Principles of Robot Autonomy I

Parameteric filtering

Today's lecture

- Aim
  - Learn about parametric filters

- Readings
  - S. Thrun, W. Burgard, and D. Fox. Probabilistic robotics. MIT press, 2005. Sections 3.1 – 3.4, 4.1, 4.3, 7.1

Instantiating the Bayes' filter

- Tractable implementations of Bayes' filter exploit structure and / or approximations; two main classes
  - Parametric filters: e.g., KF, EKF, UKF, etc.
  - Non parametric filters: e.g., histogram filter, particle filter, etc.

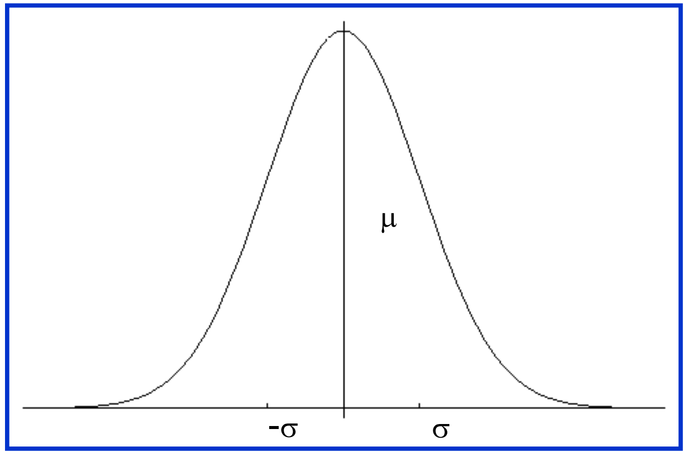
Gaussian distributions

- Key idea: belief represented as multivariate normal distribution

Univariate

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

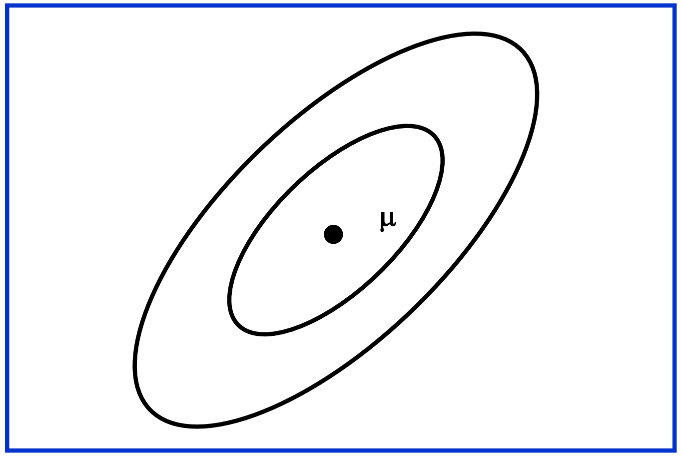
$$\sim \mathcal{N}(x; \mu, \sigma^2)$$



Multivariate

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

$$\sim \mathcal{N}(\mu, \Sigma)$$



Key properties of Gaussian random variables

- If $X \sim \mathcal{N}(\mu, \Sigma)$ then

$$Y = AX + b \sim \mathcal{N}(A\mu + b, A\Sigma A^T)$$

- The sum of two independent Gaussian RVs

$$X_i \sim \mathcal{N}(\mu_i, \Sigma_i), \qquad i = 1, 2$$

is Gaussian, specifically

$$X_1 + X_2 \sim \mathcal{N}(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$$

- The product of Gaussian pdf is also Gaussian

Kalman filter (KF)

- Assumption #1: linear dynamics

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

  - Independent process noise ϵ_t is $\mathcal{N}(0, R_t)$
- Assumption #1 implies that the probabilistic generative model is Gaussian

$$p(x_t \mid u_t, x_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1}(x_t - A_t x_{t-1} - B_t u_t)\right)$$

Kalman filter (KF)

- Assumption #2: linear measurement model

$$z_t = C_t x_t + \delta_t$$

  - Independent measurement noise δ_t is $\mathcal{N}(0, Q_t)$
- Assumption #2 implies that the measurement probability is Gaussian

$$p(z_t \mid x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t)\right)$$

Kalman filter (KF)

- Assumption #3: the initial belief is Gaussian

$$bel(x_0) = p(x_0) = \det(2\pi\Sigma_0)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1}(x_0 - \mu_0)\right)$$

- Key fact: These three assumptions ensure that the posterior $bel(x_t)$ is Gaussian for all t, i.e., $bel(x_t) = \mathcal{N}(\mu_t, \Sigma_t)$
- Note:
  - KF implements belief computation for continuous states
  - Gaussians are unimodal -> commitment to single-hypothesis filtering

