Probabilistic Robotics

Bayes Filter Implementations

Gaussian filters

1

Kalman Filter Algorithm

1. Algorithm **Kalman_filter**(μ_{t-1}, Σ_{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. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

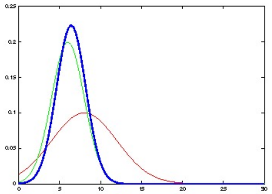
9. Return μ_t, Σ_t

3

3

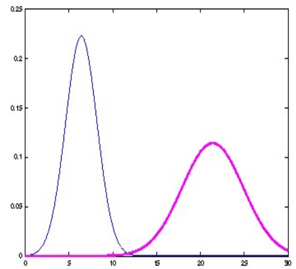
The Prediction-Correction-Cycle

Prediction



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

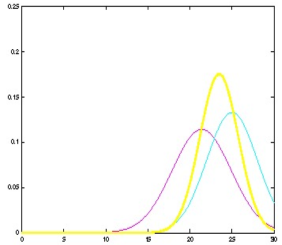
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$



4

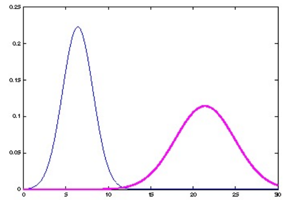
4

The Prediction-Correction-Cycle



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

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

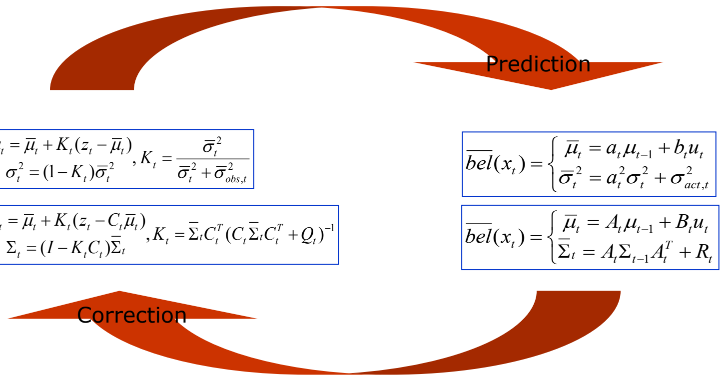


Correction

5

5

The Prediction-Correction-Cycle



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

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

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

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

6

6

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!

7

7

Nonlinear Dynamic Systems

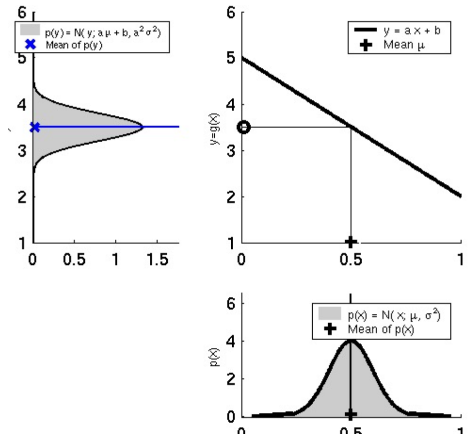
- Most realistic robotic problems involve nonlinear functions

