AA 274
Principles of Robotic Autonomy

Localization II: parametric and non-parametric filters

Stanford University

Autonomous Systems Lab

Today's lecture

- Aim
  - Learn about parametric and non-parametric filters
  - Introduction to mobile robot localization

- Readings
  - SNS: 5.1 – 5.6 (up to 5.6.3)
  - 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

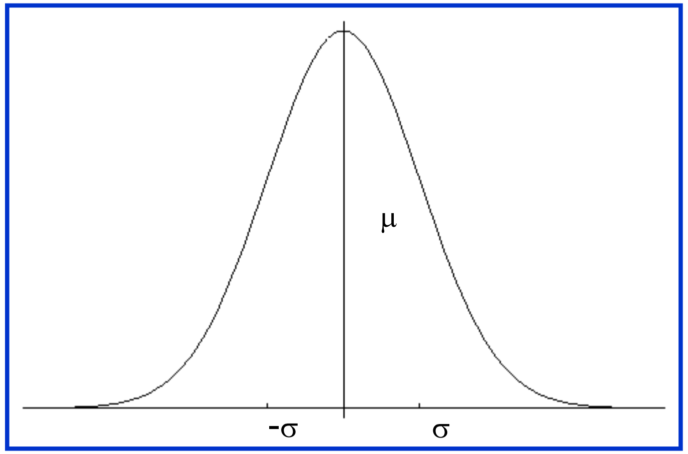
- Tractable implementations of Bayes' filter exploit structure and / or approximations; two main classes
  - Parametric filters: e.g., KF, EKF, UKF, etc.
  - Non parametric filters: e.g., histogram filter, particle filter, etc.

Gaussian distributions

- Key idea: belief represented as multivariate normal distribution

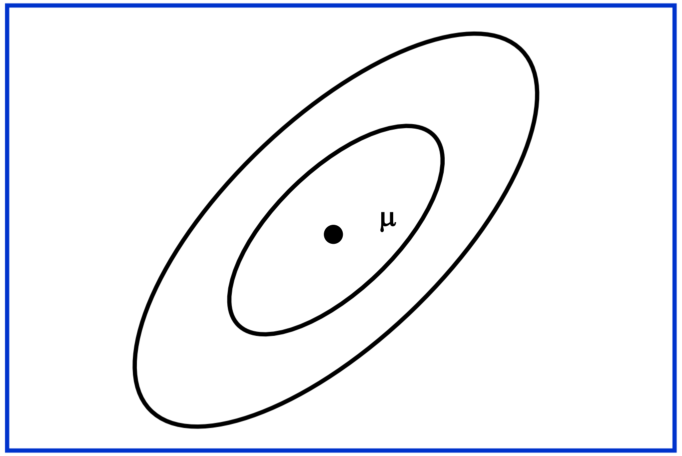
Univariate

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



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

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

Project covariance ahead

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

Correction

Compute Kalman gain

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

Update estimate with new measurement

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

Update covariance

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

$$\textbf{Data: } \overbrace{(\mu_{t-1}, \Sigma_{t-1})}^{bel(x_{t-1})}, u_t, z_t$$

$$\textbf{Result: } (\mu_t, \Sigma_t)$$

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

$$\left\{ \begin{aligned} \overline{\mu}_t &= A_t \mu_{t-1} + B_t u_t \; ; \\ \overline{\Sigma}_t &= A_t \Sigma_{t-1} A_t^T + R_t ; \end{aligned} \right.$$

Correction: $bel(x_t)$

$$\left\{ \begin{aligned} K_t &= \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1} ; \\ \mu_t &= \overline{\mu}_t + K_t (z_t - C_t \overline{\mu}_t) ; \\ \Sigma_t &= (I - K_t C_t) \overline{\Sigma}_t ; \end{aligned} \right.$$

$$\text{Return } \underbrace{(\mu_t, \Sigma_t)}_{bel(x_t)}$$

Kalman filter: derivation (sketch)

- Prediction

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

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

- 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

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

$$\mathcal{N}(Cx_t, Q_t) \qquad \qquad \mathcal{N}(\overline{\mu}_t,\, \overline{\Sigma}_t)$$

