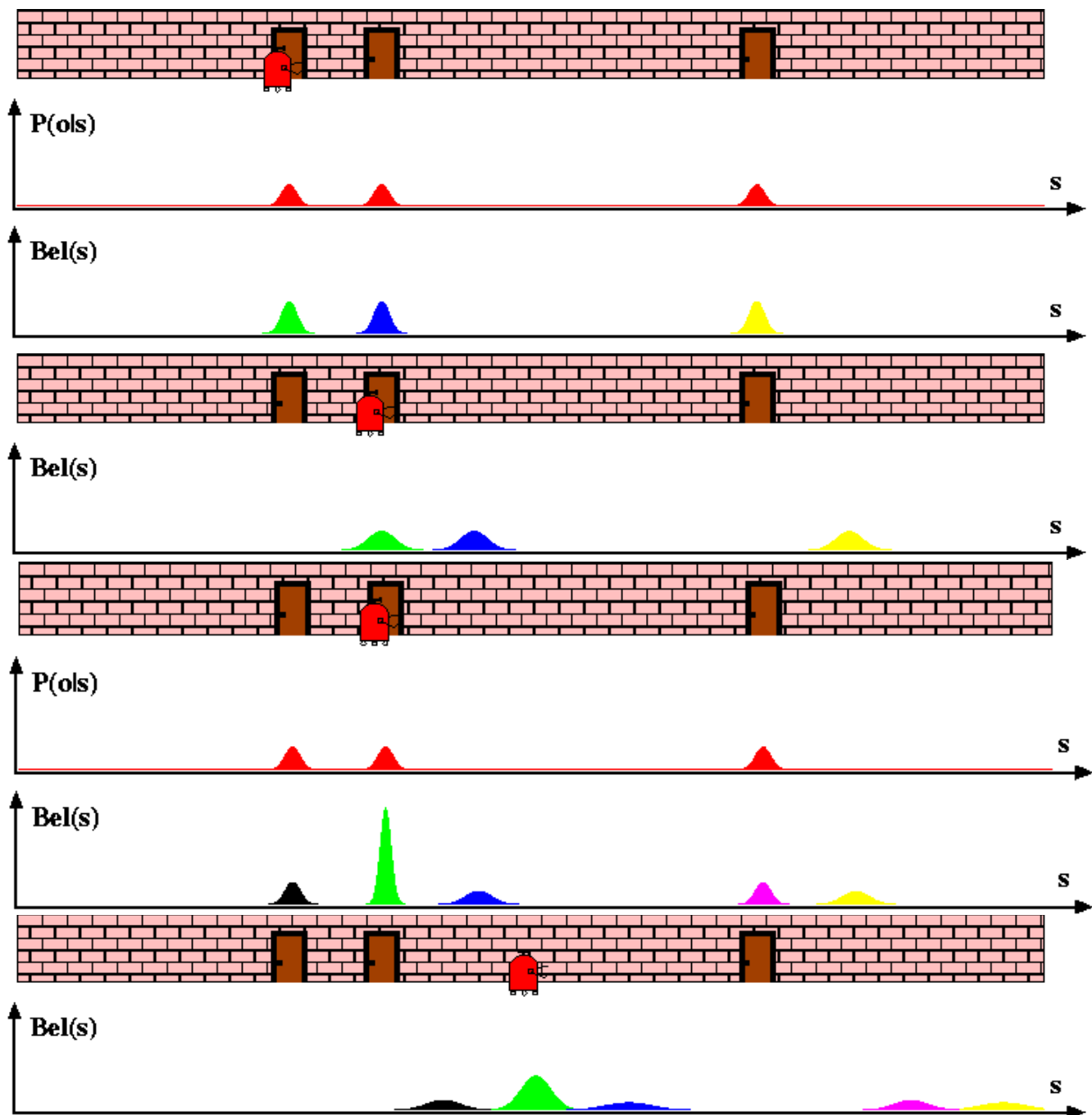
Information Filter Algorithm

1. Algorithm **Information_filter**($\xi_{t-1}, \Omega_{t-1}, u_t, z_t$):
2. Prediction:
3. $\bar{\Omega}_t = (A_t \Omega_{t-1} A_t^T + R_t)^{-1}$
4. $\bar{\xi}_t = \bar{\Omega}_t (A_t \Omega_{t-1}^{-1} \xi_{t-1} + B_t u_t)$
5. Correction:
- 6.
7. $\Omega_t = C_t^T Q_t^{-1} C_t + \bar{\Omega}_t$
8. $\xi_t = C_t^T Q_t^{-1} z_t - \bar{\mu}_t + \bar{\xi}_t$
9. Return ξ_t, Ω_t

IF Summary

- **Highly efficient:** Same complexity as KF, only shifted from correction to prediction step
- **Can be linearization equivalent to KF:** for instance the extended information filter EIF
- **Easier fusion of measurements:** simply adding matrices
- **Locality corresponds with sparse Ω !**

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.

