Cognitive Robotics

03. Kalman Filter Extensions

AbdElMoniem Bayoumi, PhD

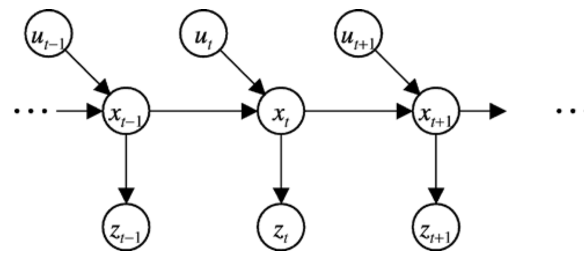
Spring 2022

Acknowledgment

- These slides have been created by Wolfram Burgard, Dieter Fox, Cyrill Stachniss and Maren Bennewitz

Previous Lecture

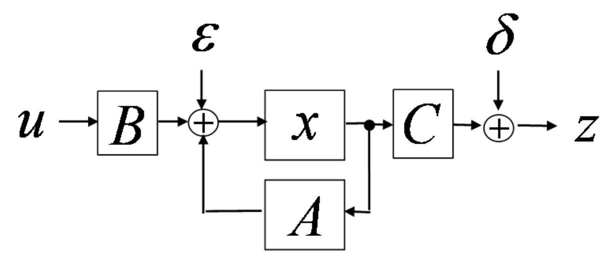
- Markov assumption



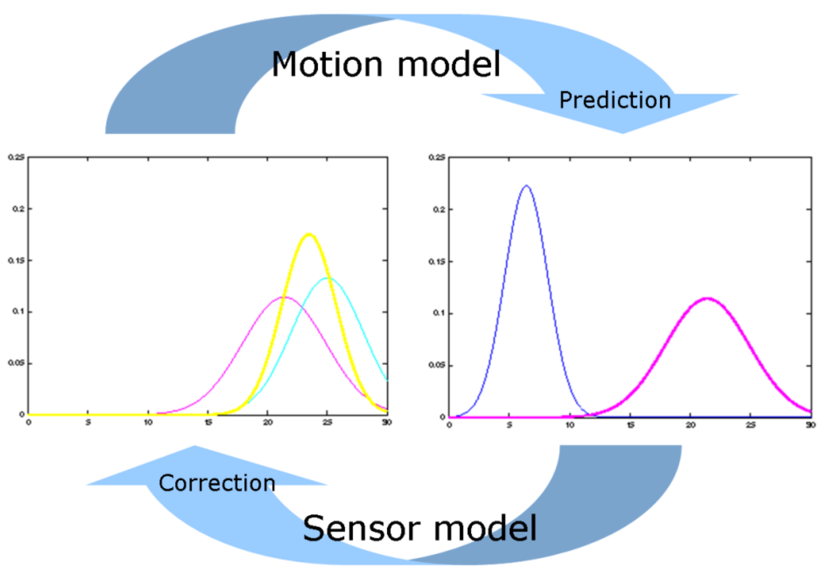
- Bayes filter

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

- Kalman filter
  - Linear systems
  - Gaussian noise
  - Recursive belief update



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



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

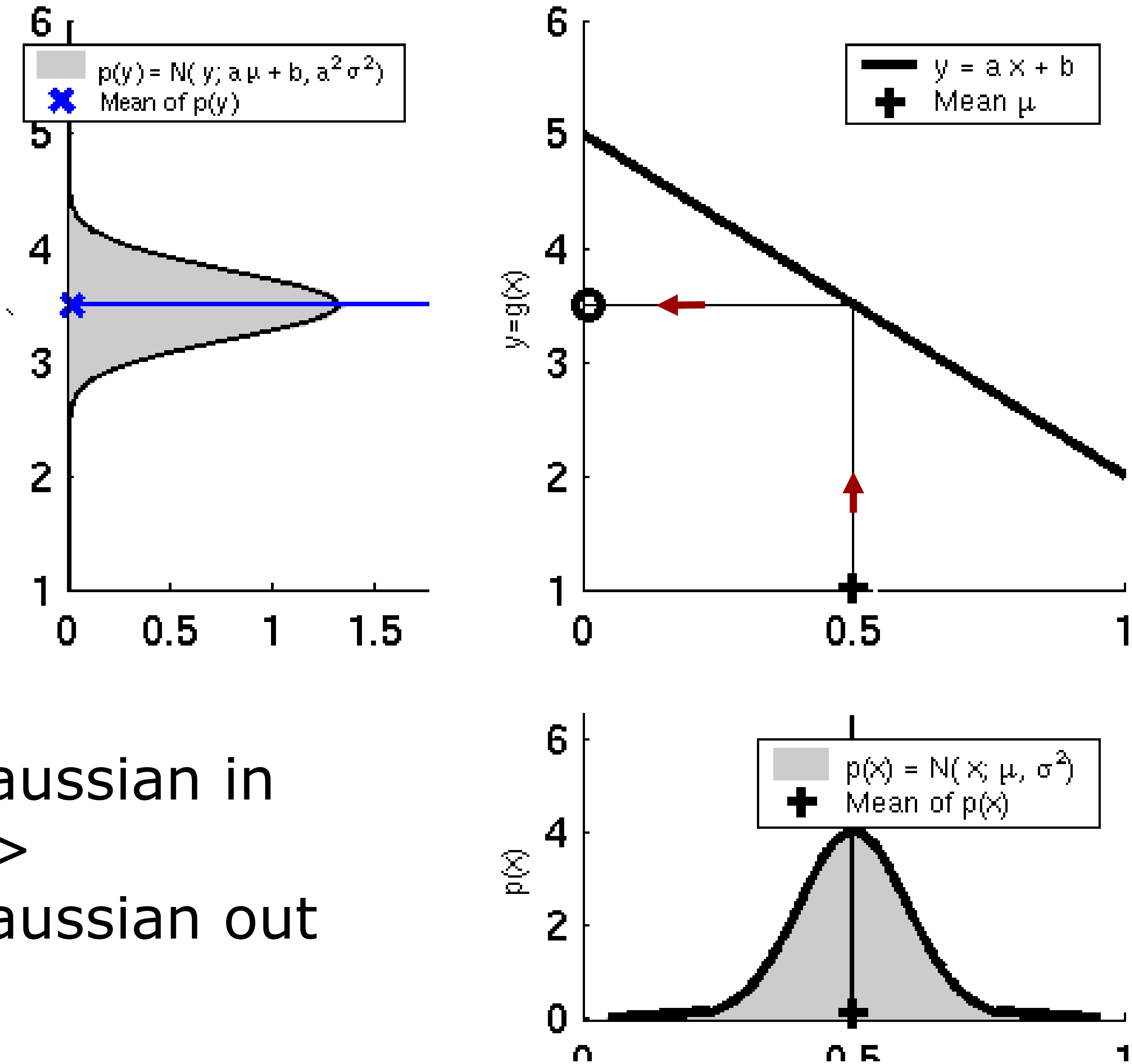
$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

Nonlinear Dynamic Systems

- Problem: Kalman filter restricted to linear systems

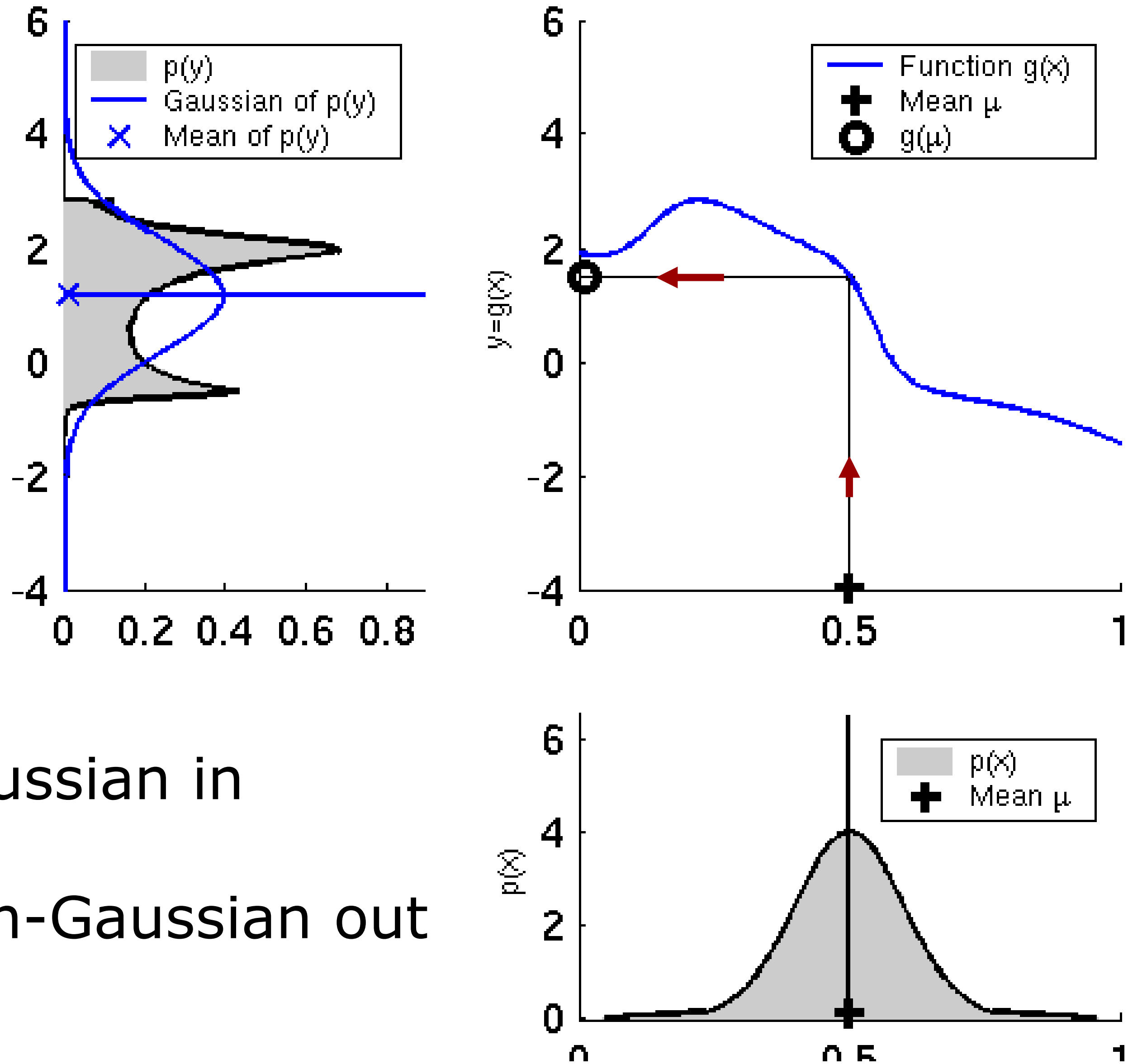
- Most realistic robotic problems involve nonlinear functions
  - Robot motion: $$\boxed{x_t = g(u_t, x_{t-1})}$$
  - Measurements: $$\boxed{z_t = h(x_t)}$$