Kalman filter: algorithm

Prediction

Project state ahead

$$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$$

Project covariance ahead

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

Correction

Compute Kalman gain

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

Update estimate with new measurement

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

Update covariance

$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

$bel(x_{t-1})$

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

Result: (μ_t, Σ_t)

Prediction: $\overline{bel}(x_t)$

$$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \; ;$$
$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t ;$$

Correction: $bel(x_t)$

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1} ;$$
$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) ;$$
$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t ;$$

Return (μ_t, Σ_t)

$bel(x_t)$

Kalman filter: derivation (sketch)

- Prediction

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

$$\downarrow \qquad\qquad\qquad\qquad \downarrow$$

$$\mathcal{N}(A_t x_{t-1} + B_t u_t,\, R_t) \qquad\qquad \mathcal{N}(\mu_{t-1},\, \Sigma_{t-1})$$

- Recalling that $x_t = A_t x_{t-1} + B_t u_t + \epsilon_t$

$$\overline{bel}(x_t) = \mathcal{N}(\overline{\mu}_t,\, \overline{\Sigma}_t)$$

with

$$\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t$$

$$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

Kalman filter: derivation (sketch)

- Correction

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

$$p(z_t \mid x_t) \rightarrow \mathcal{N}(Cx_t, Q_t) \qquad \overline{bel(x_t)} \rightarrow \mathcal{N}(\overline{\mu}_t, \overline{\Sigma}_t)$$

- After some algebraic manipulations

$$bel(x_t) = \mathcal{N}(\mu_t, \Sigma_t)$$

with

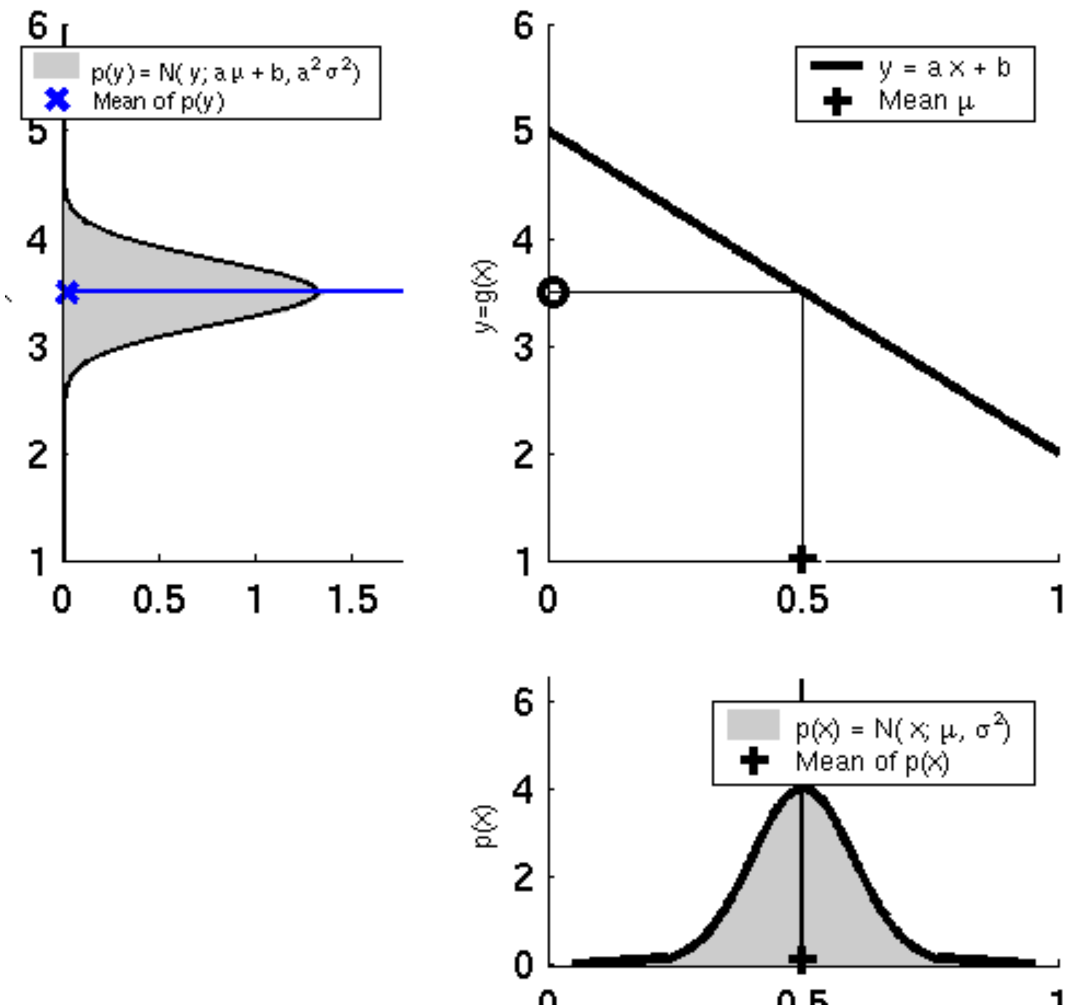
$$K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$$

$$\mu_t = \overline{\mu}_t + K_t(z_t - C_t \overline{\mu}_t)$$

$$\Sigma_t = (I - K_t C_t)\overline{\Sigma}_t$$

- Other derivations are possible; see, e.g., R. E. Kalman, A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 82(1), 35-45, 1960.