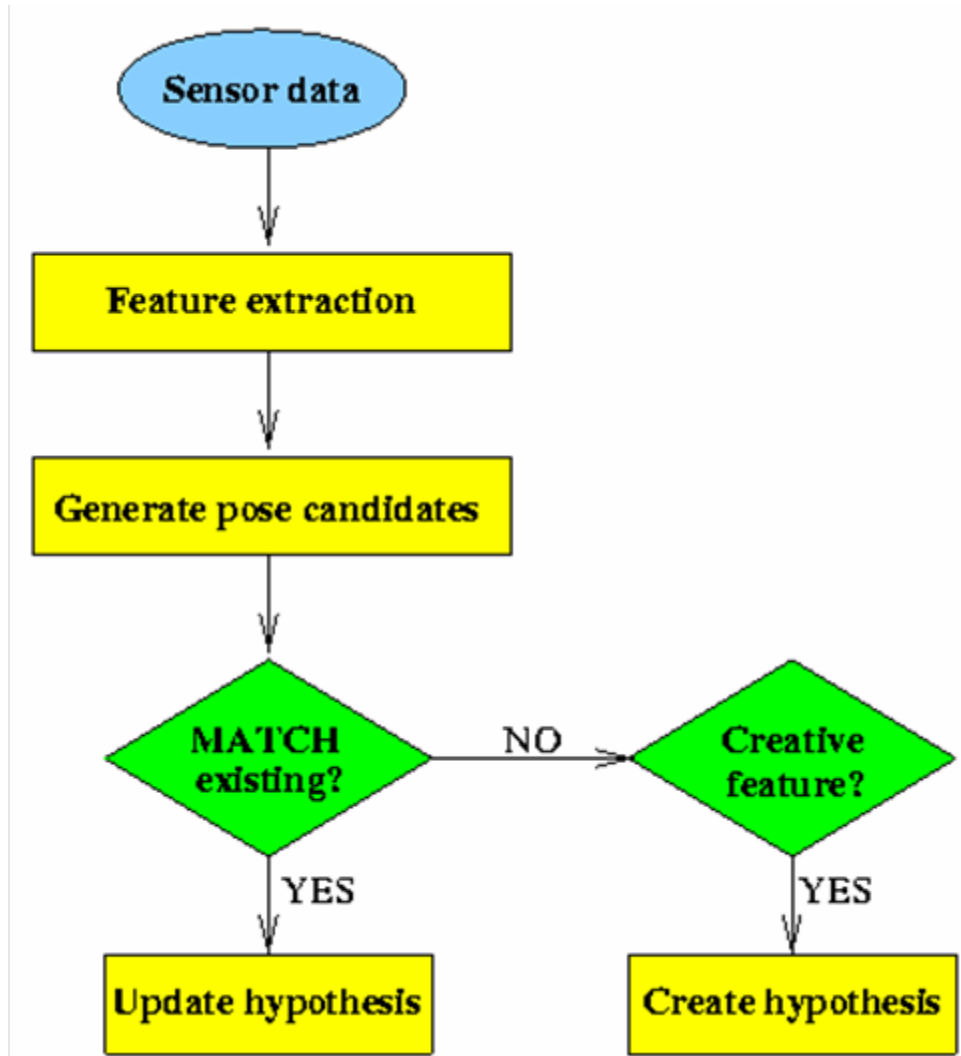
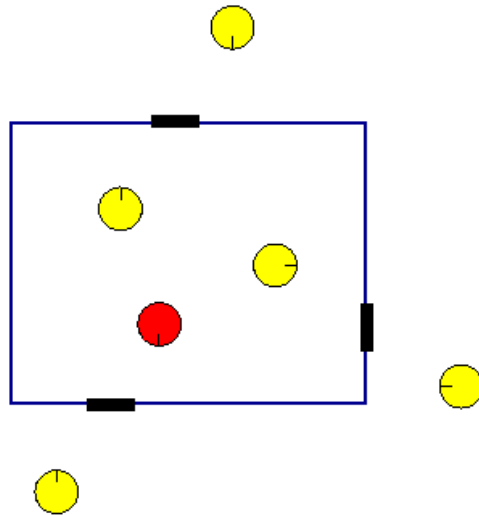
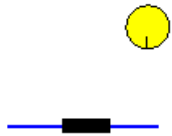
$$C_j = \{z_j, R_j\}$$

[Jensfelt et al. '00]

MHT: Implemented System (2)