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

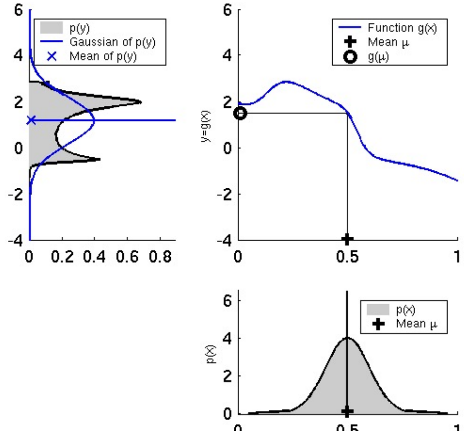
$$z_t = h(x_t)$$

- To be continued

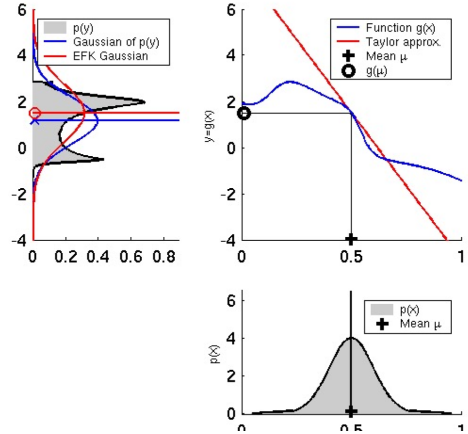
Linearity Assumption Revisited



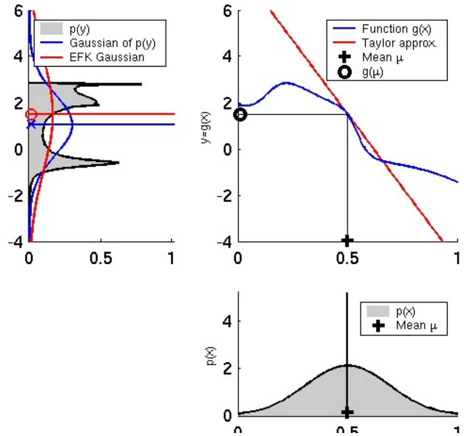
Non-linear Function



EKF Linearization (1)



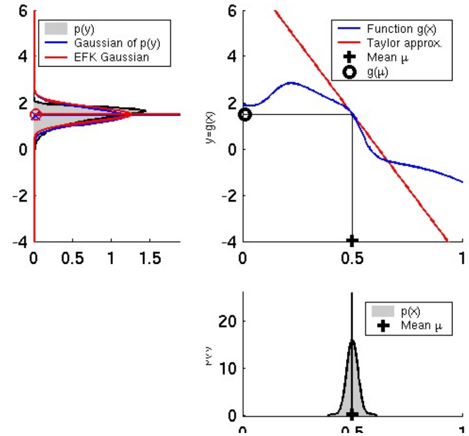
EKF Linearization (2)



12

12

EKF Linearization (3)



Depends on uncertainty

13

13

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:

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

$$h(x_t) \approx h(\bar{\mu}_t) + H_t\,(x_t - \bar{\mu}_t)$$

14

14

EKF Algorithm

1. **Extended_Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):
2. Prediction:
3. $\bar{\mu}_t = g(u_t, \mu_{t-1})$ ⟵ $\bar{\mu}_t = A_t\mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = G_t\Sigma_{t-1}G_t^T + R_t$ ⟵ $\bar{\Sigma}_t = A_t\Sigma_{t-1}A_t^T + R_t$
5. Correction:
6. $K_t = \bar{\Sigma}_t H_t^T(H_t\bar{\Sigma}_t H_t^T + Q_t)^{-1}$ ⟵ $K_t = \bar{\Sigma}_t C_t^T(C_t\bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$ ⟵ $\mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t)$
8. $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$ ⟵ $\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$
9. Return μ_t, Σ_t

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

15

15

Robot Position Prediction

- In a first step, the robots position at time step k+1 is predicted based on its old location (time step k) and its movement due to the control input $u(k)$:

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

f: Odometry function

$$\Sigma_p(k+1|k) = \nabla_p f \cdot \Sigma_p(k|k) \cdot \nabla_p f^T + \nabla_u f \cdot \Sigma_u(k) \cdot \nabla_u f^T$$

16

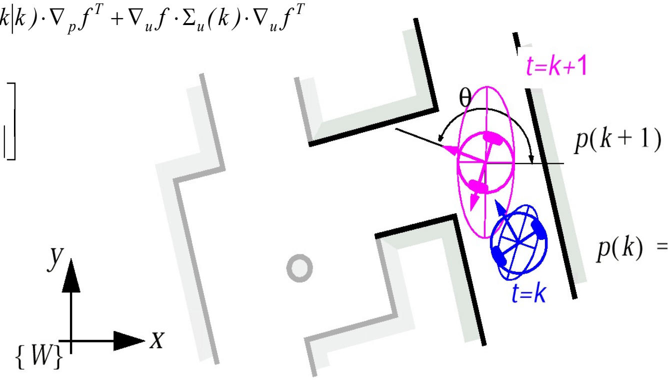
Robot Position Prediction: *Example*

$$\hat{p}(k+1|k) = \hat{p}(k|k) + u(k) = \begin{bmatrix} \hat{x}(k) \\ \hat{y}(k) \\ \hat{\theta}(k) \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$

