# CSE-571
# Robotics

**Kalman Filters**

Dieter Fox

1

## Bayes Filter Reminder

- Prediction

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

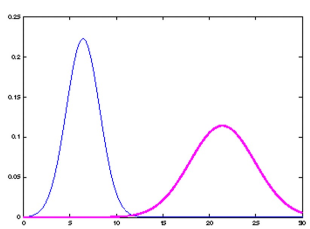
- Correction

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

3

## Properties of Gaussians

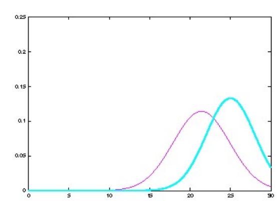
$$\left.\begin{array}{l} X \sim N(\mu, \sigma^2) \\ Y = aX + b \end{array}\right\} \Rightarrow \quad Y \sim N(a\mu + b, a^2\sigma^2)$$

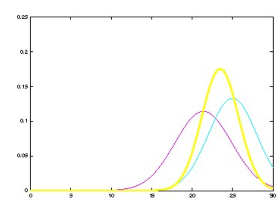


5

## Properties of Gaussians

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

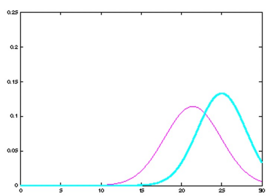


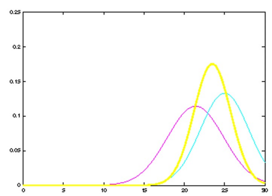


6

## Properties of Gaussians

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





7

## Multivariate Gaussians

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

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

- 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.

9

9

## 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

$$z_t = C_t x_t + \delta_t$$

10

11

## 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 $t-1$.

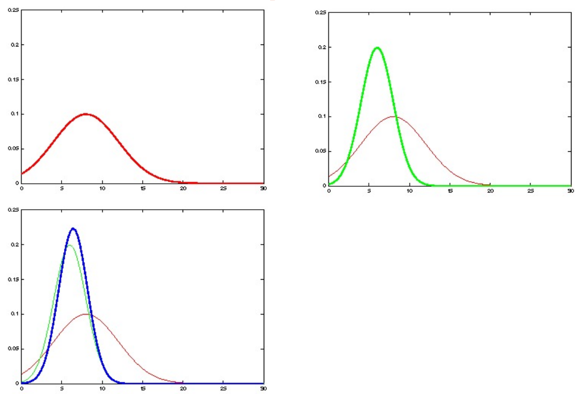
$C_t$ Matrix (kxn) that describes how to map the state $x_t$ to an observation $z_t$.

$\varepsilon_t$ $\delta_t$ 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.

12

12

## Kalman Filter Updates in 1D



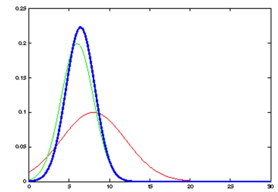
13

13

## Kalman Filter Updates in 1D

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

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



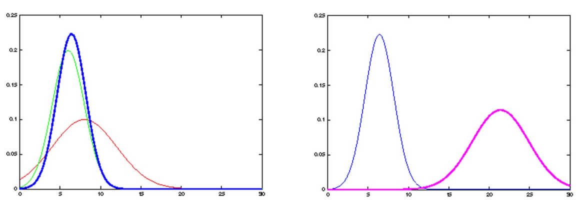
14

15

## Kalman Filter Updates in 1D

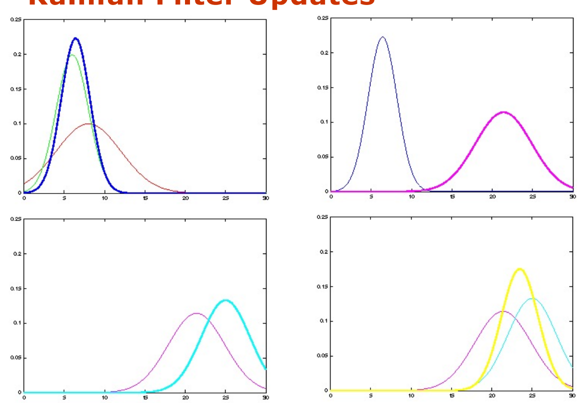
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