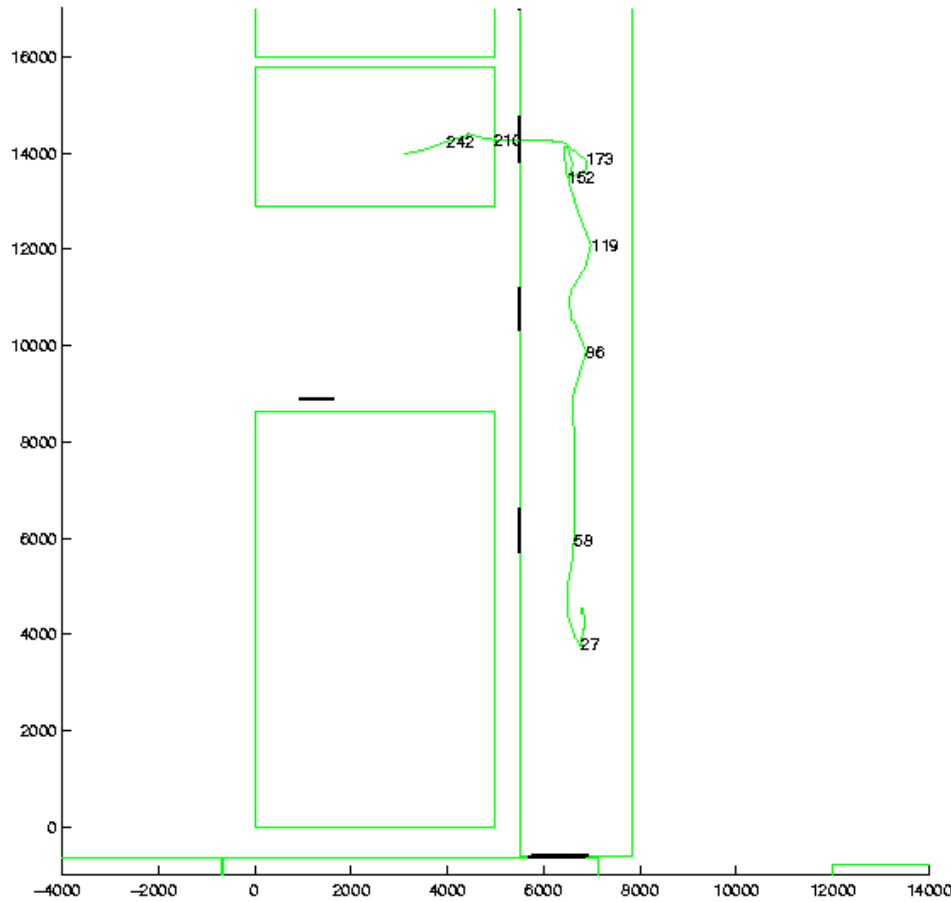
Robot view

Pose candidates

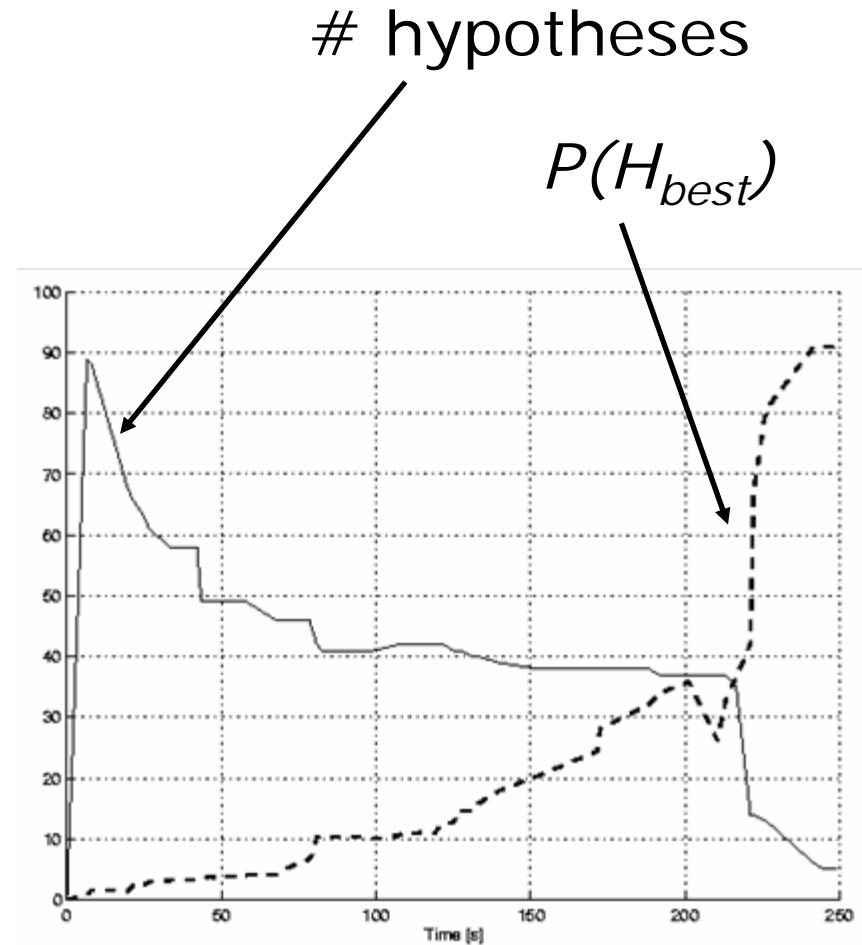


MHT: Implemented System (3)

Example run



Map and trajectory



#hypotheses vs. time

Summary: Kalman Filter

- Gaussian Posterior, Gaussian Noise, efficient when applicable
- KF: Motion, Sensing = linear
- EKF: nonlinear, uses Taylor expansion
- UKF: nonlinear, uses sampling
- IF & EIF: dual of KF & EKF
- MHKF: Combines best of Kalman Filters and particle filters