Linearity Assumption Revisited



- Gaussian in
  =>
  Gaussian out

Non-linear Function



- Gaussian in
  =>
  Non-Gaussian out

Non-Gaussian Distributions

- The non-linear functions lead to non-Gaussian distributions
- Kalman filter is not applicable anymore!

What can be done to resolve this?

Non-Gaussian Distributions

- The non-linear functions lead to non-Gaussian distributions
- Kalman filter is not applicable anymore!

What can be done to resolve this?

Local linearization!

EKF Linearization: First Order Taylor 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:

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

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

Jacobian matrices

Reminder: Jacobian Matrix

- It is a **non-square matrix** $n \times m$ in general
- Given a vector-valued function

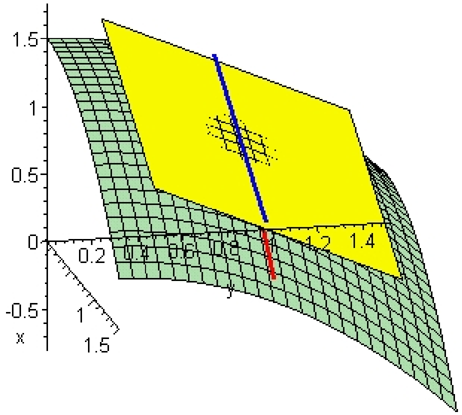
$$f(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}$$

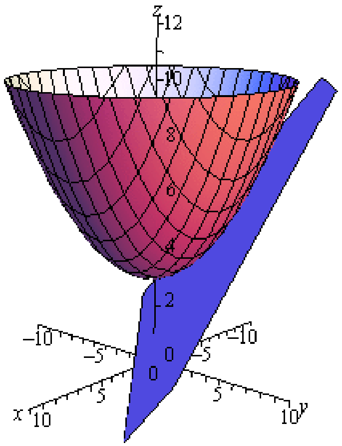
- The **Jacobian matrix** is defined as

$$\mathbf{F_x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Reminder: Jacobian Matrix

- It is the orientation of the tangent plane to the vector-valued function at a given point





- Generalizes the gradient of a scalar valued function

EKF Linearization: First Order Taylor Expansion

- Prediction:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} (x_{t-1} - \mu_{t-1})$$

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t \, (x_{t-1} - \mu_{t-1})$$

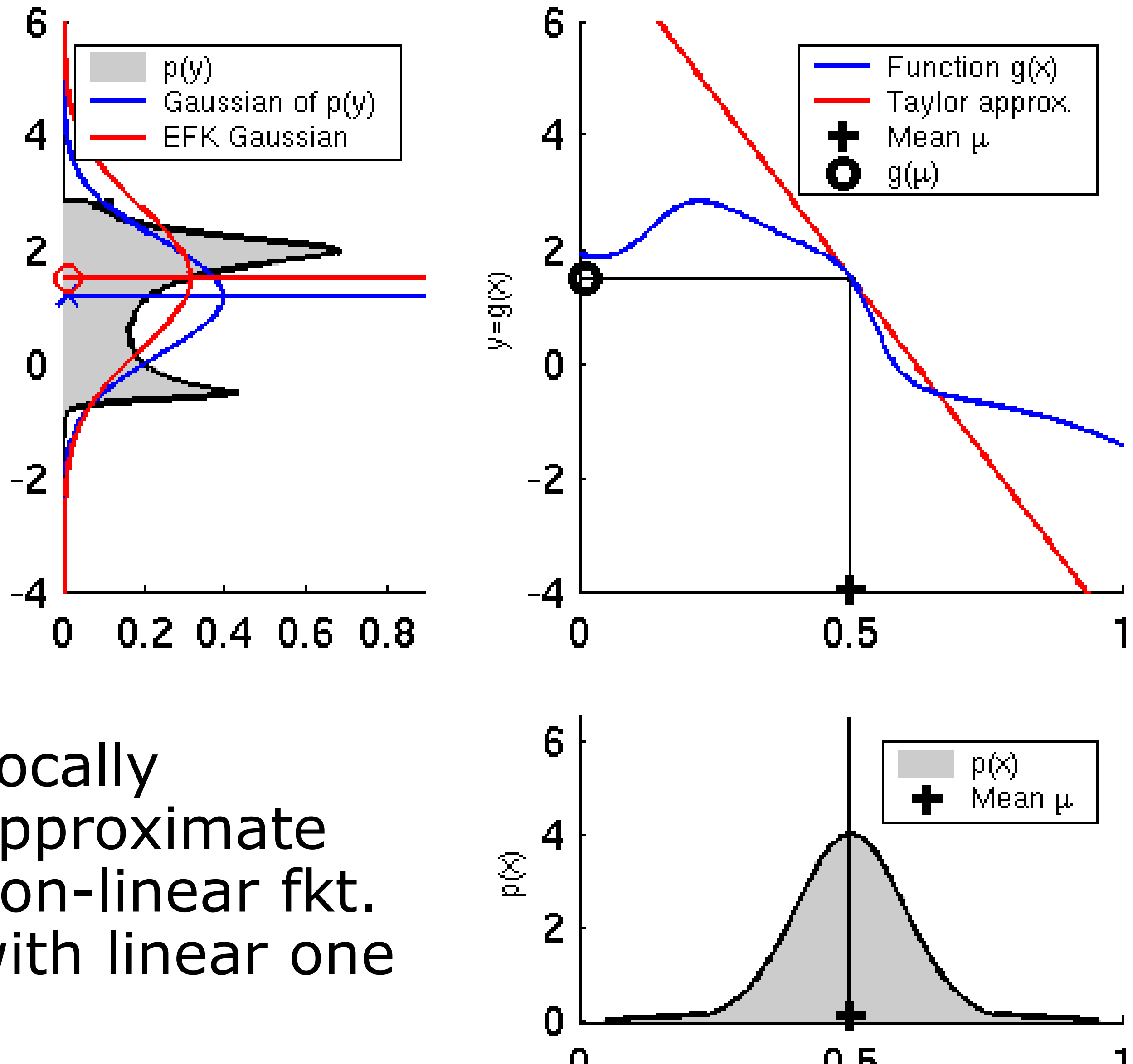
- Correction:

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

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

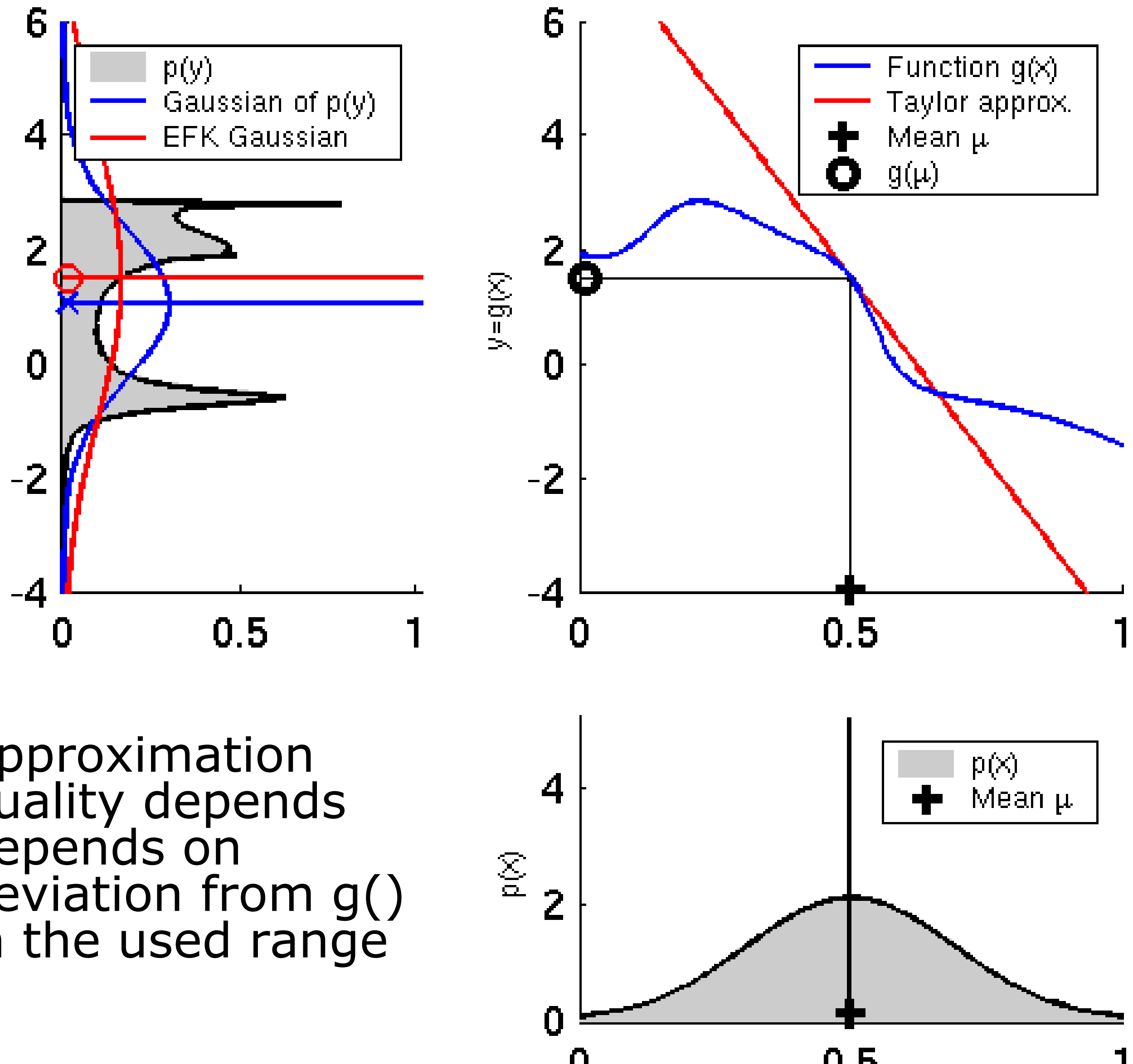
Linear function!