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



17

## Kalman Filter Updates



18

## Linear Gaussian Systems: Initialization

- Initial belief is normally distributed:

$$bel(x_0) = N(x_0; \mu_0, \Sigma_0)$$

20

## Linear Gaussian Systems: Dynamics

- Dynamics are linear function of state and control plus additive noise:

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

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

$$\begin{array}{ll} \overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) & bel(x_{t-1})\, dx_{t-1} \\ \Downarrow & \Downarrow \\ \sim N(x_t; A_t x_{t-1} + B_t u_t, R_t) & \sim N(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}) \end{array}$$

22

## Linear Gaussian Systems: Dynamics

$$\begin{array}{ll} \overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) & bel(x_{t-1})\, dx_{t-1} \\ \Downarrow & \Downarrow \\ \sim N(x_t; A_t x_{t-1} + B_t u_t, R_t) & \sim N(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}) \\ \Downarrow & \end{array}$$

$$\overline{bel}(x_t) = \eta \int \exp\left\{-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)\right\}$$

$$\exp\left\{-\frac{1}{2}(x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - \mu_{t-1})\right\} dx_{t-1}$$

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

24

## Linear Gaussian Systems: Observations

- Observations are linear function of state plus additive noise:

$$z_t = C_t x_t + \delta_t$$

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

$$\begin{array}{lll} bel(x_t) = \eta & p(z_t \mid x_t) & \overline{bel}(x_t) \\ & \Downarrow & \Downarrow \\ & \sim N(z_t; C_t x_t, Q_t) & \sim N(x_t; \bar{\mu}_t, \bar{\Sigma}_t) \end{array}$$

26

## Linear Gaussian Systems: Observations

$$bel(x_t) = \eta \quad p(z_t \mid x_t) \qquad \overline{bel}(x_t)$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$\sim N(z_t; C_t x_t, Q_t) \qquad \sim N(x_t; \bar{\mu}_t, \bar{\Sigma}_t)$$

$$\Downarrow$$

$$bel(x_t) = \eta \exp\left\{-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t)\right\} \exp\left\{-\frac{1}{2}(x_t - \bar{\mu}_t)^T \bar{\Sigma}_t^{-1}(x_t - \bar{\mu}_t)\right\}$$

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

28

## 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 $\mu_t, \Sigma_t$

30

## 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!

31

Going non-linear

## EXTENDED KALMAN FILTER

32

32

## Nonlinear Dynamic Systems

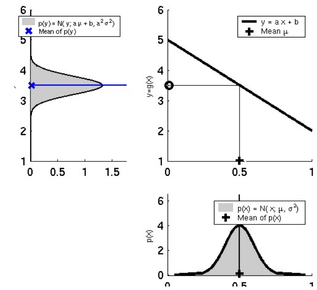
- Most realistic robotic problems involve nonlinear functions