- After some algebraic manipulations

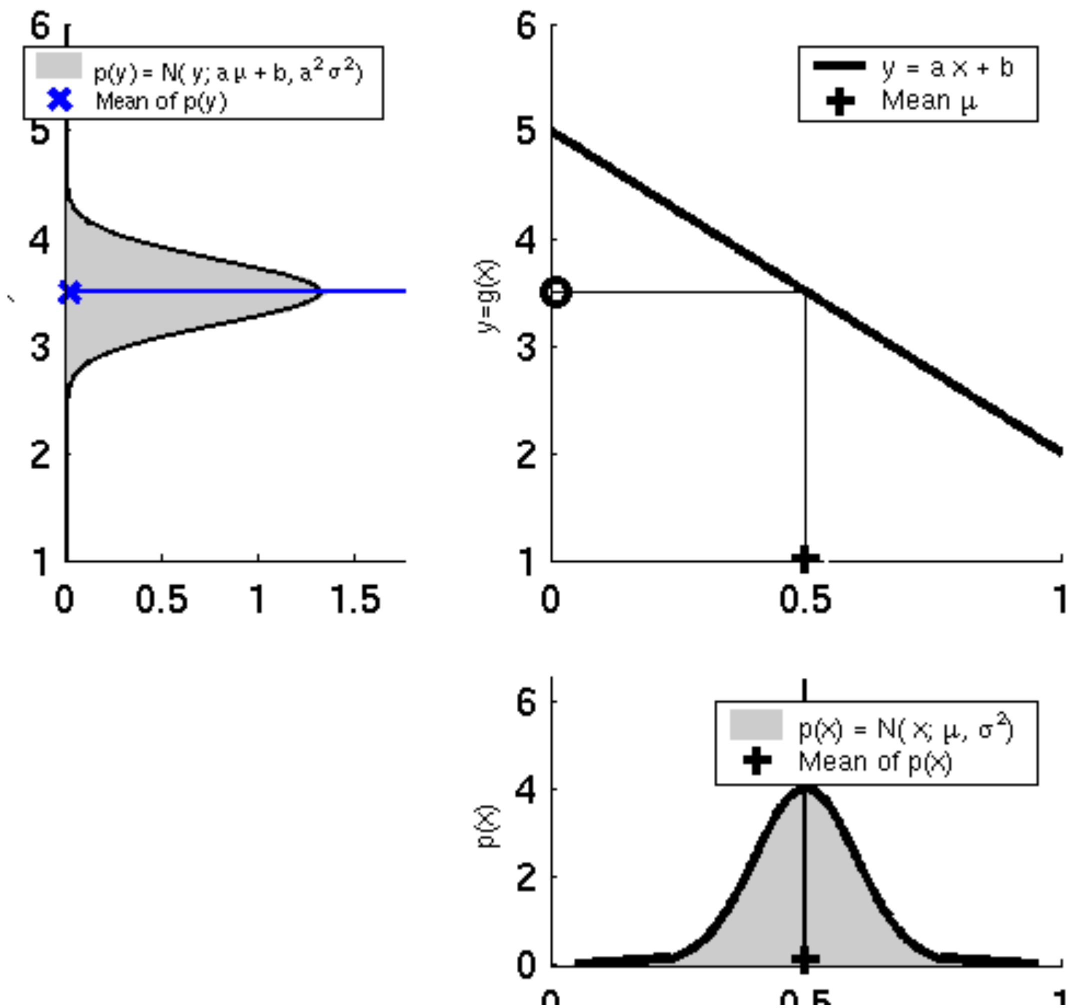
$$bel(x_t) = \mathcal{N}(\mu_t,\, \Sigma_t)$$

with

$$\begin{aligned} K_t &= \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1} \\ \mu_t &= \overline{\mu}_t + K_t(z_t - C_t \overline{\mu}_t) \\ \Sigma_t &= (I - K_t C_t)\overline{\Sigma}_t \end{aligned}$$