Revisiting linearity assumption



- KF crucially exploits the property that a linear transformation of a Gaussian RV results in a Gaussian RV
- However, linearity assumptions are severely restrictive for robotics applications

Extended Kalman filter (EKF)

- Goal: relax the linearity assumption
- The dynamics are now given by

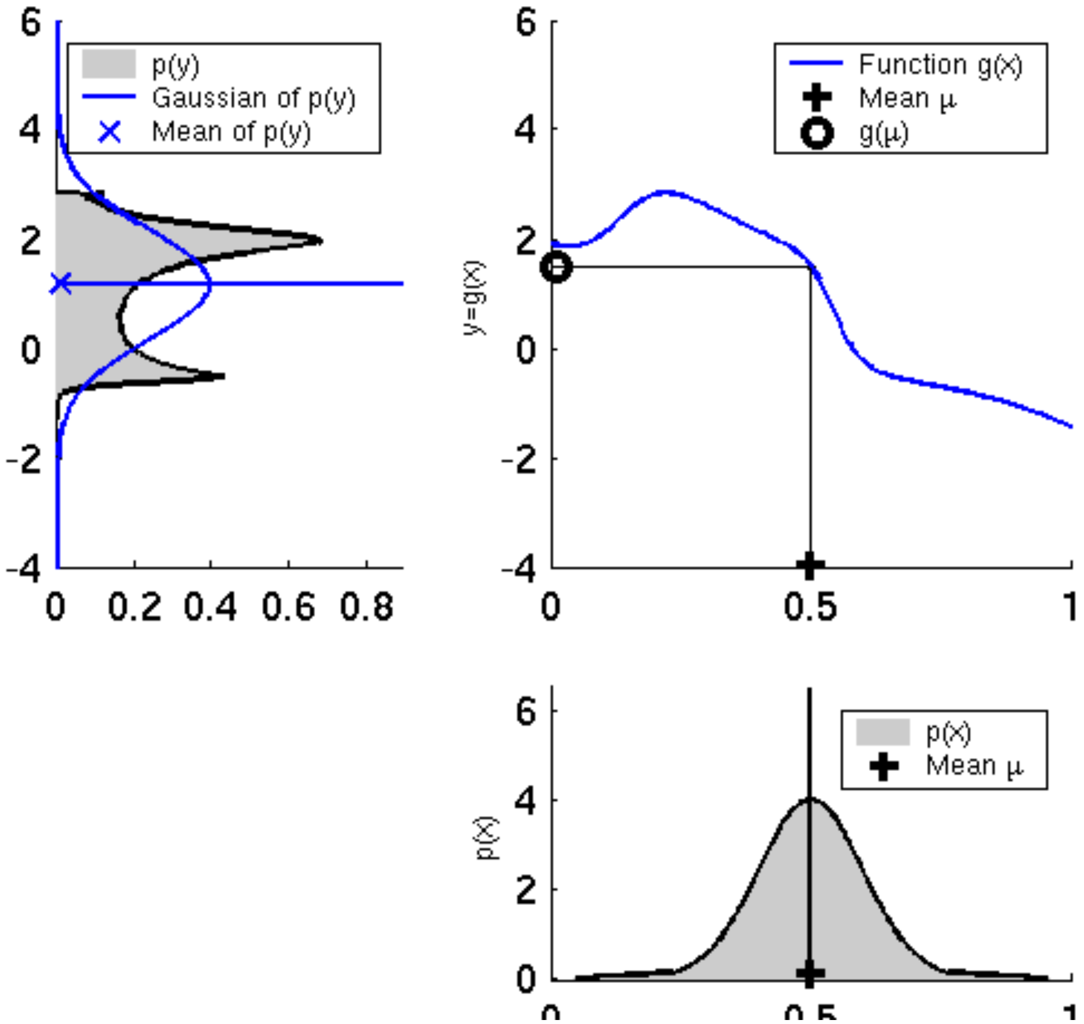
$$x_t = g(u_t, x_{t-1}) + \epsilon_t$$

- And the measurement model is now given by

$$z_t = h(x_t) + \delta_t$$

- Key idea: shift focus from computing exact posterior to efficiently compute a Gaussian approximation