Odometry

$$\Sigma_p(k+1|k) = \nabla_p f \cdot \Sigma_p(k|k) \cdot \nabla_p f^T + \nabla_u f \cdot \Sigma_u(k) \cdot \nabla_u f^T$$

$$\Sigma_u(k) = \begin{bmatrix} k_r|\Delta s_r| & 0 \\ 0 & k_l|\Delta s_l| \end{bmatrix}$$

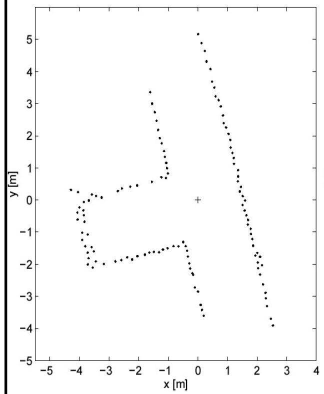


17

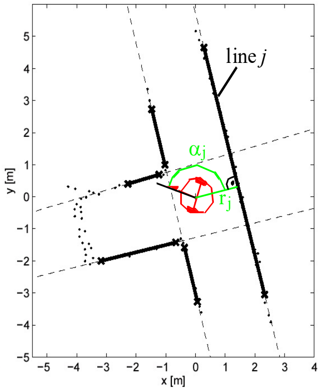
Kalman Filter for Mobile Robot Localization

Observation: *Example*

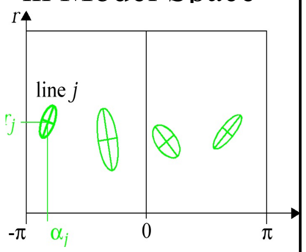
Raw Date of Laser Scanner



Extracted Lines



Extracted Lines in Model Space



$$z_j(k+1) = {}^{R}\begin{bmatrix} \alpha_j \\ r_j \end{bmatrix}$$

Sensor (robot) frame

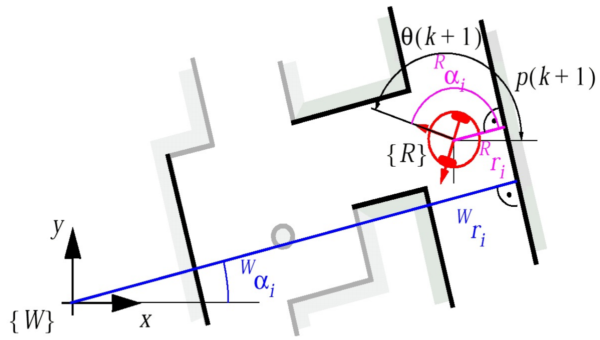
$$\Sigma_{R,j}(k+1) = \begin{bmatrix} \sigma_{\alpha\alpha} & \sigma_{\alpha r} \\ \sigma_{r\alpha} & \sigma_{rr} \end{bmatrix}_j$$

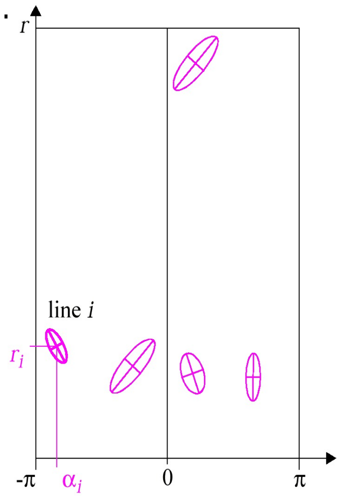
Set of discrete line features

Kalman Filter for Mobile Robot Localization

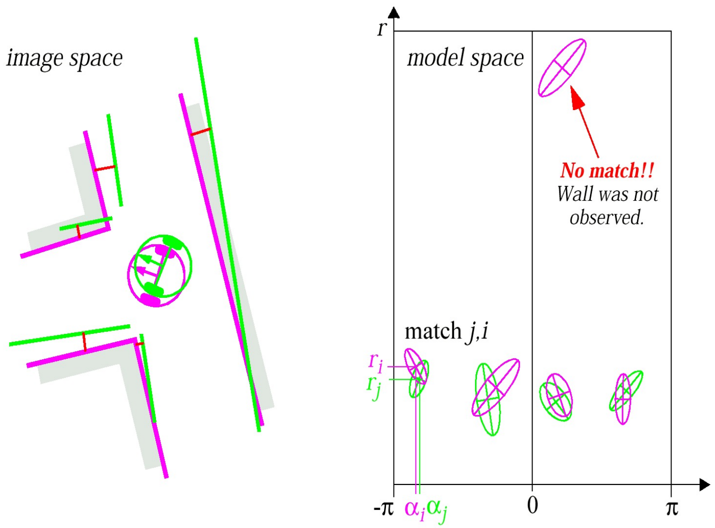
Measurement Prediction: *Example*

- For prediction, only the walls that are in the field of view of the robot are selected.
- This is done by linking the individual lines to the nodes of the path