- 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

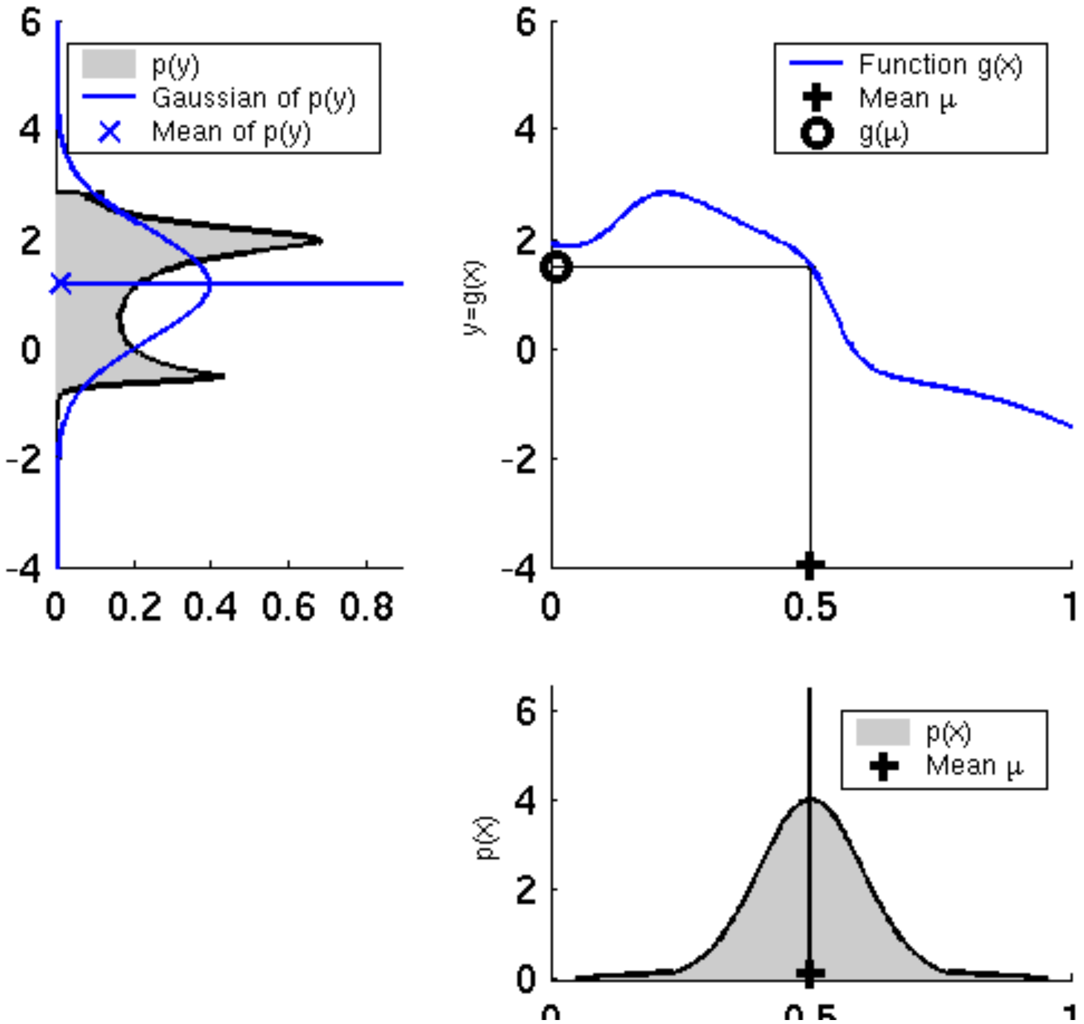
$$x_t = g(u_t, x_{t-1}) + \epsilon_t$$

- 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

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

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,