Goal of EKF



p(y)
Gaussian of p(y)
Mean of p(y)
Function g(x)
Mean μ
g(μ)
y=g(x)
p(x)
Mean μ
p(x)

EKF: key idea

- Key idea: linearize *g* and *h* around the most likely state and transform beliefs according to such linear approximations
- For the dynamics equation

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \underbrace{J_g(u_t, \mu_{t-1})}_{:=G_t}(x_{t-1} - \mu_{t-1})$$

Jacobian of *g*

- Accordingly

$$p(x_t \mid u_t, x_{t-1}) = \det(2\pi R_t)^{-1/2}$$
$$\exp\left(-\frac{1}{2}[x_t - g(u_t, \mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1})]^T R_t^{-1}[x_t - g(u_t, \mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1})]\right)$$

EKF: key idea

- For the measurement model

$$h(x_t) \approx h(\overline{\mu}_t) + \underbrace{J_h(\overline{\mu}_t)}_{:=H_t}(x_t - \overline{\mu}_t)$$

- Accordingly,

$$p(z_t \,|\, x_t) = \det(2\pi Q_t)^{-1/2} \exp\left(-\frac{1}{2}[z_t - h(\overline{\mu}_t) - H_t(x_t - \overline{\mu}_t)] Q_t^{-1} [z_t - h(\overline{\mu}_t) - H_t(x_t - \overline{\mu}_t)]\right)$$

EKF: algorithm

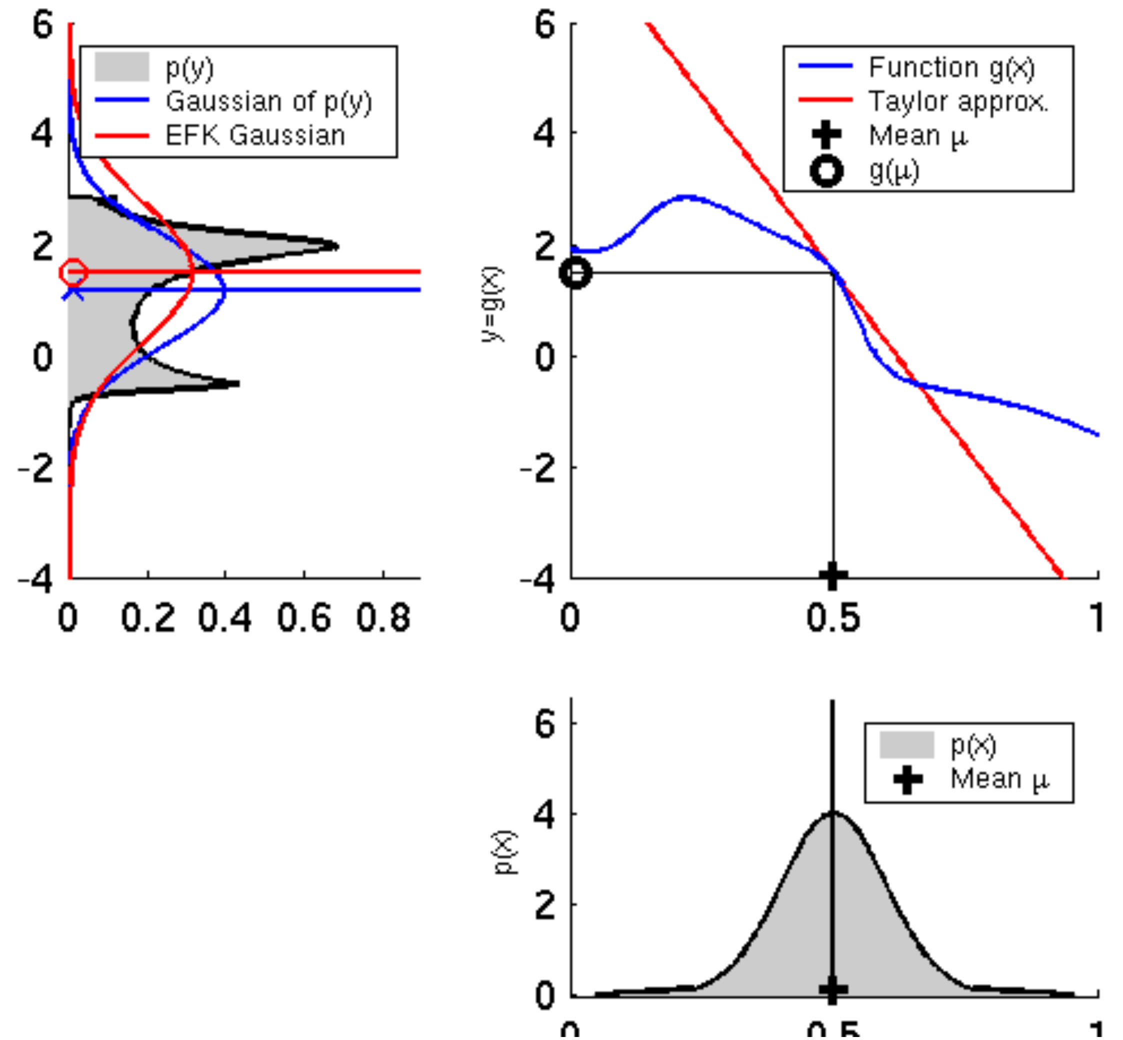
- Main differences:
  1. Linear predictions are replaced by their nonlinear generalizations
  2. EKF uses Jacobians instead of linear system matrices
  3. Mathematical derivation of EKF parallels that of KF