EKF Linearization (1)



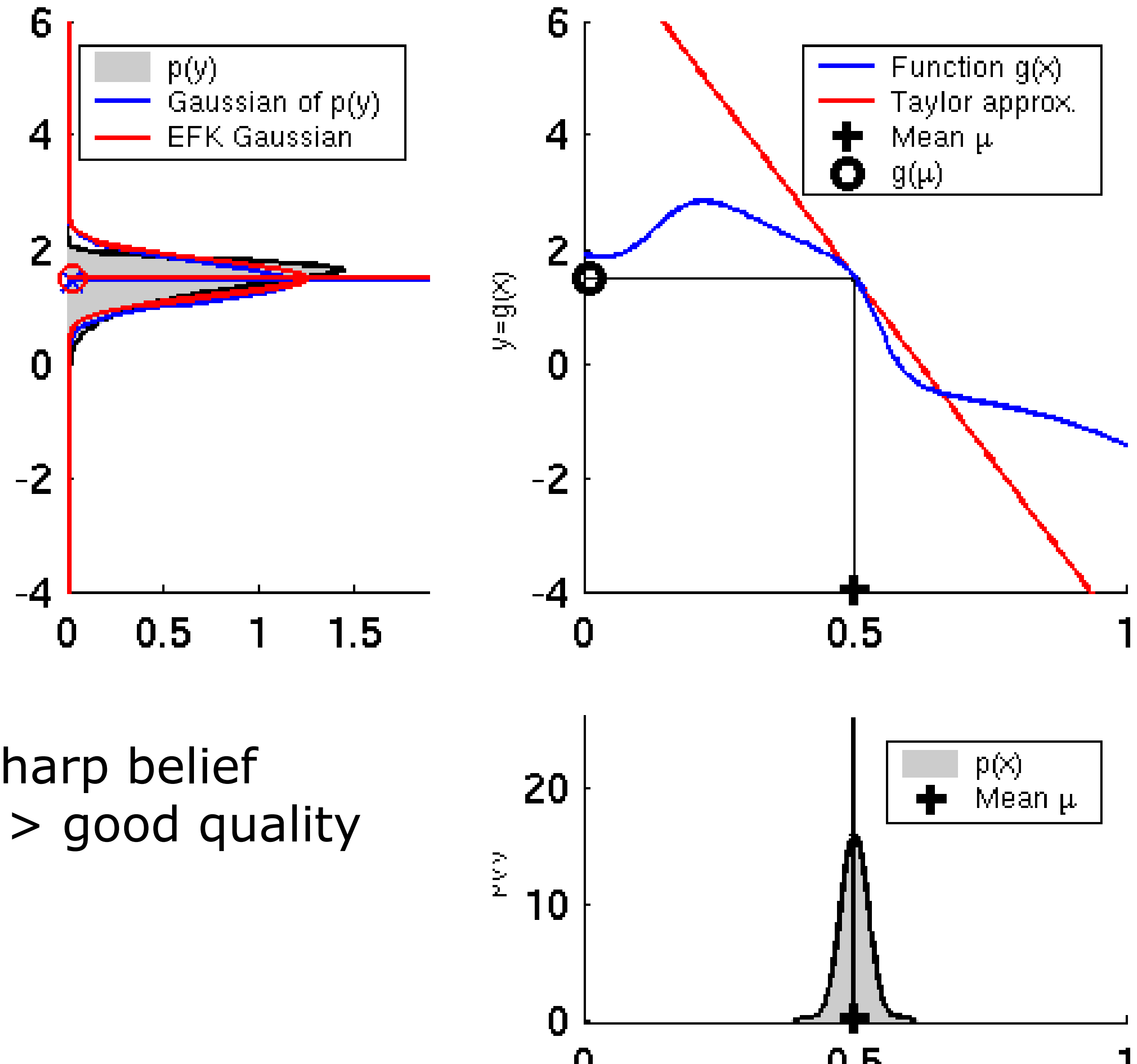
- Locally approximate non-linear fkt. with linear one

EKF Linearization (2)



- Approximation quality depends depends on deviation from g() in the used range

EKF Linearization (3)



- Sharp belief
  => good quality

EKF Algorithm

1. **Extended_Kalman_filter**(μ_{t-1}, Σ_{t-1}, u_t, z_t):

2. Prediction:

Kalman filter

3. $\bar{\mu}_t = g(u_t, \mu_{t-1})$ $\longleftarrow$ $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$

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

$$G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$$

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

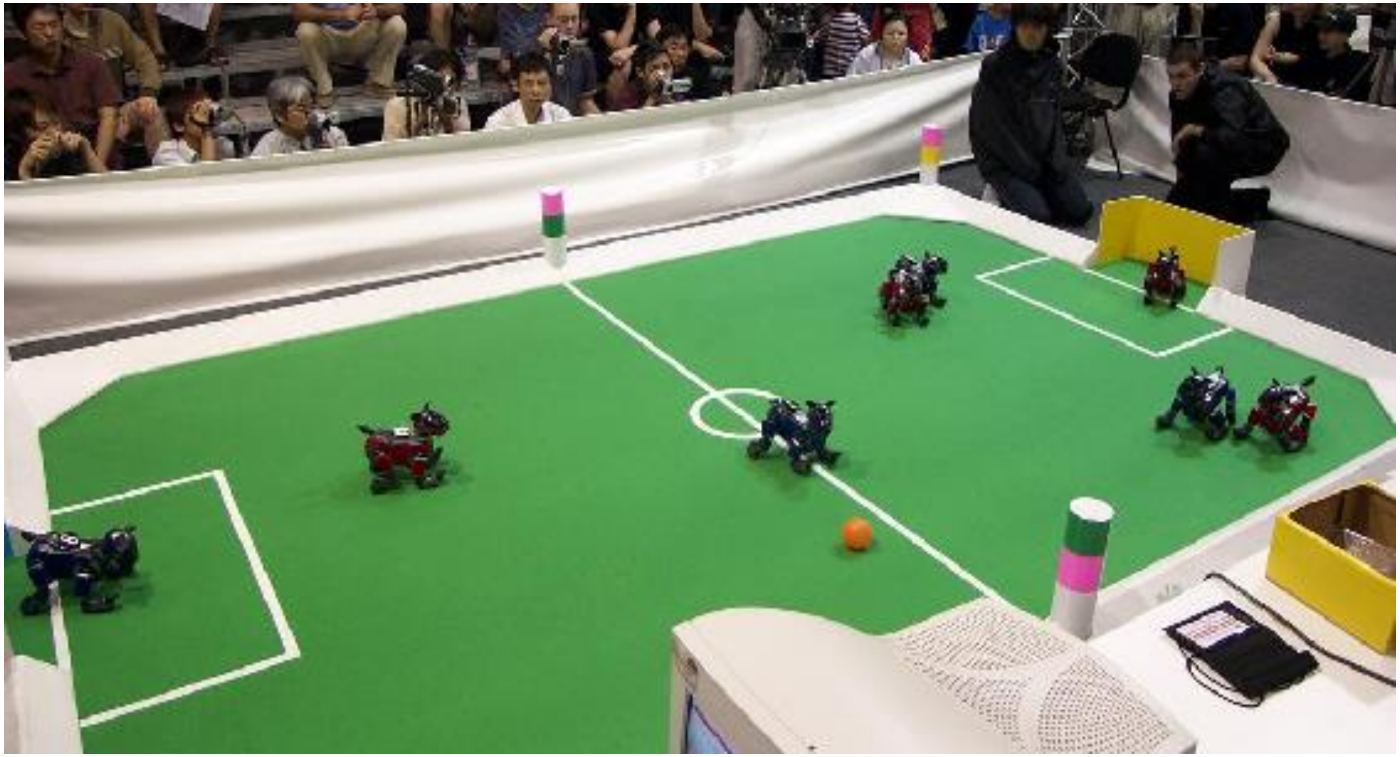
$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t}$$

EKF Example: 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 (initial pose known)
  - Global localization (initial pose unknown)
  - Kidnapped robot problem (recovery)

Landmark-based Localization



- Goal: Estimate robot pose $\mu_t=(x, y, \theta)$ and its covariance Σ_t
- Given: Map m with landmark positions
- Control u_t: Forward speed v, rotational speed ω
- Observations z_t: Angle and distance of landmarks