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

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

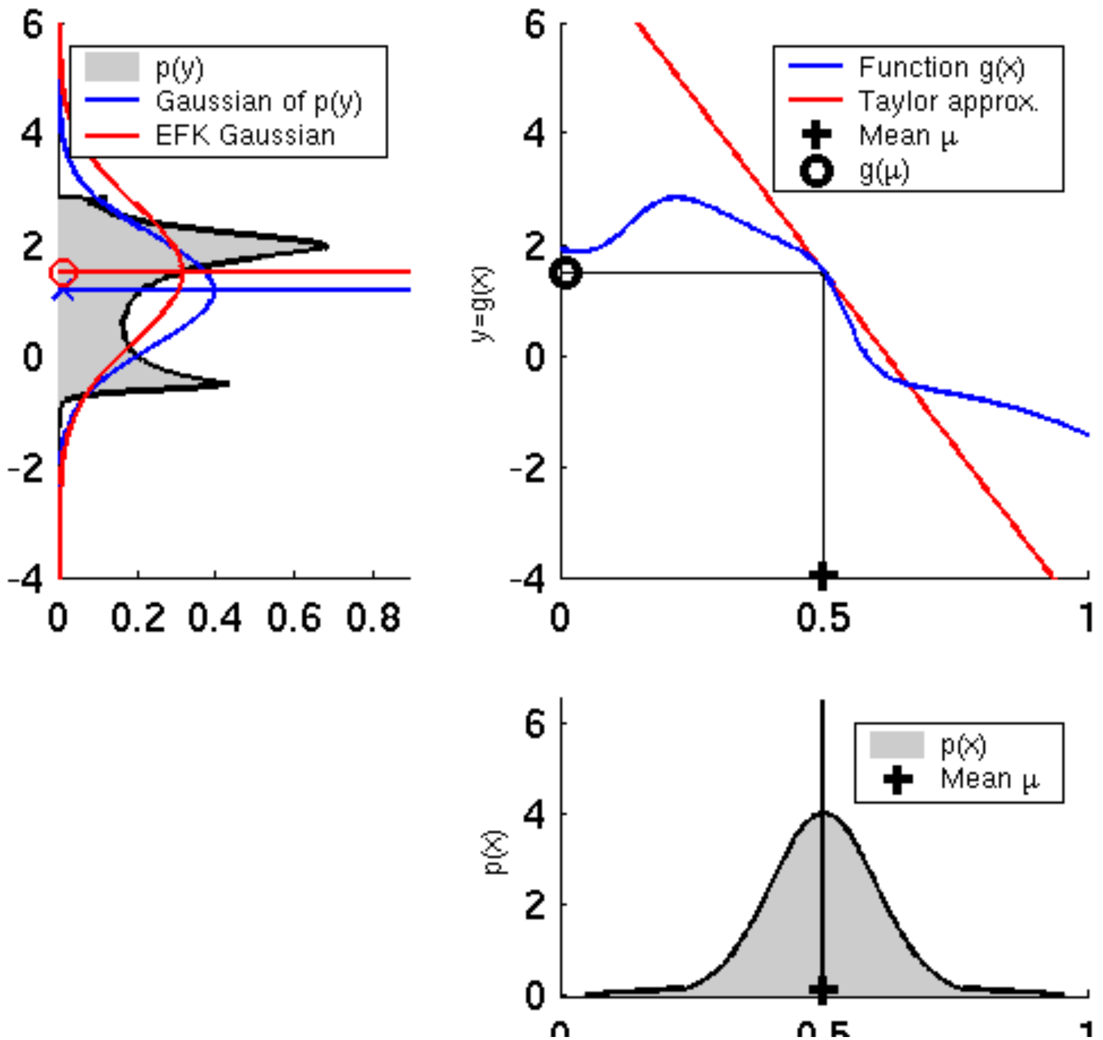
Data: $(\mu_{t-1}, \Sigma_{t-1}), u_t, z_t$
Result: (μ_t, Σ_t)

$$\overline{\mu}_t = g(u_t, \mu_{t-1}) \ ;$$
$$\overline{\Sigma}_t = G_t \Sigma_{t-1} {G_t}^T + R_t;$$