Data: $(\mu_{t-1}, \Sigma_{t-1}), u_t, z_t$
Result: (μ_t, Σ_t)

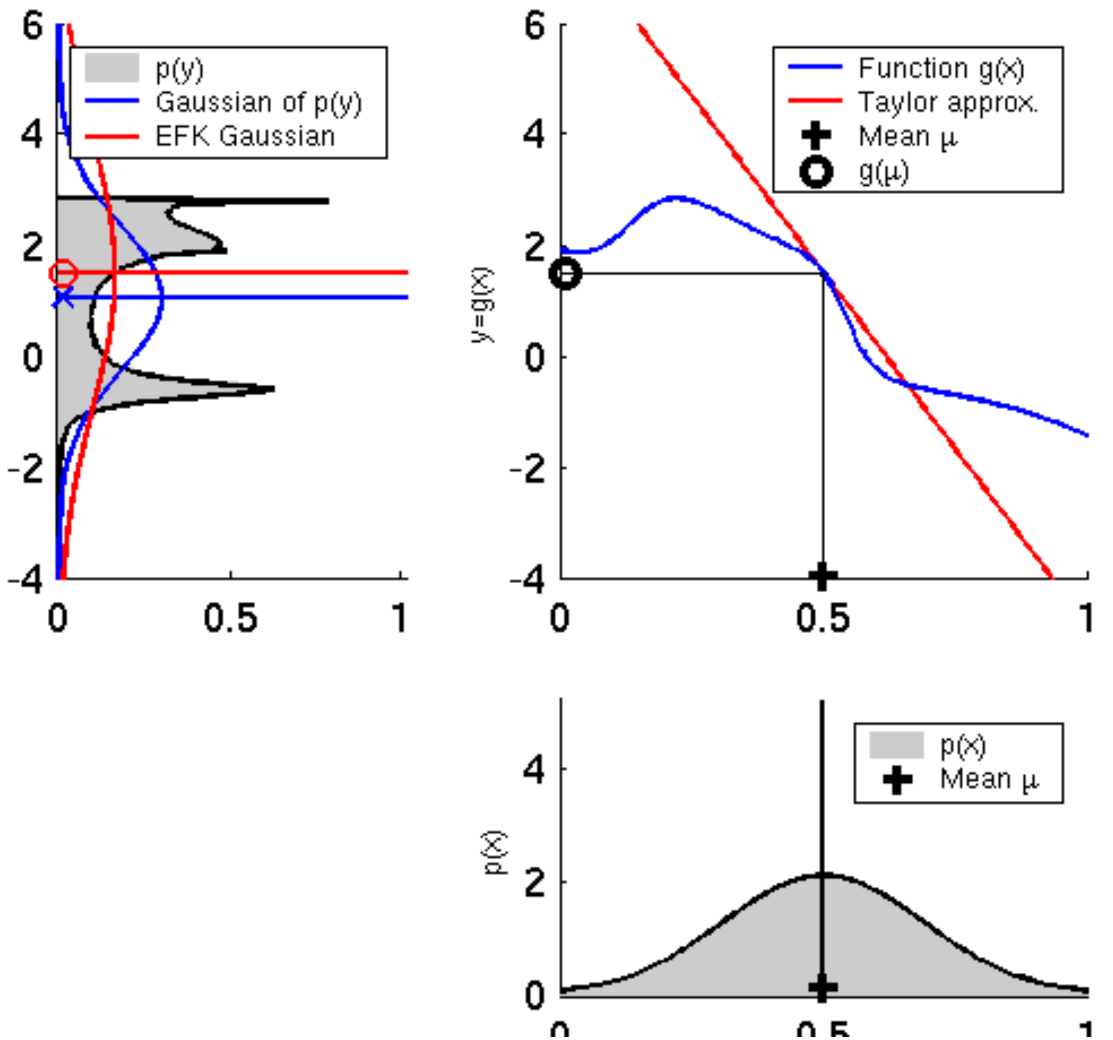
$$\overline{\mu}_t = g(u_t, \mu_{t-1}) \;;$$
$$\overline{\Sigma}_t = G_t \Sigma_{t-1} {G_t}^T + R_t;$$
$$K_t = \overline{\Sigma}_t H_t^T (H_t \overline{\Sigma}_t H_t^T + Q_t)^{-1};$$
$$\mu_t = \overline{\mu}_t + K_t(z_t - h(\overline{\mu}_t));$$
$$\Sigma_t = (I - K_t H_t)\overline{\Sigma}_t;$$