Matching: *Example*

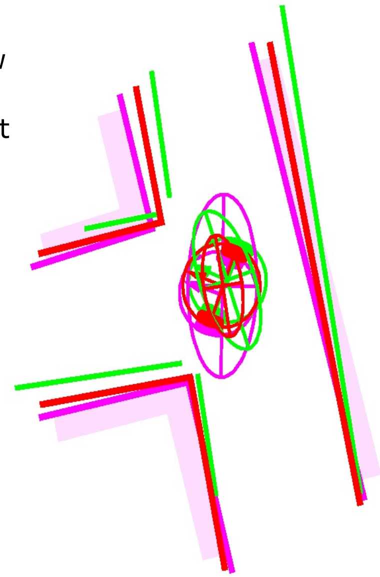


24

Kalman Filter for Mobile Robot Localization

Estimation: *Example*

- Kalman filter estimation of the new robot position $\hat{p}(k|k)$:
  - By fusing the prediction of robot position (magenta) with the innovation gained by the measurements (green) we get the updated estimate of the robot position (red)



28

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]

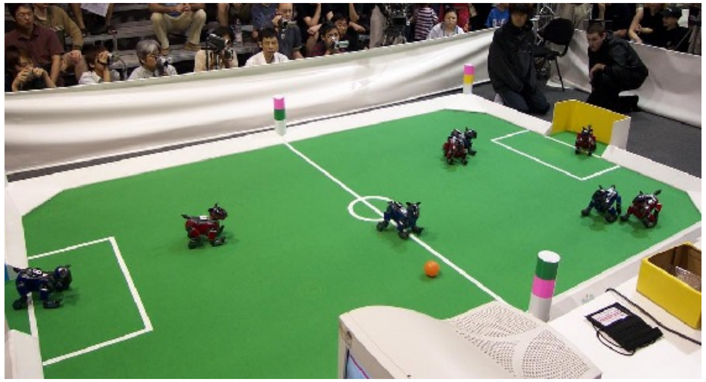
- **Given**
  - Map of the environment.
  - Sequence of sensor measurements.
- **Wanted**
  - Estimate of the robot's position.
- **Problem classes**
  - Position tracking
  - Global localization
  - Kidnapped robot problem (recovery)

30

30

Landmark-based Localization

Chapter 7.4 Probabilistic robotics



31

31

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

Prediction:

3. $G_t = \dfrac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{pmatrix} \frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial \theta'}{\partial \mu_{t-1,x}} & \frac{\partial \theta'}{\partial \mu_{t-1,y}} & \frac{\partial \theta'}{\partial \mu_{t-1,\theta}} \end{pmatrix}$ Jacobian of *g* w.r.t location

5. $V_t = \dfrac{\partial g(u_t, \mu_{t-1})}{\partial u_t} = \begin{pmatrix} \frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial \omega_t} \\ \frac{\partial y'}{\partial v_t} & \frac{\partial y'}{\partial \omega_t} \\ \frac{\partial \theta'}{\partial v_t} & \frac{\partial \theta'}{\partial \omega_t} \end{pmatrix}$ Jacobian of *g* w.r.t control