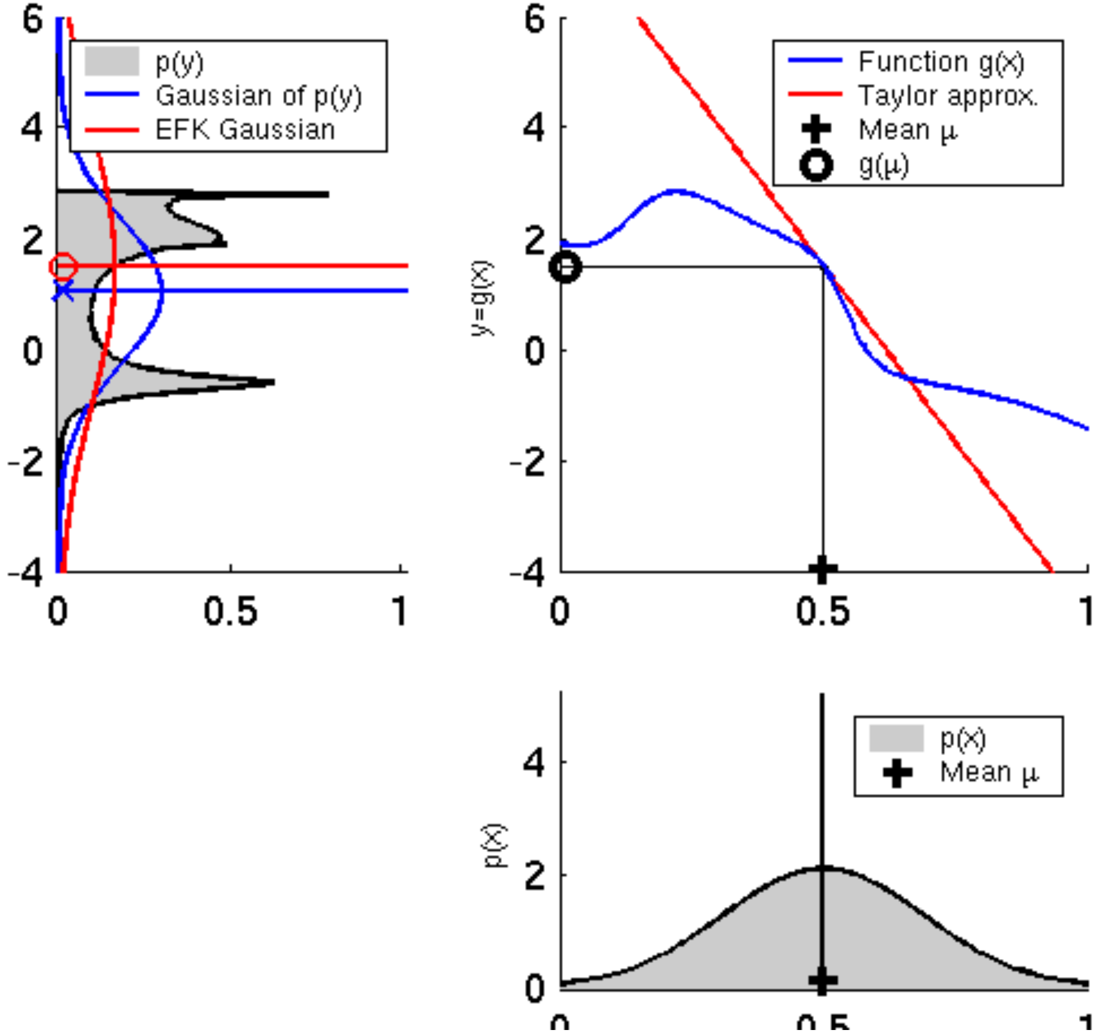
$$K_t = \overline{\Sigma}_t H_t^T (H_t \overline{\Sigma}_t H_t^T + Q_t)^{-1};$$
$$\mu_t = \overline{\mu}_t + K_t(z_t - h(\overline{\mu}_t));$$
$$\Sigma_t = (I - K_t H_t)\overline{\Sigma}_t;$$

Return (μ_t, Σ_t)