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

$$\boxed{z_t = h(x_t)}$$

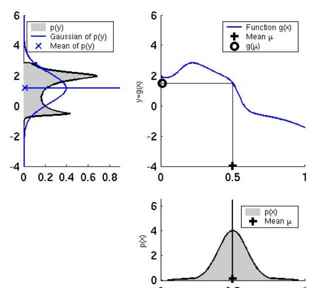
33

## Linearity Assumption Revisited



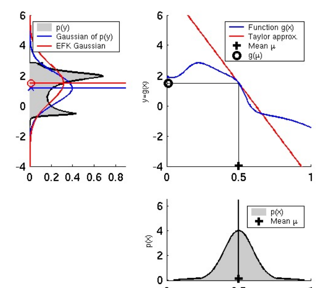

34

## Non-linear Function



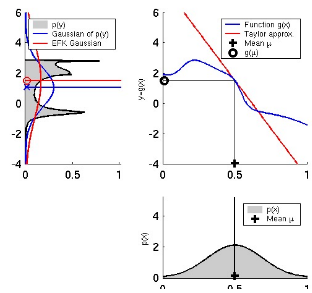

35

## EKF Linearization (1)



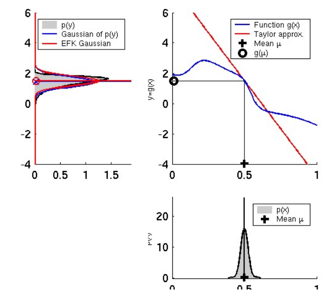

36

## EKF Linearization (2)




37

## EKF Linearization (3)




38

## 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:

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

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

40

## 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 $\mu_t, \Sigma_t$

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

42

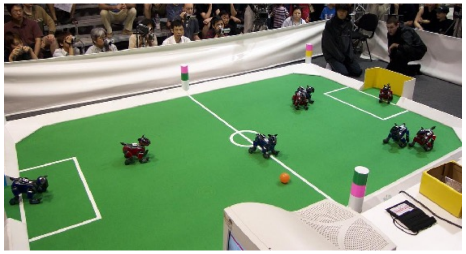
## Localization

> "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)

43

## Landmark-based Localization



44

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

**Prediction:**

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

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

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

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

45

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

**Prediction:**

$$\theta = \mu_{t-1,\theta}$$

$$G_t = \begin{pmatrix} 1 & 0 & -\frac{v_t}{\omega_t}\cos\theta + \frac{v_t}{\omega_t}\cos(\theta + \omega_t \Delta t) \\ 0 & 1 & -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta + \omega_t \Delta t) \\ 0 & 0 & 1 \end{pmatrix}$$

$$V_t = \begin{pmatrix} \frac{-\sin\theta + \sin(\theta + \omega_t \Delta t)}{\omega_t} & \frac{v_t(\sin\theta - \sin(\theta + \omega_t \Delta t))}{\omega_t^2} + \frac{v_t \cos(\theta + \omega_t \Delta t)\Delta t}{\omega_t} \\ \frac{\cos\theta - \cos(\theta + \omega_t \Delta t)}{\omega_t} & -\frac{v_t(\cos\theta - \cos(\theta + \omega_t \Delta t))}{\omega_t^2} + \frac{v_t \sin(\theta + \omega_t \Delta t)\Delta t}{\omega_t} \\ 0 & \Delta t \end{pmatrix}$$

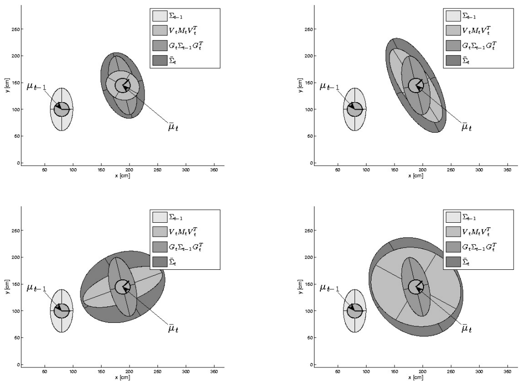
$$M_t = \begin{pmatrix} \alpha_1 v_t^2 + \alpha_2 \omega_t^2 & 0 \\ 0 & \alpha_3 v_t^2 + \alpha_4 \omega_t^2 \end{pmatrix}$$

$$\bar{\mu}_t = \mu_{t-1} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}$$

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

46

## EKF Prediction Step



47

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

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

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

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

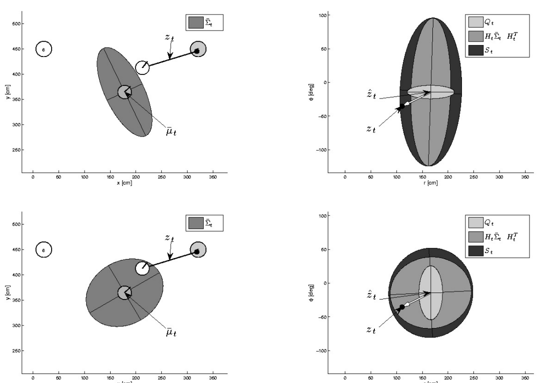
8. $K_t = \bar{\Sigma}_t H_t^T S_t^{-1}$ Kalman gain