EKF_localization (μ_{t-1}, Σ_{t-1}, u_t, z_t, m):

Prediction:

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

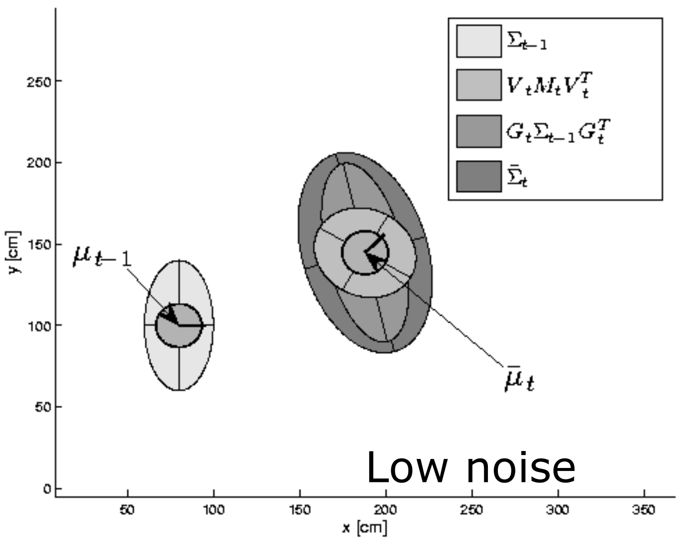
3. $$V_t = \frac{\partial g(u_t,\mu_{t-1})}{\partial u_t} = \begin{pmatrix} \frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial \omega_t} \\ \frac{\partial y'}{\partial v_t} & \frac{\partial y'}{\partial \omega_t} \\ \frac{\partial \theta'}{\partial v_t} & \frac{\partial \theta'}{\partial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

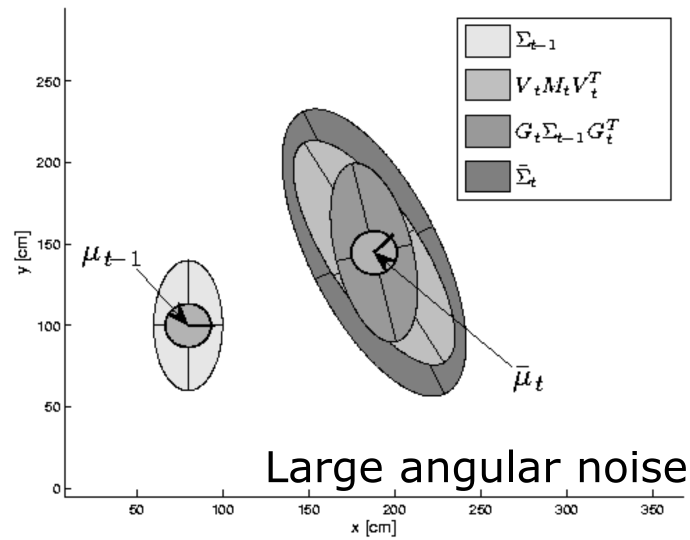
4. $$M_t = \begin{pmatrix} (\alpha_1 |v_t| + \alpha_2 |\omega_t|)^2 & 0 \\ 0 & (\alpha_3 |v_t| + \alpha_4 |\omega_t|)^2 \end{pmatrix}$$ Motion noise

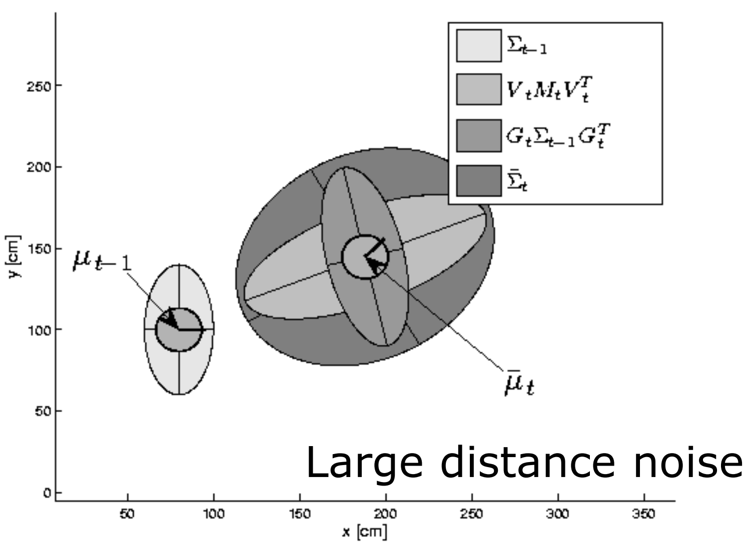
5. $\bar{\mu}_t = g(u_t, \mu_{t-1})$ Predicted mean

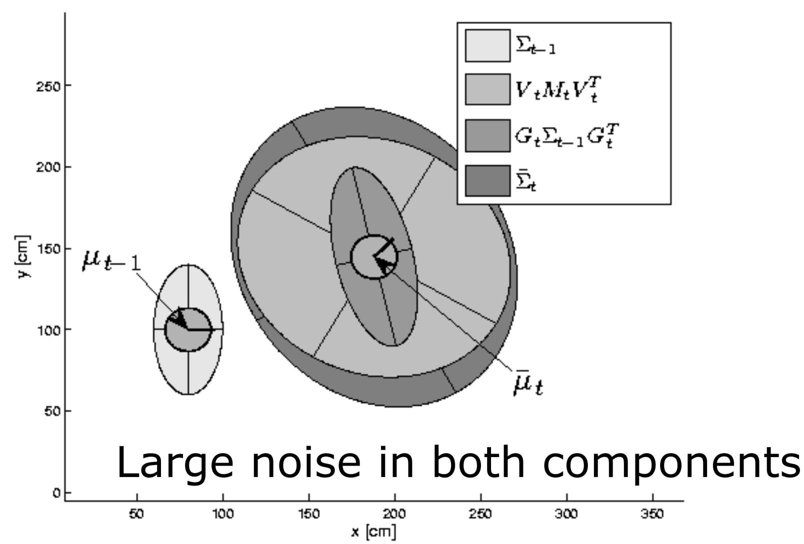
6. $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T$ Predicted covariance

EKF Prediction Step









EKF_localization (μ_{t-1}, Σ_{t-1}, u_t, z_t, m):

(distance, angle to landmark)

Correction:

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

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

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

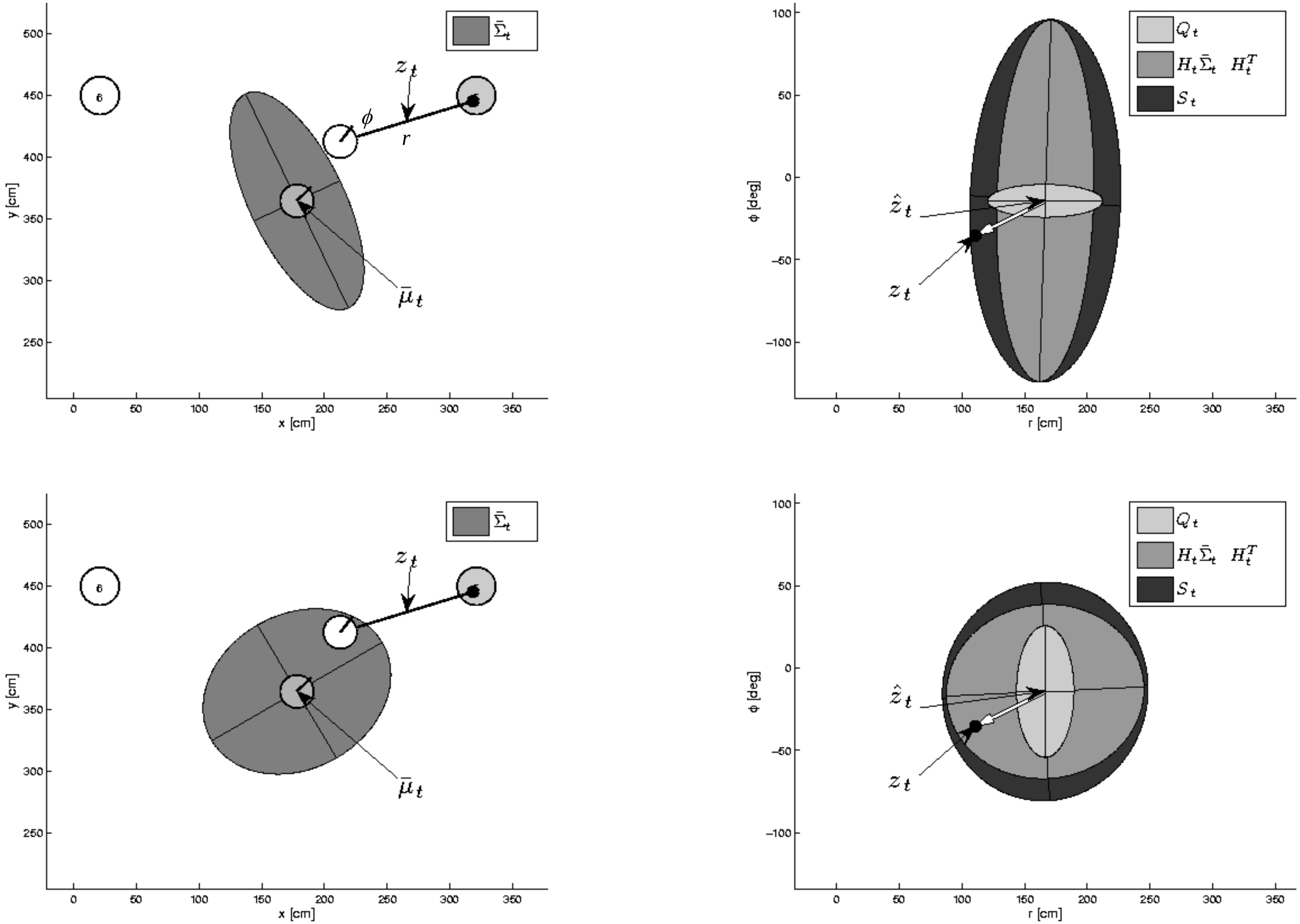
6. $$S_t = H_t \bar{\Sigma}_t H_t^T + Q_t$$ Pred. measurement covariance