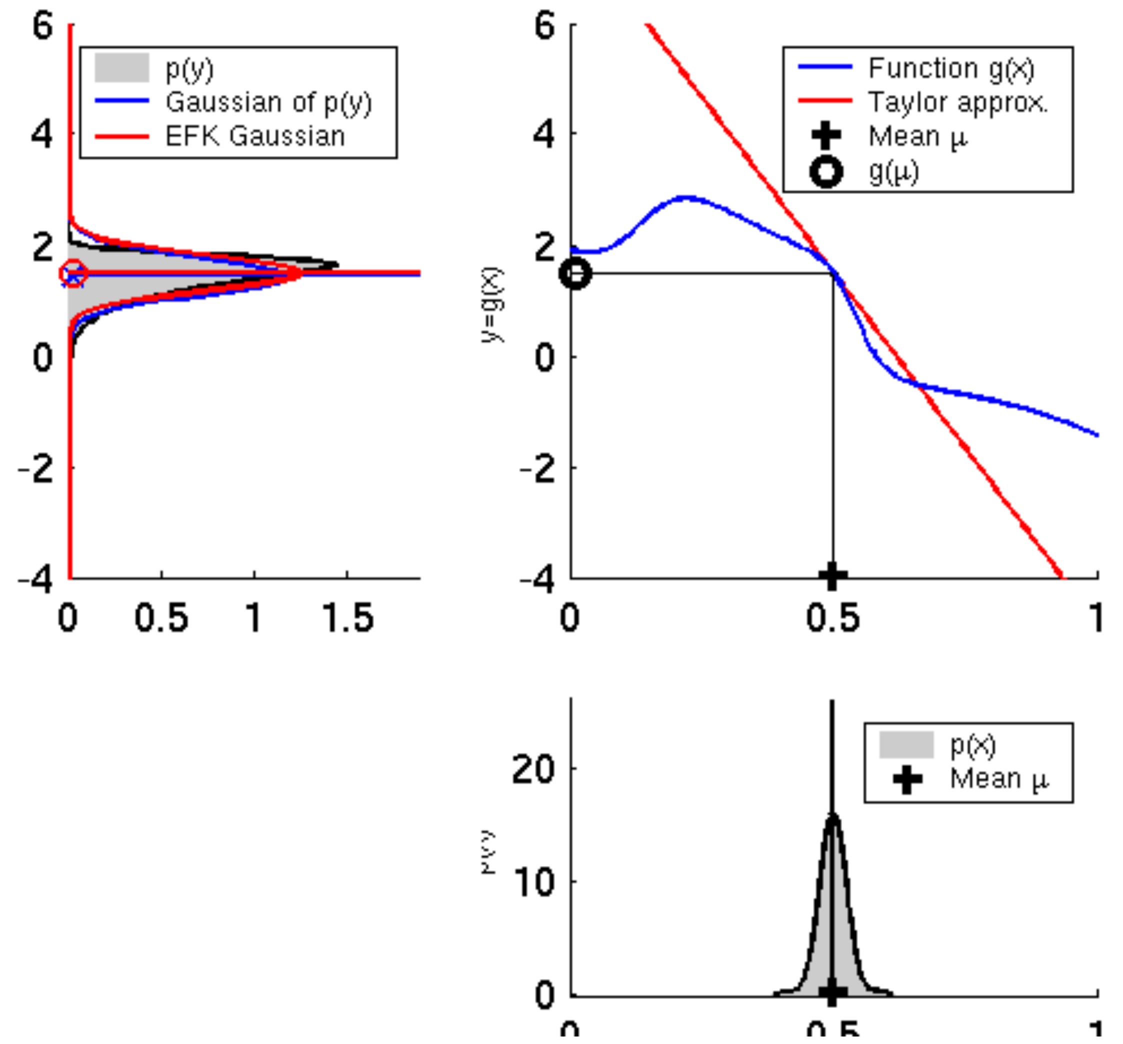
EKF: examples



EKF: examples



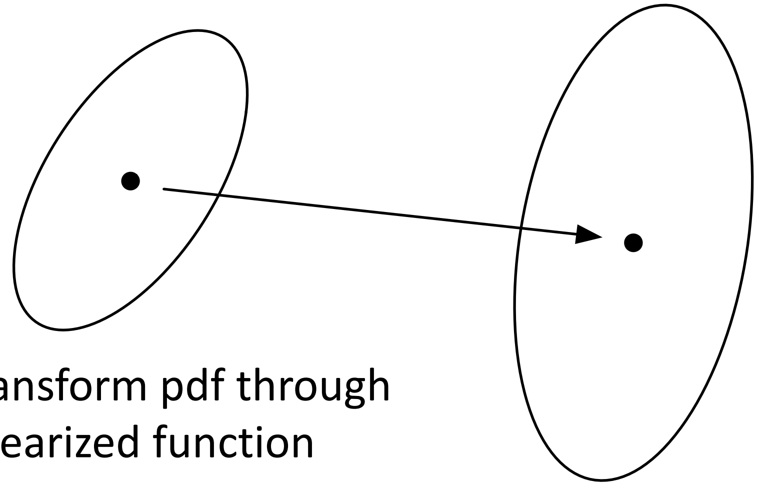
EKF: examples



Unscented Kalman filter (UKF) – basic idea

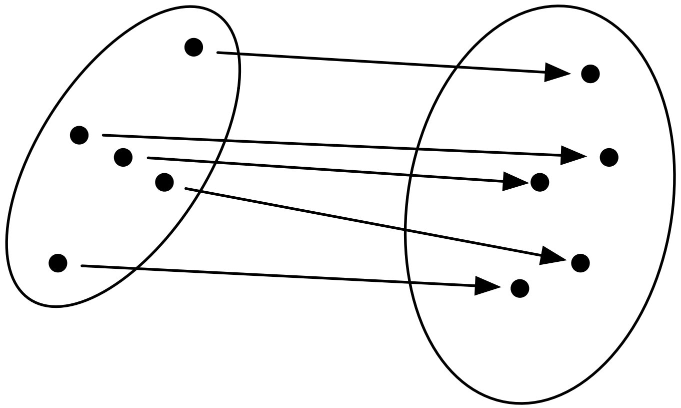
- Taylor series expansion applied by EKF is not the only way to approximate the transformation of a Gaussian; other approaches
    - Assumed density filter
    - Unscented Kalman filter (UKF)