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

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

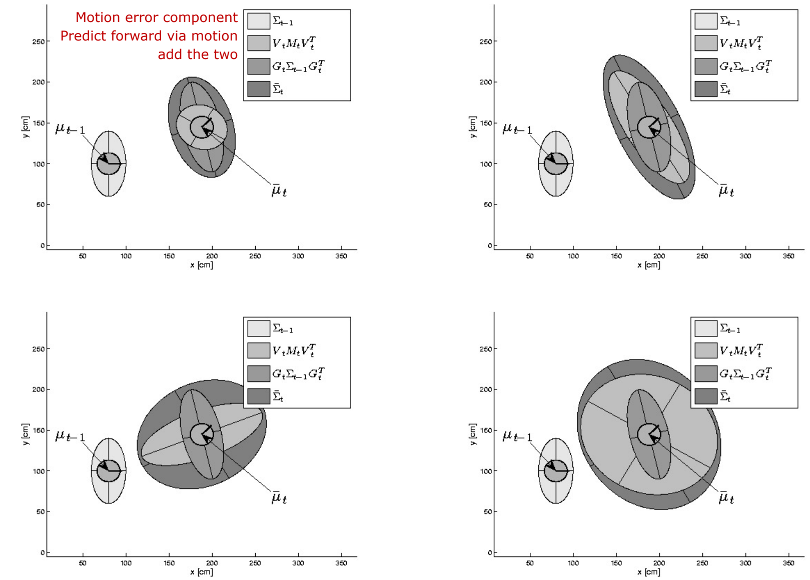
8. $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T$ Predicted covariance

32

32

EKF Prediction Step

Prediction step for different motion noise parameters (10%,10%), (30%,10%),(10%,30%) (30%,30%) – for motion 10cm/sec, 5 degrees/sec for 9 seconds), previous mean (80,100,0)



33

33

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

Correction:

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

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

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

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

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

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

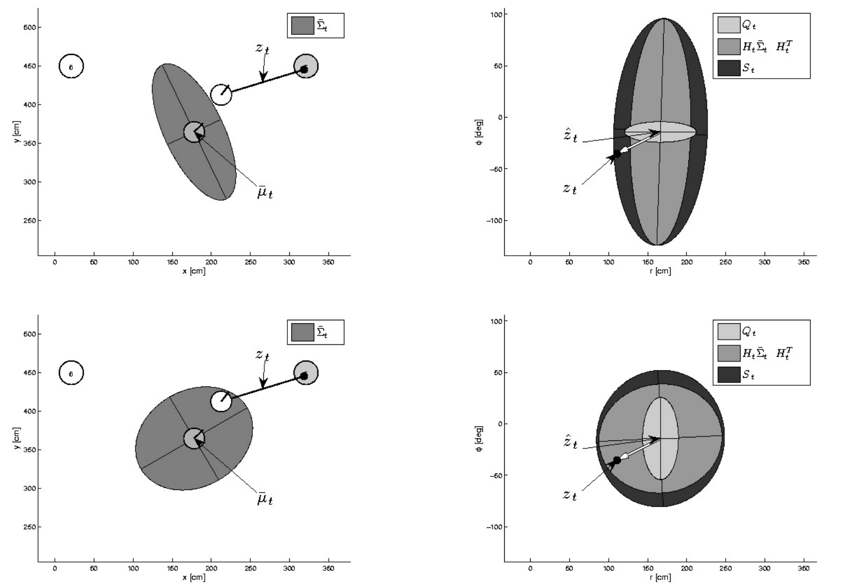
10. $\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$ Updated covariance