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

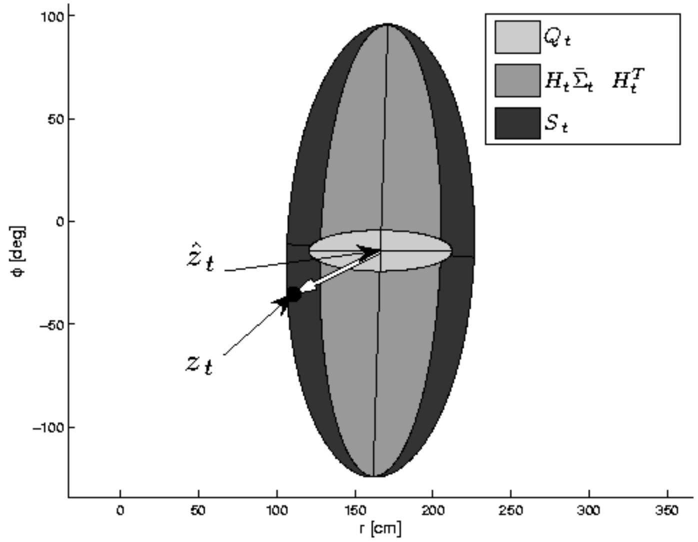
8. $$\mu_t = \bar{\mu}_t + K_t (z_t - \hat{z}_t)$$ Updated mean

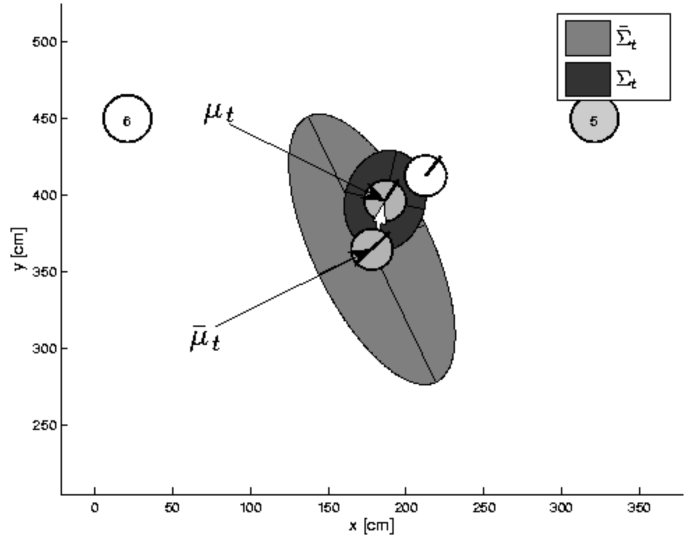
9. $$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$$ Updated covariance