EKF



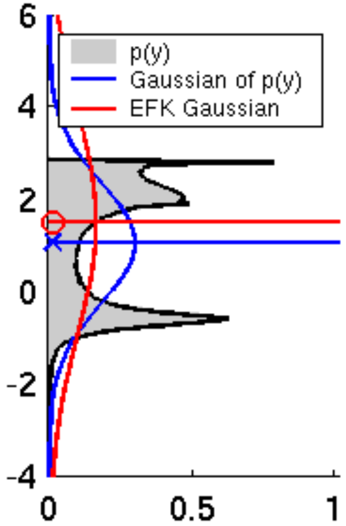
Transform pdf through linearized function

UKF

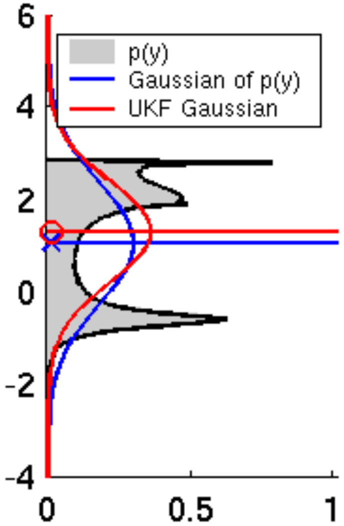


1. Compute sigma-points
2. Transform each sigma point through nonlinear function
3. Compute Gaussian from the transformed and weighted sigma-points