34

34

EKF Observation Prediction Step

S_t predicted
and actual measurements
measurement covariance Q_t

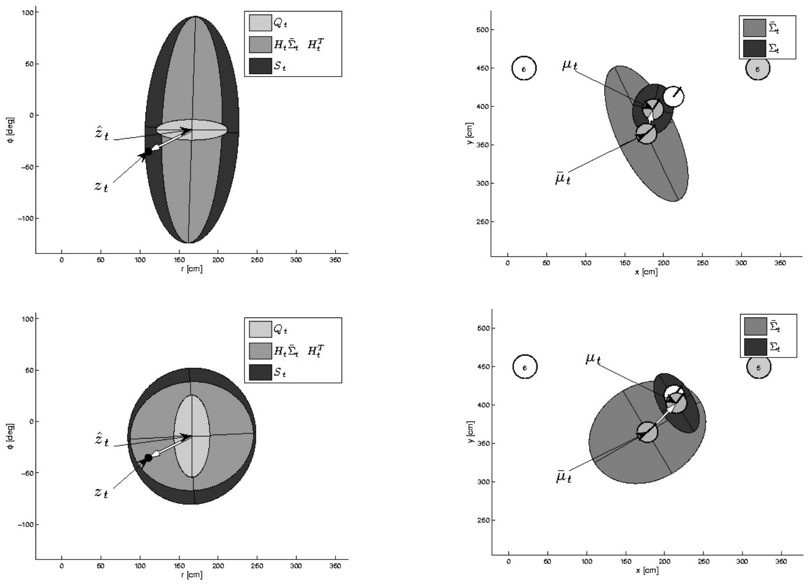


35

35

EKF Correction Step

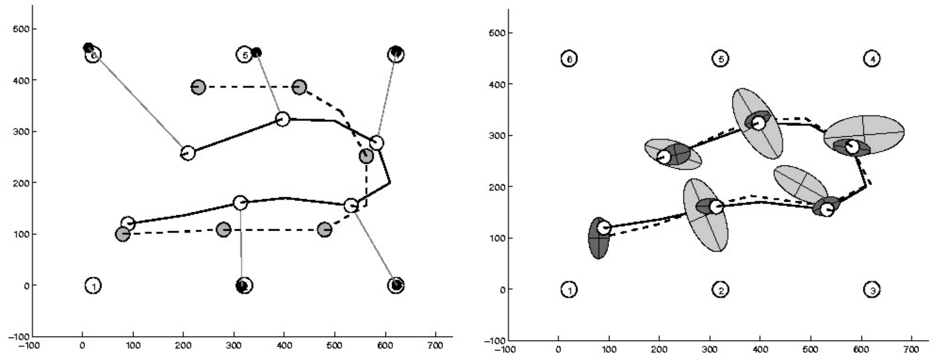
Reduction in uncertainty
and different mean location



36

36

Estimation Sequence (1)

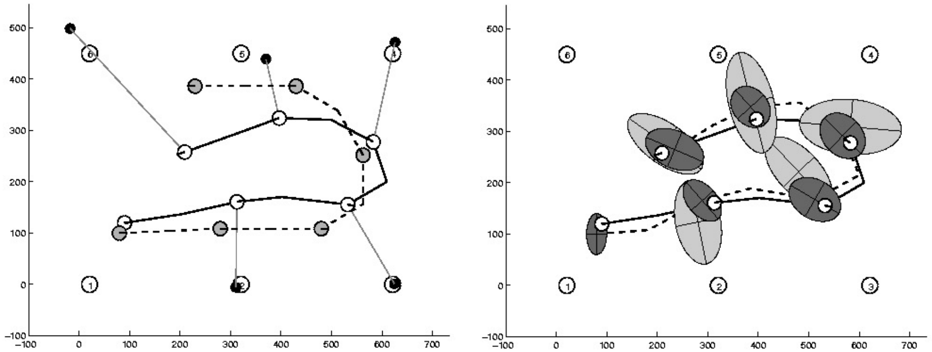


Errors with motion models, with landmark based localization

37

37

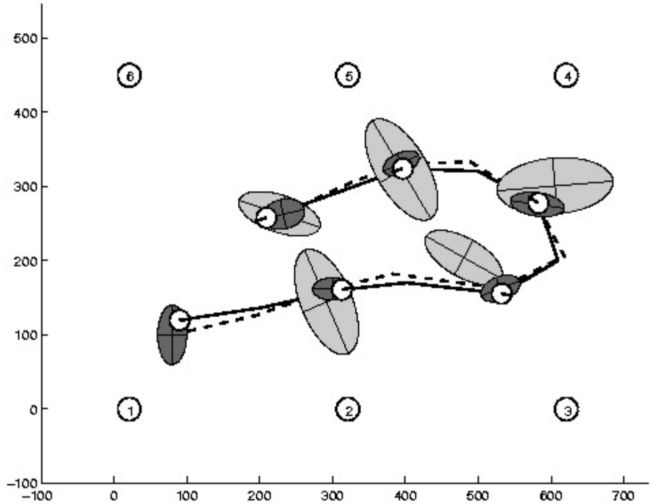
Estimation Sequence (2)



Errors with motion models, with landmark based localization with less accurate sensor

38

Comparison to GroundTruth



39

EKF Summary

- Highly efficient: Polynomial in measurement dimensionality k and state dimensionality n:

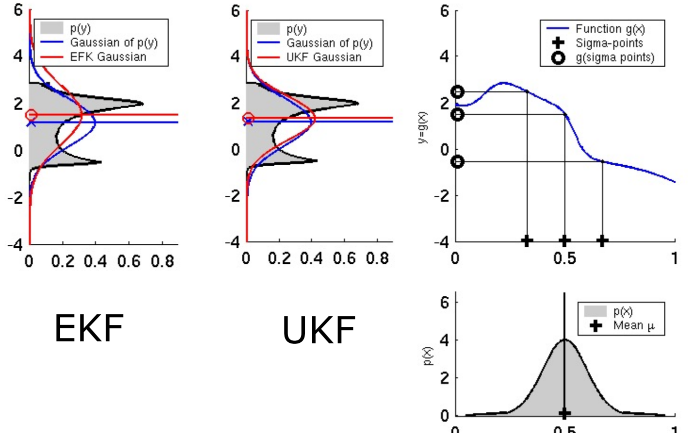
$$O(k^{2.376} + n^2)$$

- Not optimal!
- Can diverge if nonlinearities are large!
- Works surprisingly well even when all assumptions are violated!

- Unscented EKF

Linearization via Unscented Transform

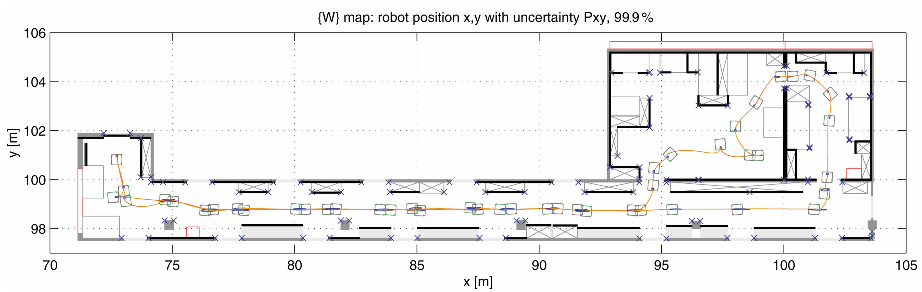
Chapter 3.4 Probabilistic robotics



Idea – better approximation of the mean and covariance Transform (sigma points of the original gaussian, to better Approximate the final distribution

Kalman Filter-based System

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

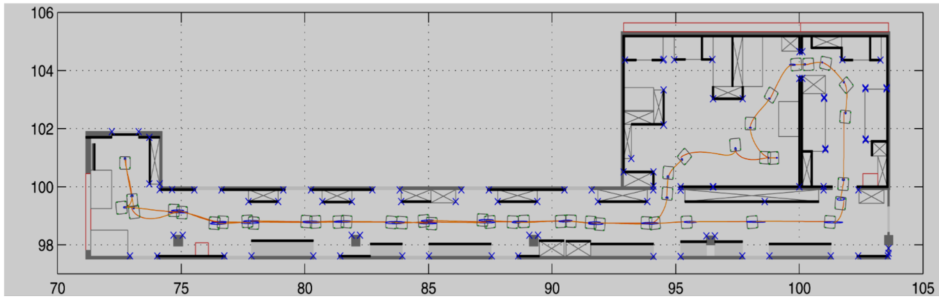


[Courtesy of Kai Arras]