Return (μ_t, Σ_t)

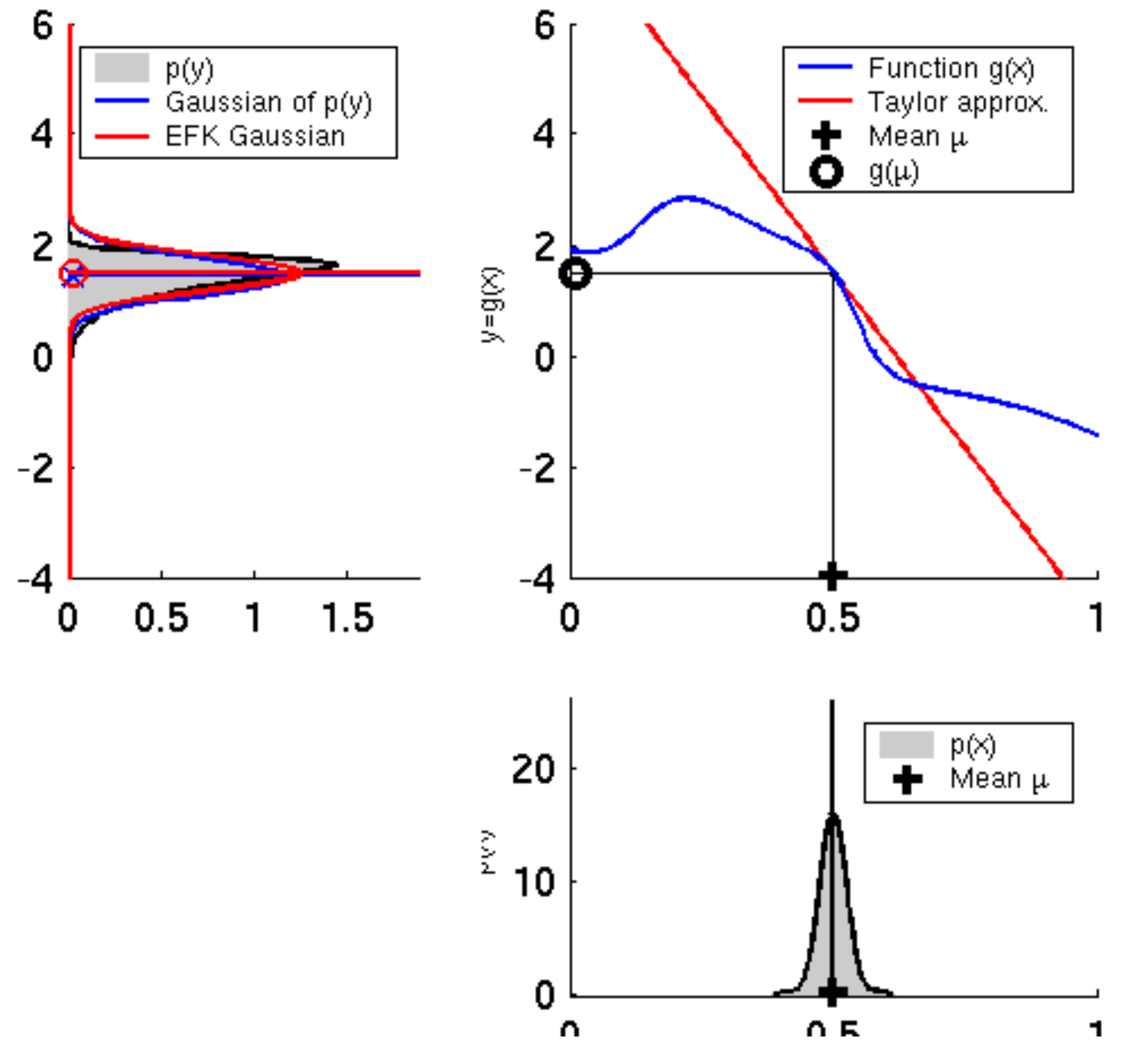
EKF: examples



EKF: examples



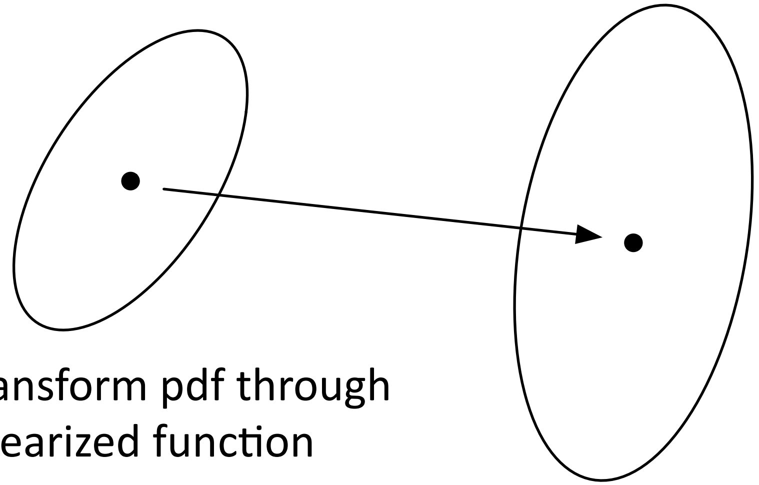
EKF: examples



Unscented Kalman filter (UKF) – basic idea

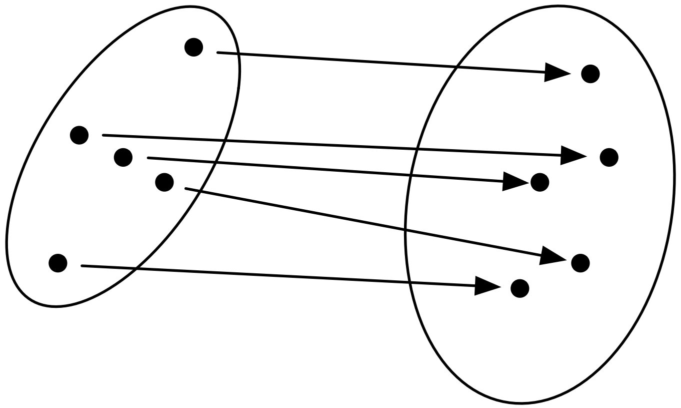
- Taylor series expansion applied by EKF is not the only way to approximate the transformation of a Gaussian; other approaches