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

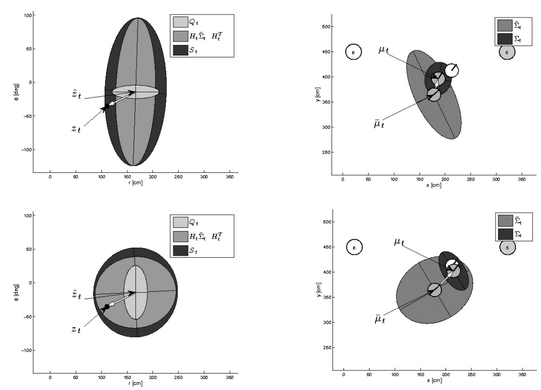
48

## EKF Observation Prediction Step



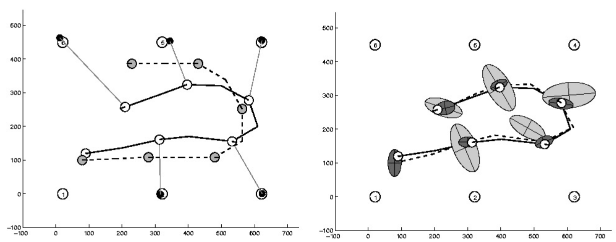
49

## EKF Correction Step



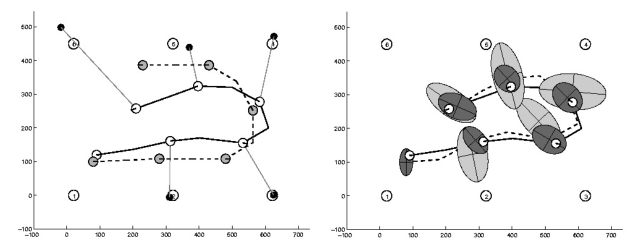
50

## Estimation Sequence (1)



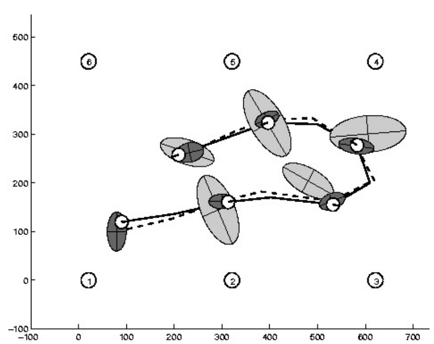
51

## Estimation Sequence (2)



52

## Comparison to GroundTruth




53

## 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!

54

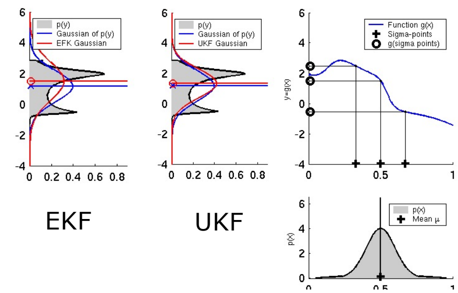
Going unscented

# UNSCENTED KALMAN FILTER

55

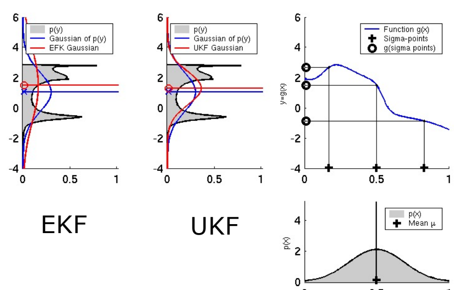
55

## Linearization via Unscented Transform



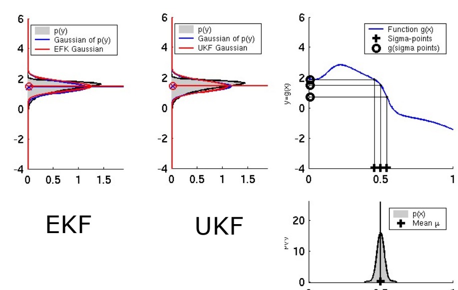

56

## UKF Sigma-Point Estimate (2)




57

## UKF Sigma-Point Estimate (3)




58

## Unscented Transform

Sigma points

$$\chi^0 = \mu$$

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

Weights

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

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

Pass sigma points through nonlinear function

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

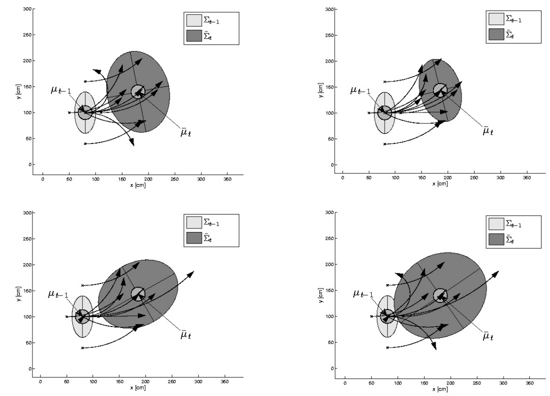
Recover mean and covariance

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

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