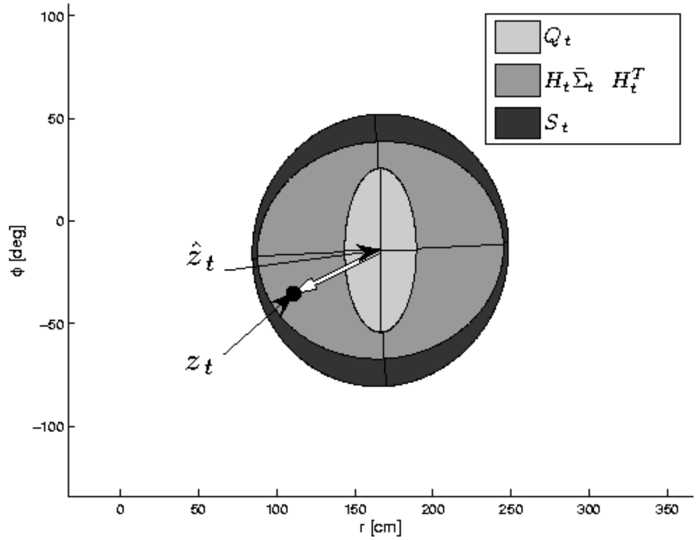
EKF Observation Prediction Step

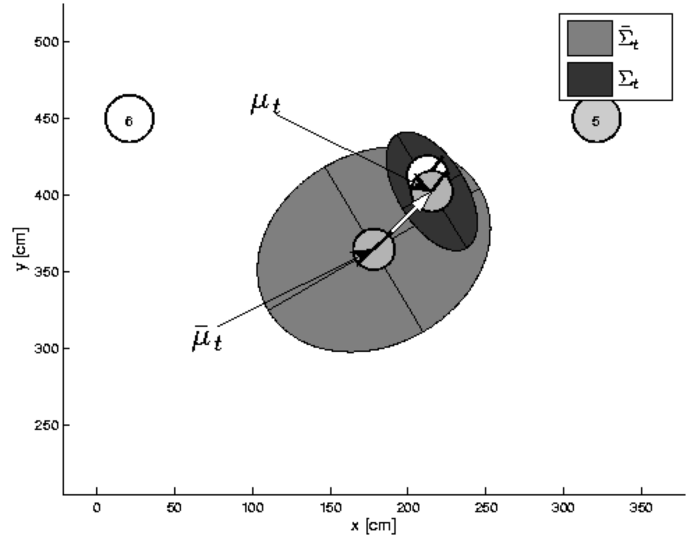


EKF Correction Step

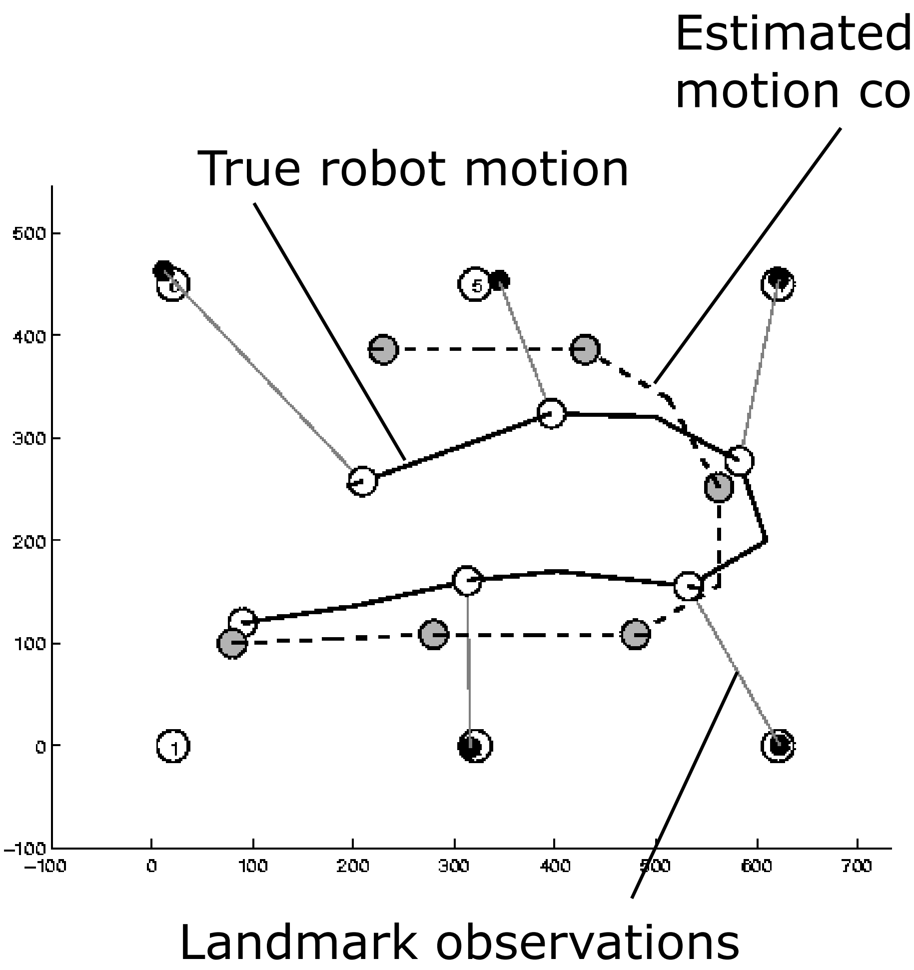


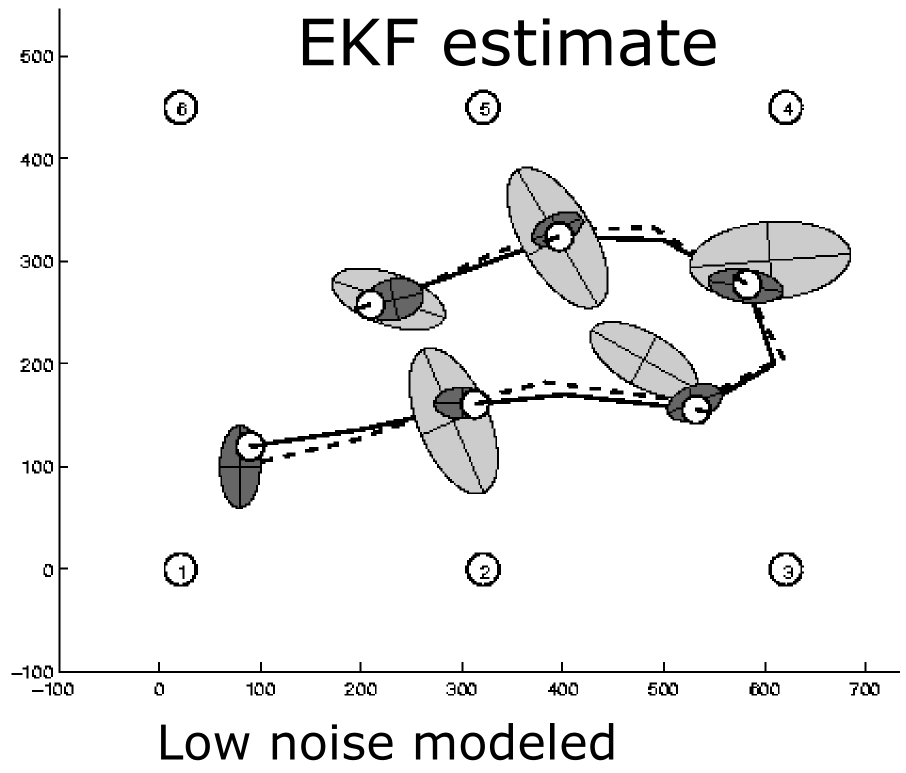




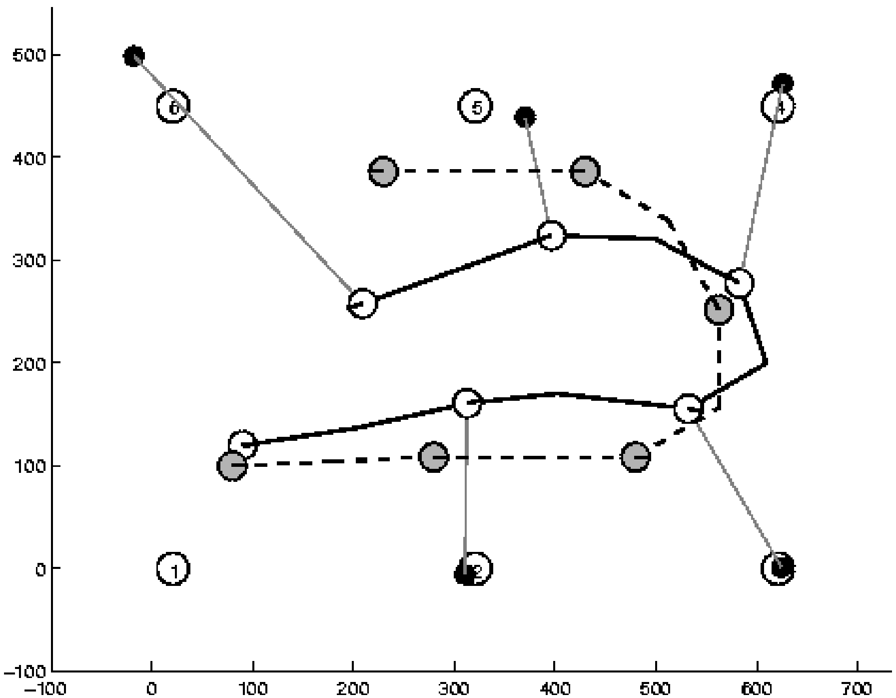


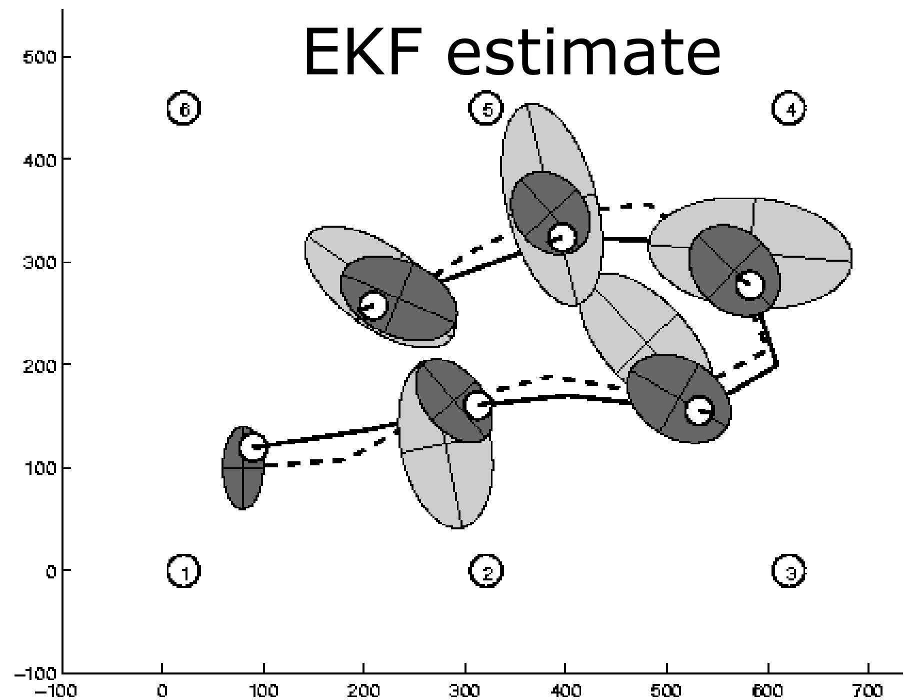
Estimation Sequence (1)





Estimation Sequence (2)





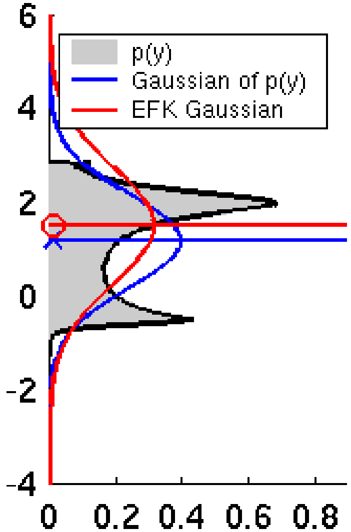
Larger noise modeled

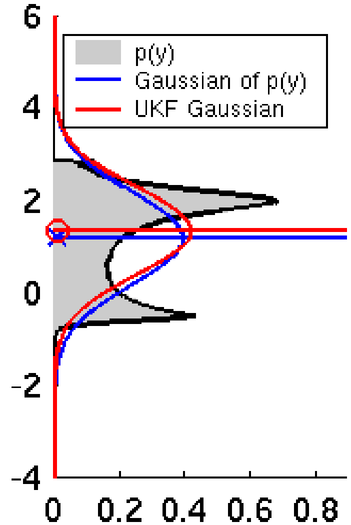
EKF Summary

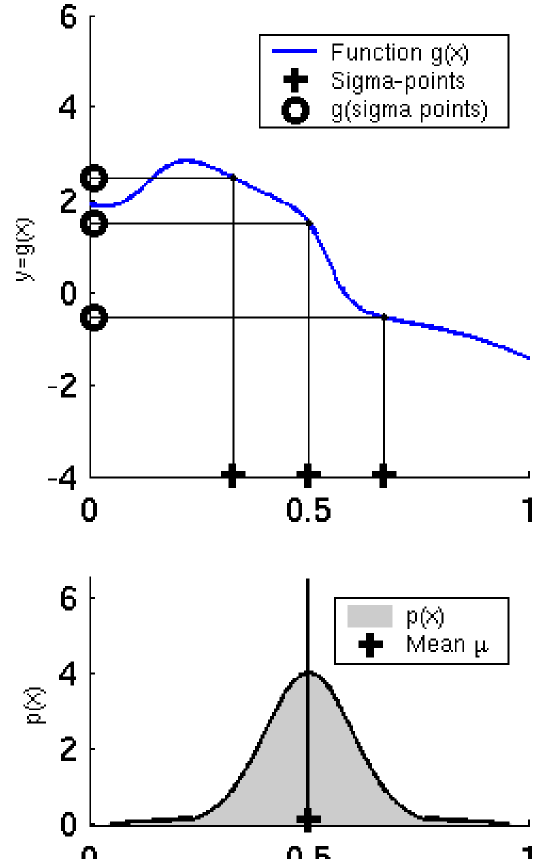
- Highly efficient: Polynomial in measurement dimensionality k and state dimensionality n:
$$O(k^{2.376} + n^2)$$

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

Linearization via Unscented Transform





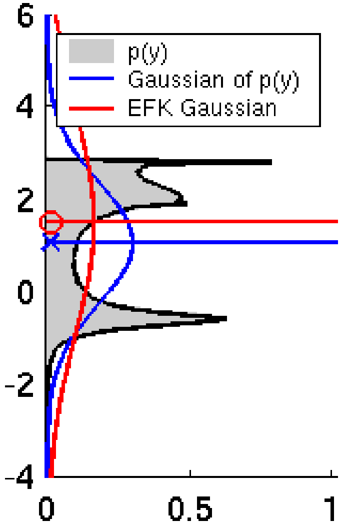


EKF

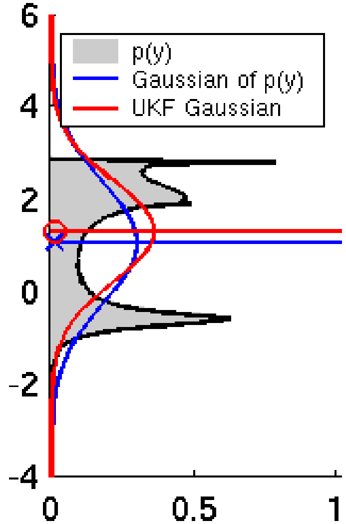
UKF

- Represent belief by Sigma-points

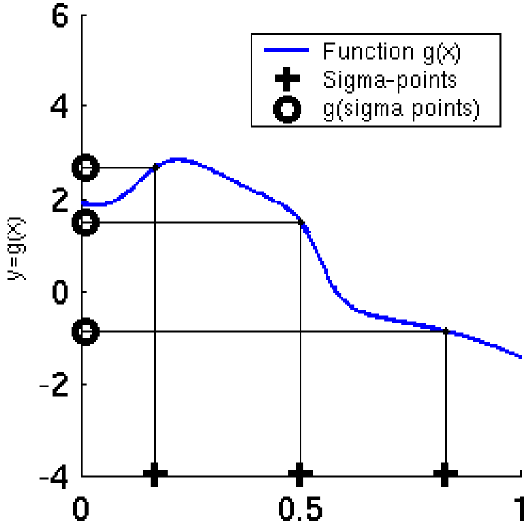
UKF Sigma-Point Estimate (2)