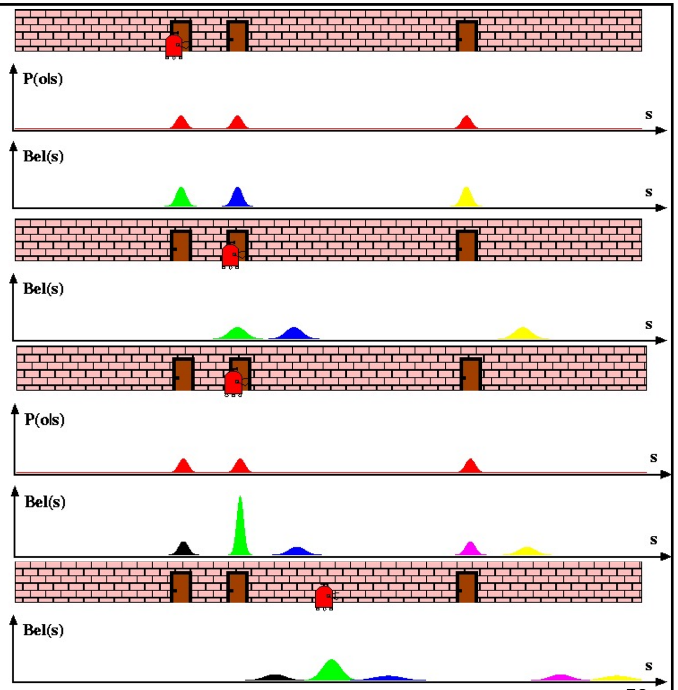
56

Autonomous Indoor Navigation (Pyg

- very robust on-the-fly localization
- one of the first systems with probabilistic s
- 47 steps,78 meter length, realistic office er
- conducted 16 times > 1km overall distance
- partially difficult surfaces (laser), partially f... edges (vision)



Multi-hypothesis Tracking



58

58

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.

59

59

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.

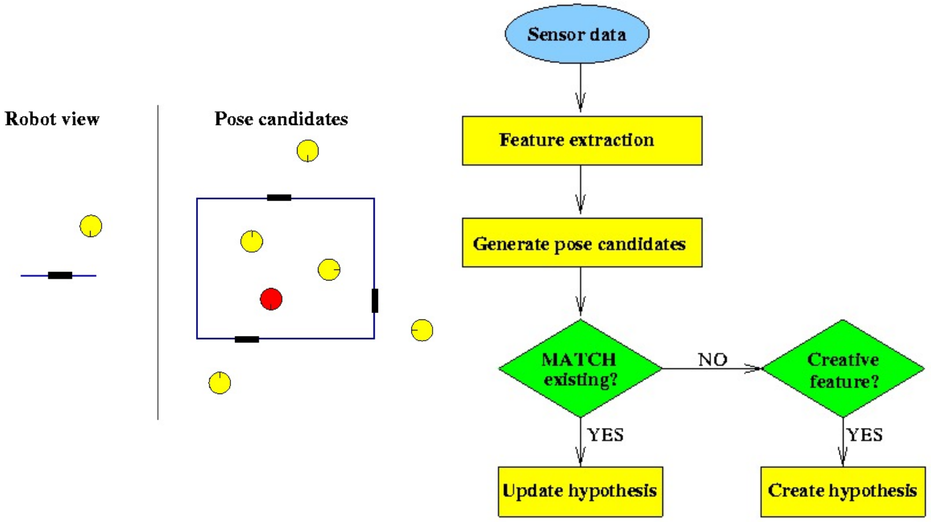
$$C_j = \{z_j, R_j\}$$

[Jensfelt et al. '00]

60

60

MHT: Implemented System (2)

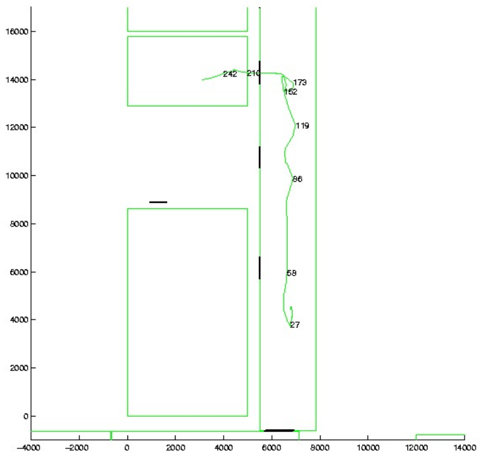


Courtesy of P. Jensfelt and S. Kristensen

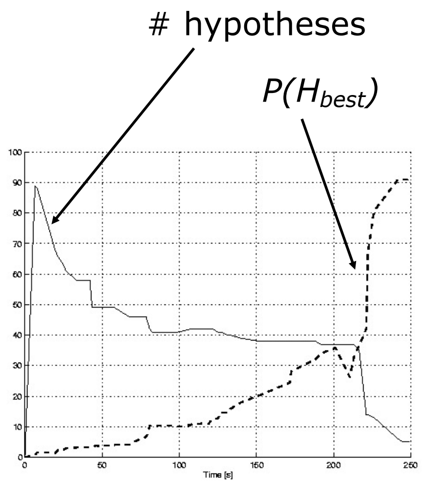
61

61

MHT: Implemented System (3)
Example run



Map and trajectory



#hypotheses vs. time

Courtesy of P. Jensfelt and S. Kristensen