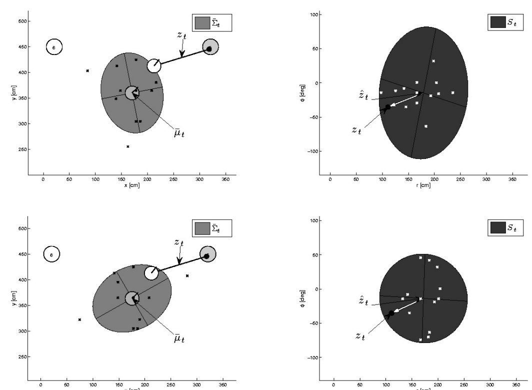
59

## UKF Prediction Step



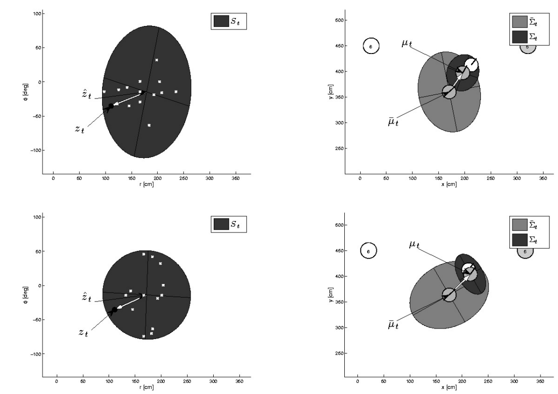
60

## UKF Observation Prediction Step



61

## UKF Correction Step



62

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

**Prediction:**

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

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

$$\mu_{t-1}^a = \begin{pmatrix} \mu_{t-1}^T & (0\,0)^T & (0\,0)^T \end{pmatrix}^T$$ 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

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

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

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

63

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

**Correction:**

$$\bar{Z}_t = h\left(\bar{\chi}_t^x\right) + \chi_t^z$$ Measurement sigma points

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

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

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

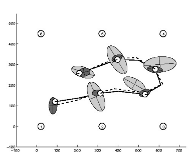
$$K_t = \Sigma_t^{x,z} S_t^{-1}$$ Kalman gain

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

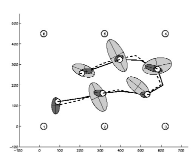
$$\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$$ Updated covariance