  - Assumed density filter
  - Unscented Kalman filter (UKF)

EKF



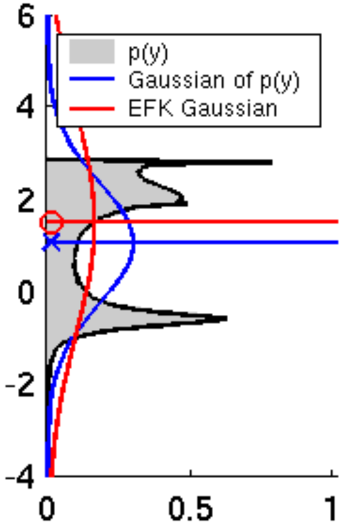
Transform pdf through linearized function

UKF

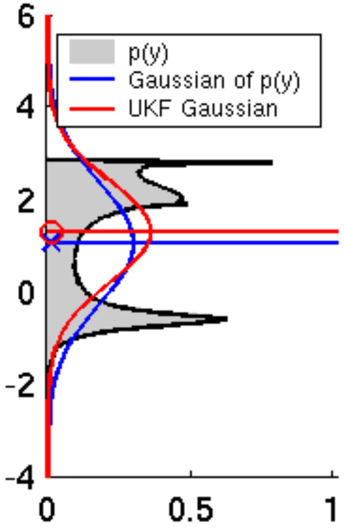


1. Compute sigma-points
2. Transform each sigma point through nonlinear function
3. Compute Gaussian from the transformed and weighted sigma-points