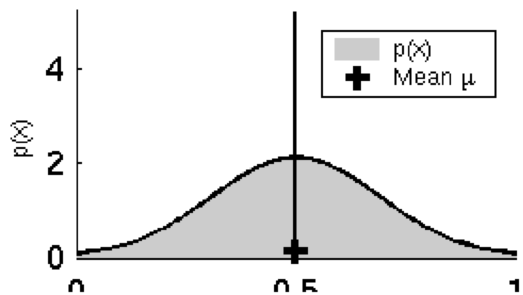


EKF

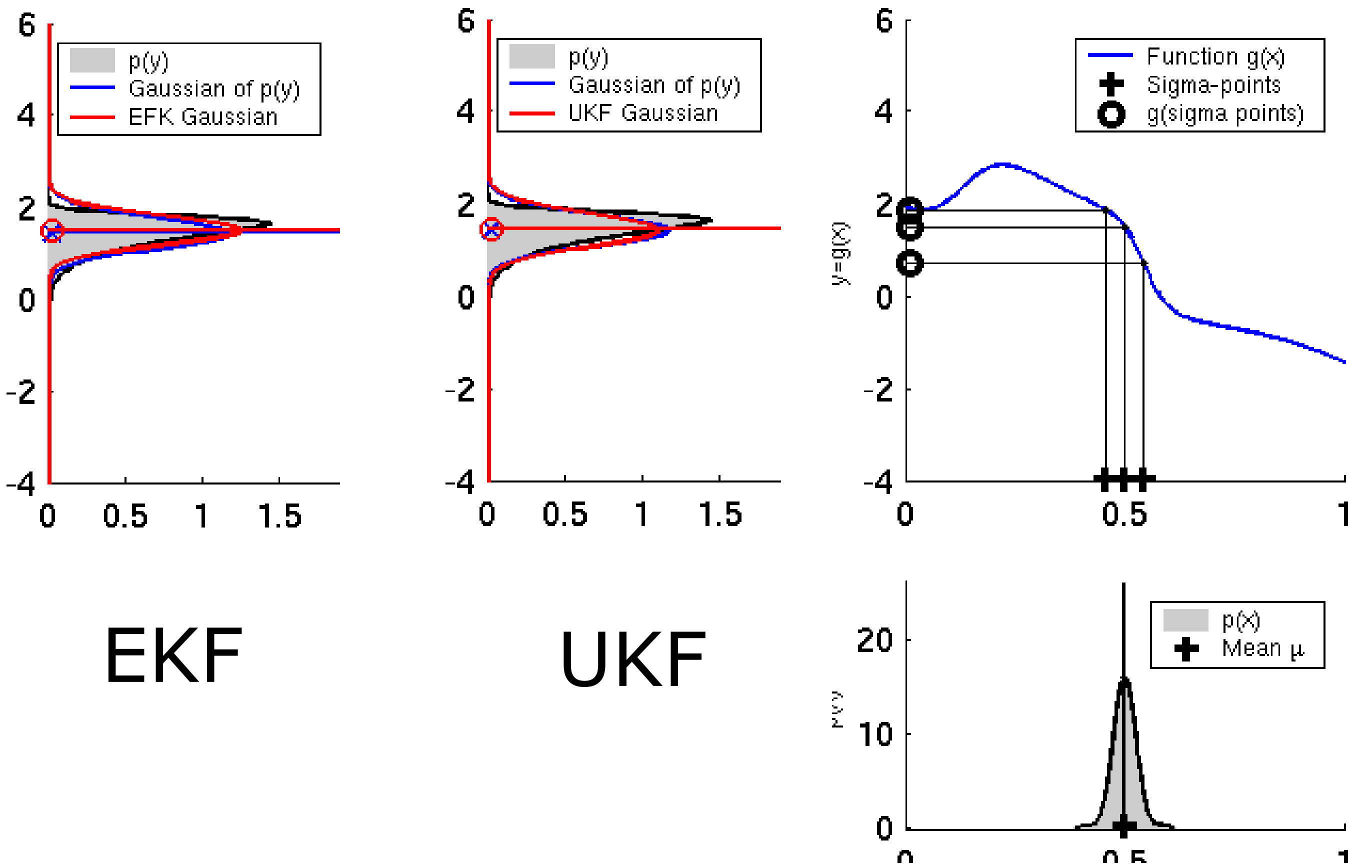


UKF





UKF Sigma-Point Estimate (3)



Unscented Transform

Sigma points

$$\chi^0 = \mu$$

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

Weights

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

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

$$\lambda = \alpha^2(n+\kappa) - n$$

- 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 (μ_{t-1}, Σ_{t-1}, u_t, z_t, m):

Prediction:

$$\chi_{t-1}^{a} = \begin{pmatrix} \chi_{t-1}^{x\;T} \\ \chi_{t}^{u\;T} \\ \chi_{t}^{z\;T} \end{pmatrix}$$

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

Motion noise

Depends on forward speed and rotational speed

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

Measurement noise

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

Augmented state mean

x,y,θ , motion noise, measurement noise

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

Augmented covariance 7 x 7

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

15 Sigma points

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

Prediction of sigma points

$$\bar{\mu}_t = \sum_{i=0}^{2L} w_m^i \, \bar{\chi}_{i,t}^{x}$$

Predicted mean

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

Predicted covariance

Sigma Points of Augmented States

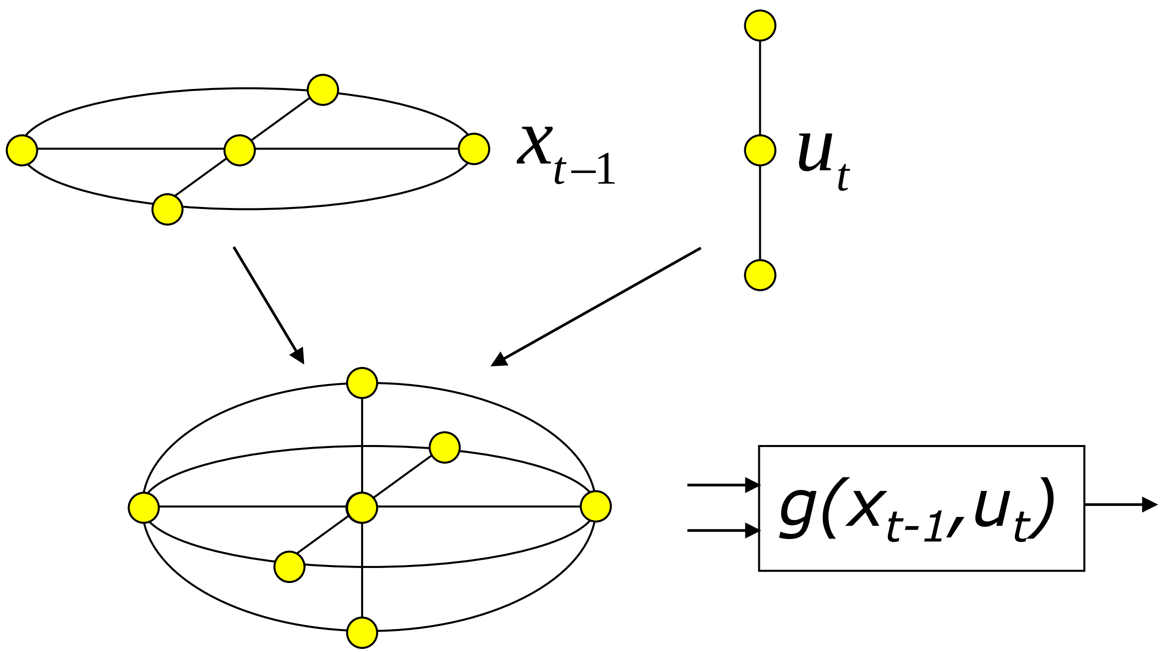
- χ_{t-1}^{a} is a sigma point representation of the augmented state estimate

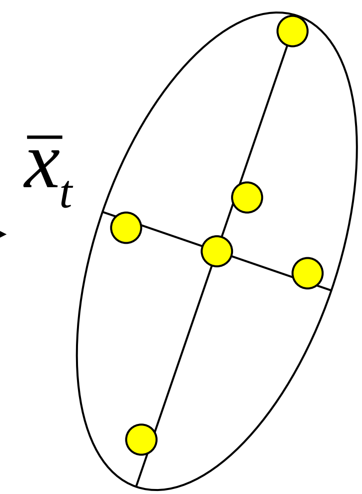
$$\chi_{t-1}^{a} = \begin{pmatrix} {\chi_{t-1}^{x}}^{T} \\ {\chi_{t}^{u}}^{T} \\ {\chi_{t}^{z}}^{T} \end{pmatrix}$$

- χ_{t-1}^{a} contains *2L+1 = 15* sigma points, each having components in state, control and measurement space

Unscented Prediction

- Construct a *N+M* dimensional Gaussian from the previous state distribution and the controls





For each of the *2(N+M)+1* samples $\langle x,u \rangle^i$ compute its mapping via $g(x,u)$

Recover a Gaussian approximation from the samples

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

Correction:

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

Prediction of 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

Between state and observation

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

Kalman gain

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

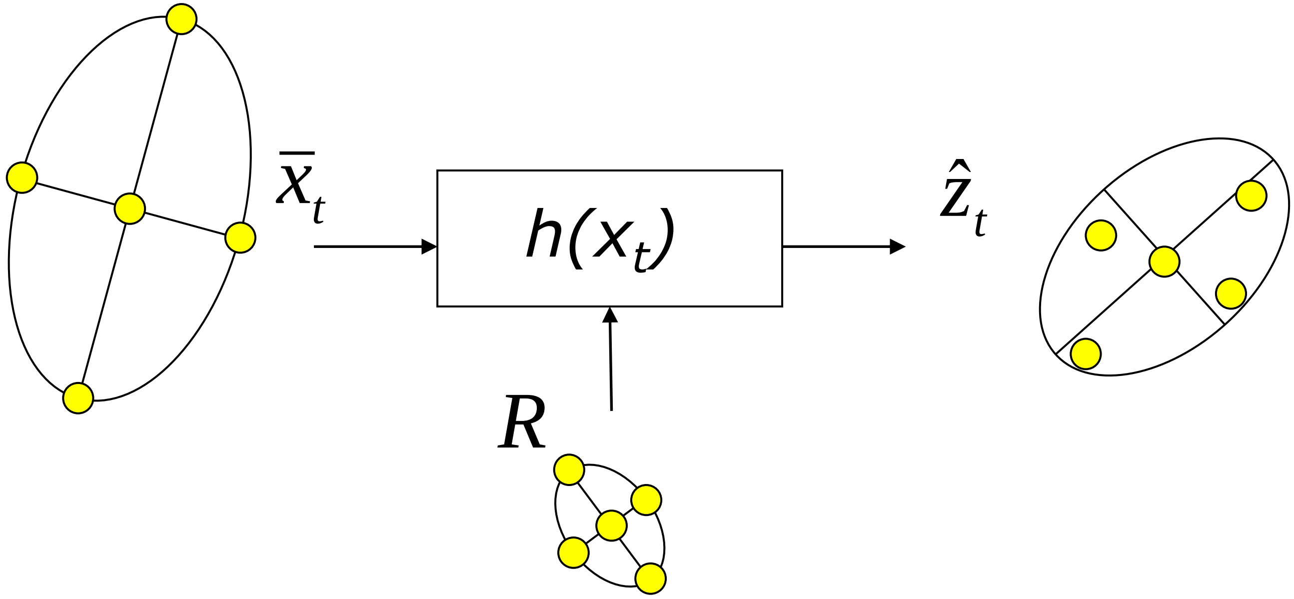
Updated mean

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

Updated covariance

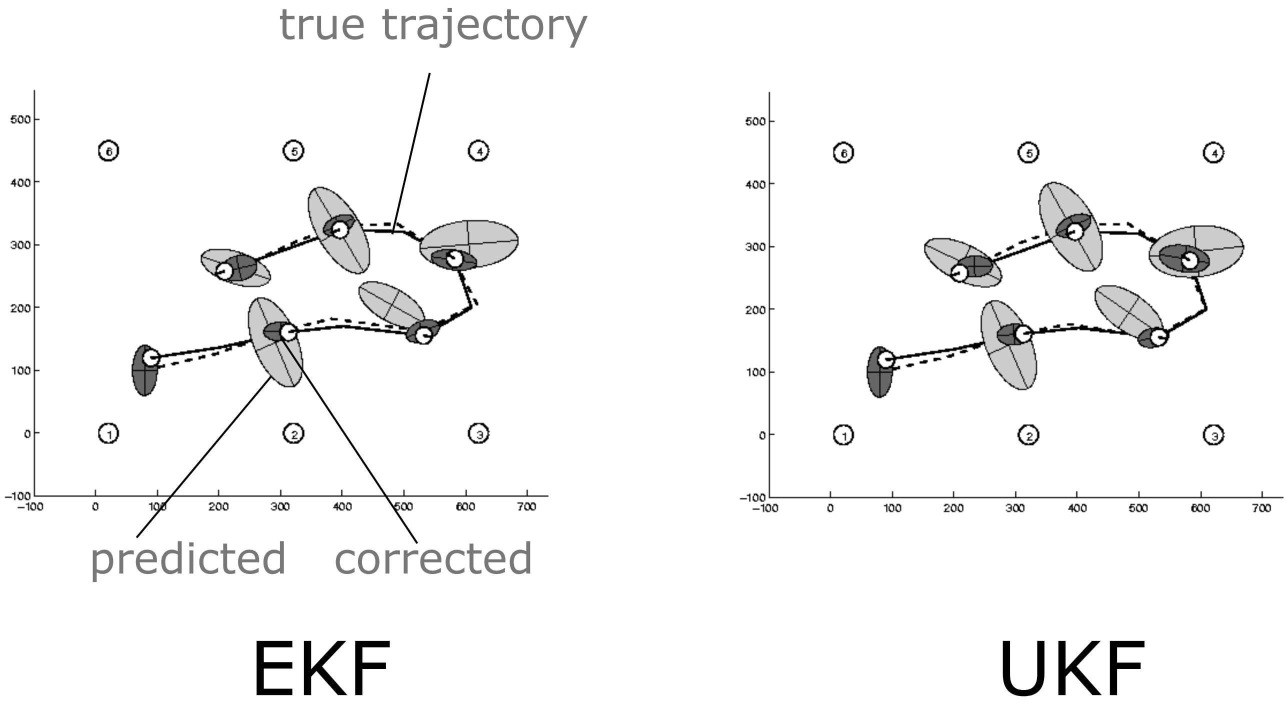
Unscented Correction

- Sample from the predicted state and the observation noise, to obtain the expected measurement



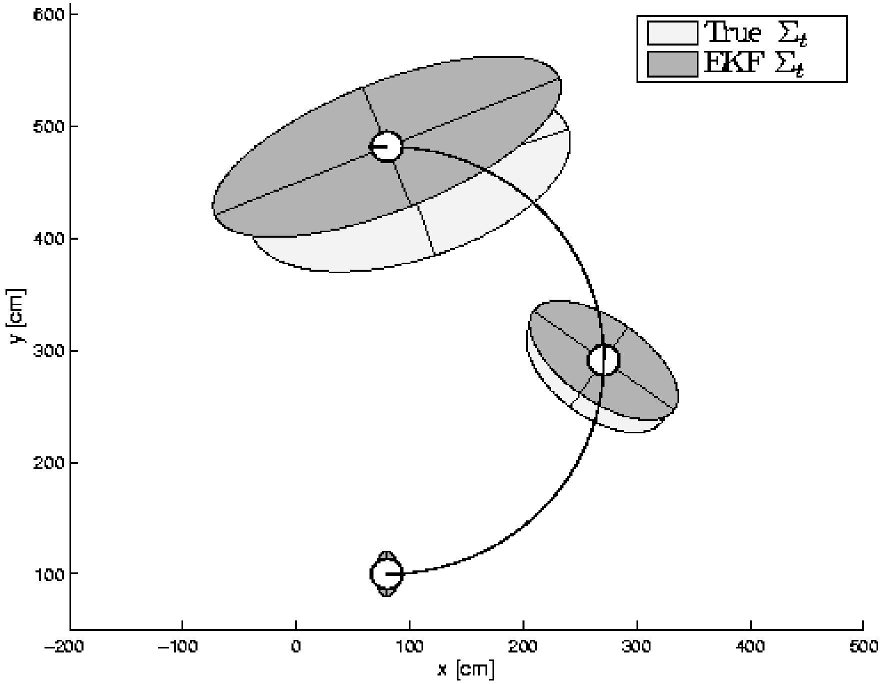
- Compute the cross correlation matrix of measurements and states, and perform a Kalman update

Estimation Sequence

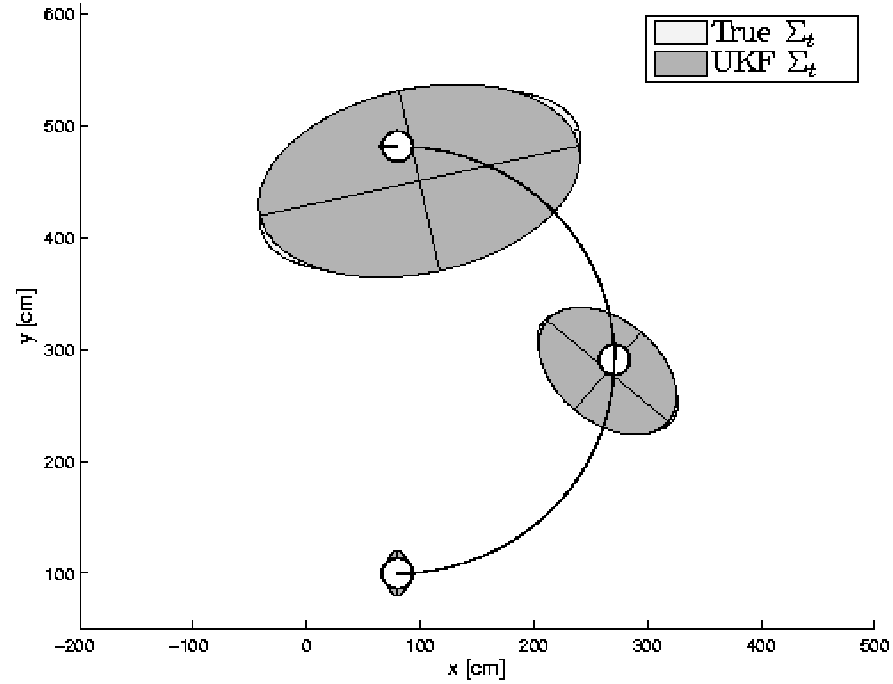


Prediction Quality

Two motion steps without observations



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!

Acknowledgment

- These slides have been created by Wolfram Burgard, Dieter Fox, Cyrill Stachniss and Maren Bennewitz