64

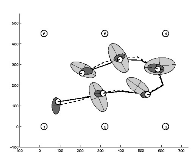
## Estimation Sequence



EKF



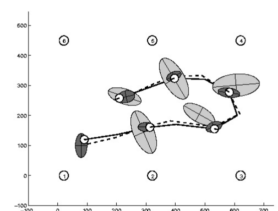
PF



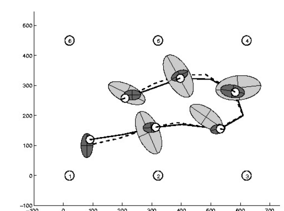
UKF

65

## Estimation Sequence



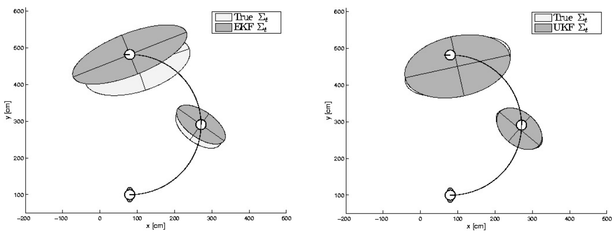
EKF



UKF

66

## Prediction Quality



EKF UKF

67

## UKF Summary

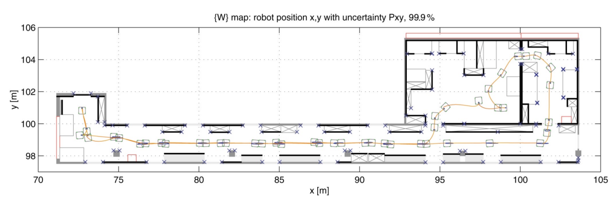
- Highly efficient: Same complexity as EKF, with a constant factor slower in typical practical applications
- Better linearization than EKF: Accurate in first two terms of Taylor expansion (EKF only first term)
- Derivative-free: No Jacobians needed
- Still not optimal!

68

68

## Kalman Filter-based System

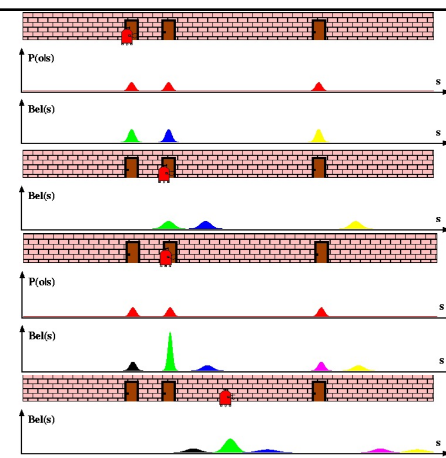
- [Arras et al. 98]:
  - Laser range-finder and vision
  - High precision (<1cm accuracy)




Courtesy of K. Arras

69

## Multi-hypothesis Tracking




70

## 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.

71

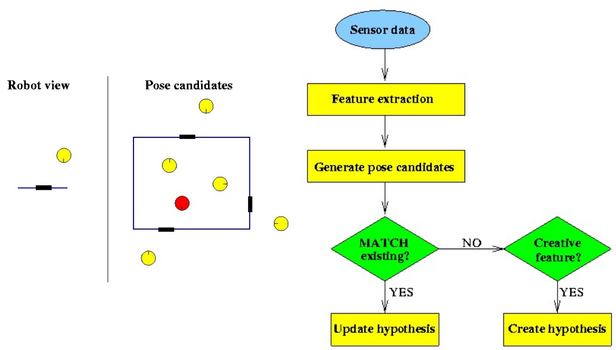
## MHT: Implemented System (1)

- 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]

72

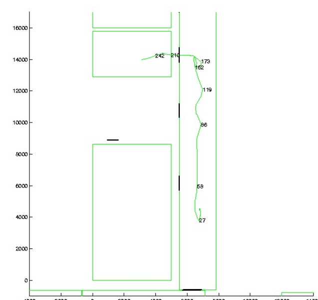
## MHT: Implemented System (2)



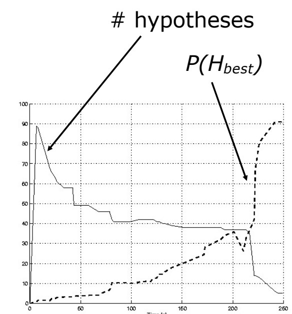

Courtesy of P. Jensfelt and S. Kristensen

73

## MHT: Implemented System (3)
## Example run



Map and trajectory




#hypotheses vs. time

Courtesy of P. Jensfelt and S. Kristensen

74