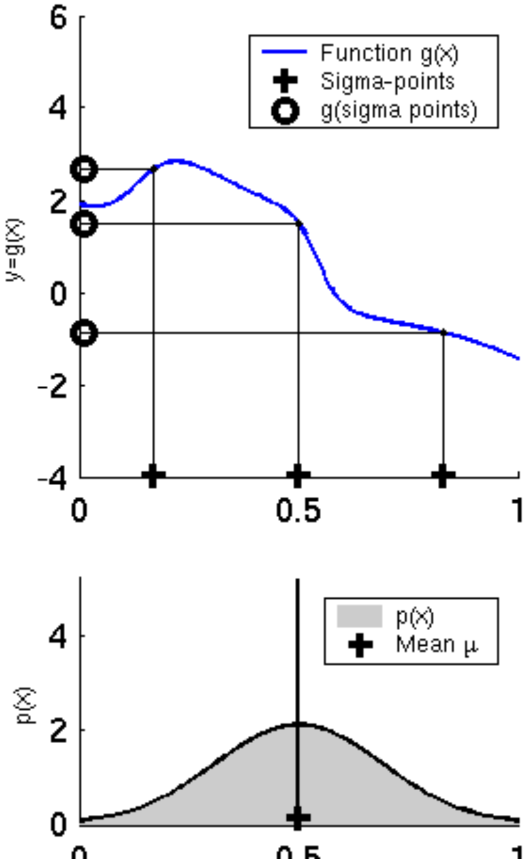
UKF: example



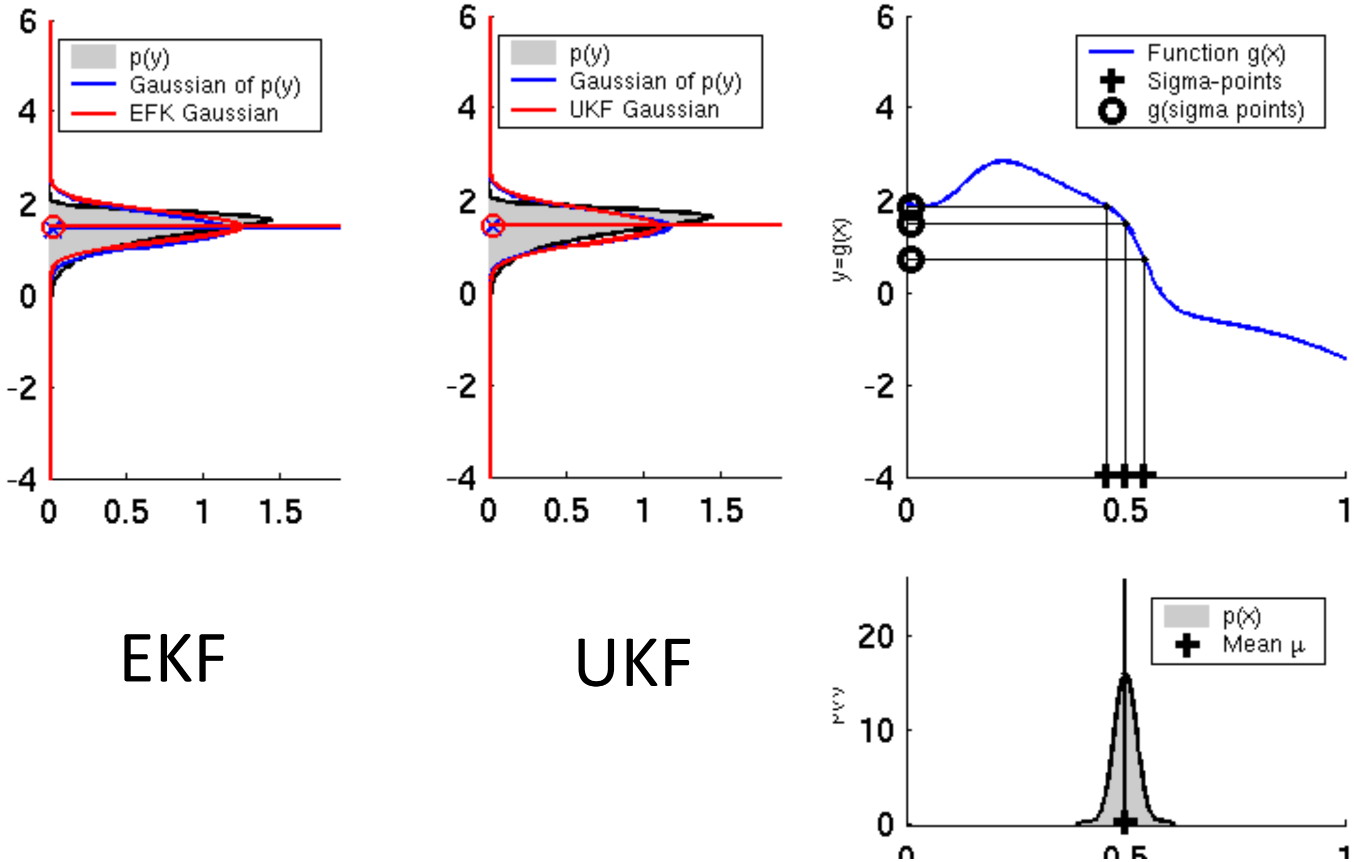
EKF



UKF



UKF: example


p(y)
Gaussian of p(y)
EFK Gaussian
p(y)
Gaussian of p(y)
UKF Gaussian
Function g(x)
Sigma-points
g(sigma points)
y=g(x)
p(x)
Mean μ
EKF
UKF

Next time