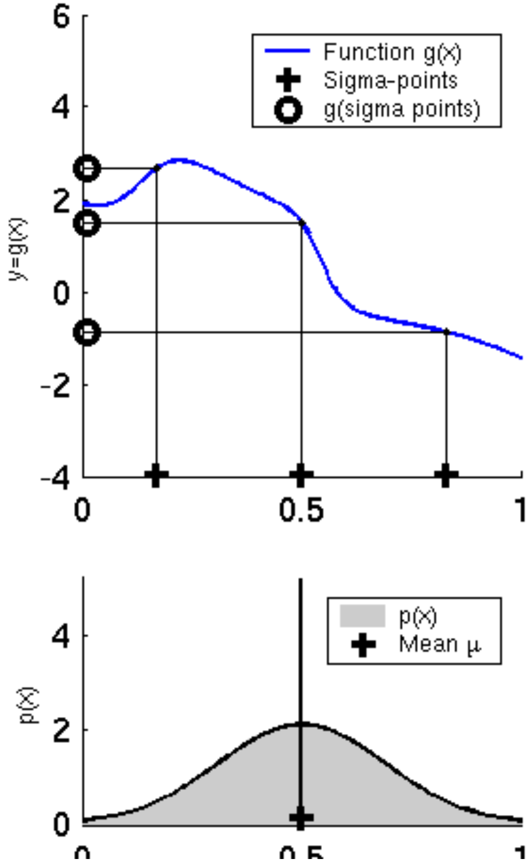
UKF: example



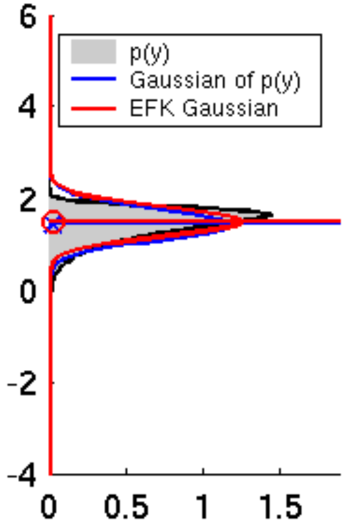
EKF



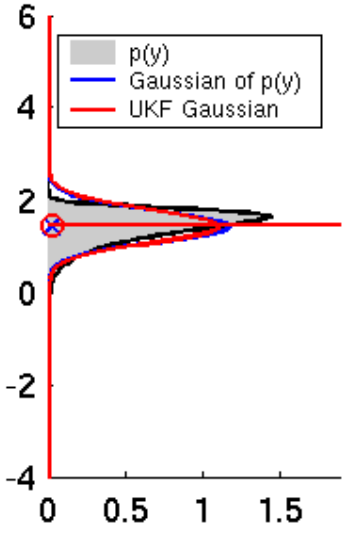
UKF



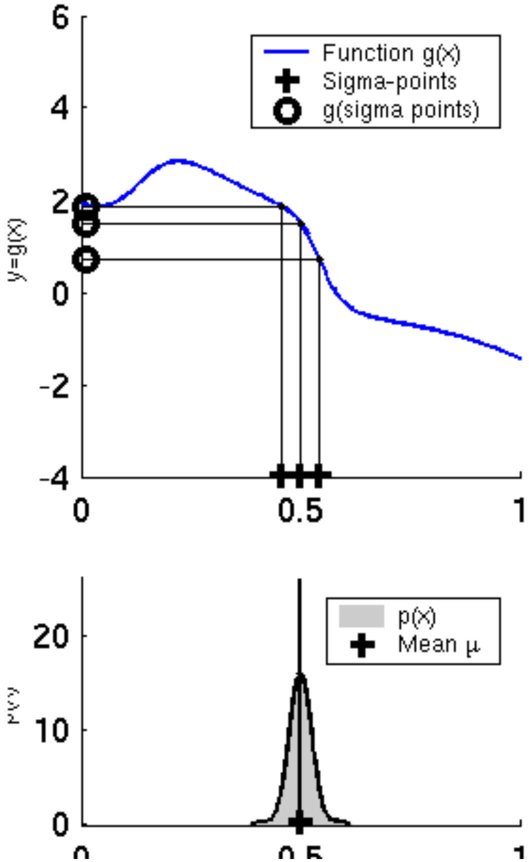
UKF: example



EKF

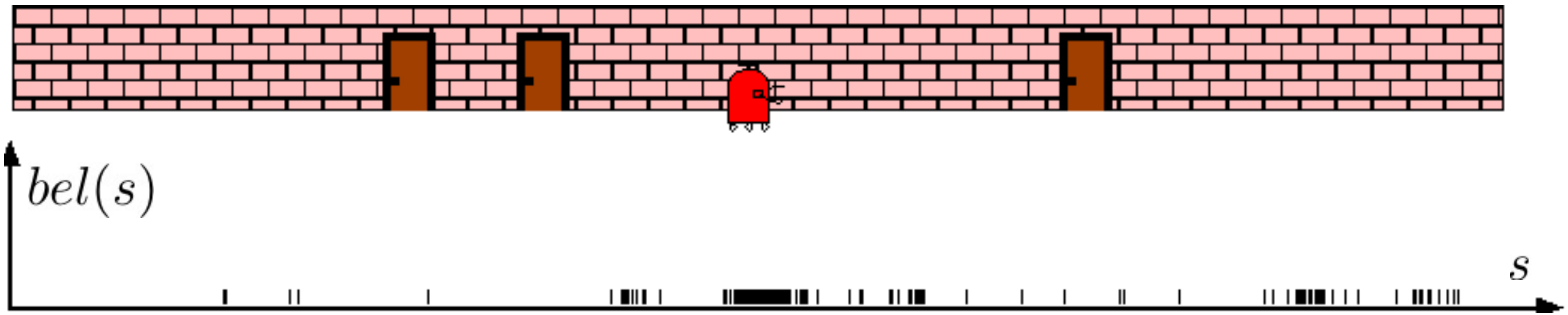


UKF



Particle filters

- Key idea: represent posterior $bel(x_t)$ by a set of random samples



- Allows one to represent non-Gaussian distributions and handle nonlinear transformations in a direct way

Particle filters

- Samples of posterior distribution are called *particles*, denoted as

$$\mathcal{X}_t := x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[M]}$$

- A particle represents a hypothesis about what the true world state might be at time t
- Ideally, particles should be distributed according to

$$x_t^{[m]} \sim p(x_t \mid z_{1:t},\, u_{1:t}) = bel(x_t)$$

- Matching exact only as $M \rightarrow \infty$, but $M \approx 1000$ usually good enough
- A particle filter constructs the particle set $\mathcal{X}_t$ from the particle set $\mathcal{X}_{t-1}$ recursively, with the goal of matching the distribution $bel(x_t)$

Particle filter: algorithm

- The temporary particle set $\overline{\mathcal{X}}_t$ represents the belief $\overline{bel}(x_t)$
- The particle set $\mathcal{X}_t$ represents the belief $bel(x_t)$
- Importance factors are used to incorporate measurement z_t in the particle set
- After resampling, particles are (as $M \to \infty$) distributed as

$$bel(x_t) = \eta\, p(z_t \mid x_t^{[m]})\, \overline{bel}(x_t)$$

Data: $\underbrace{\mathcal{X}_{t-1}}_{bel(x_{t-1})}, u_t, z_t$
Result: $\mathcal{X}_t$
$\overline{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$;
for $i = 1$ **to** M **do**

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

  Sample $x_t^{[m]} \sim p(x_t \mid u_t, x_{t-1}^{[m]})$;

Importance factor

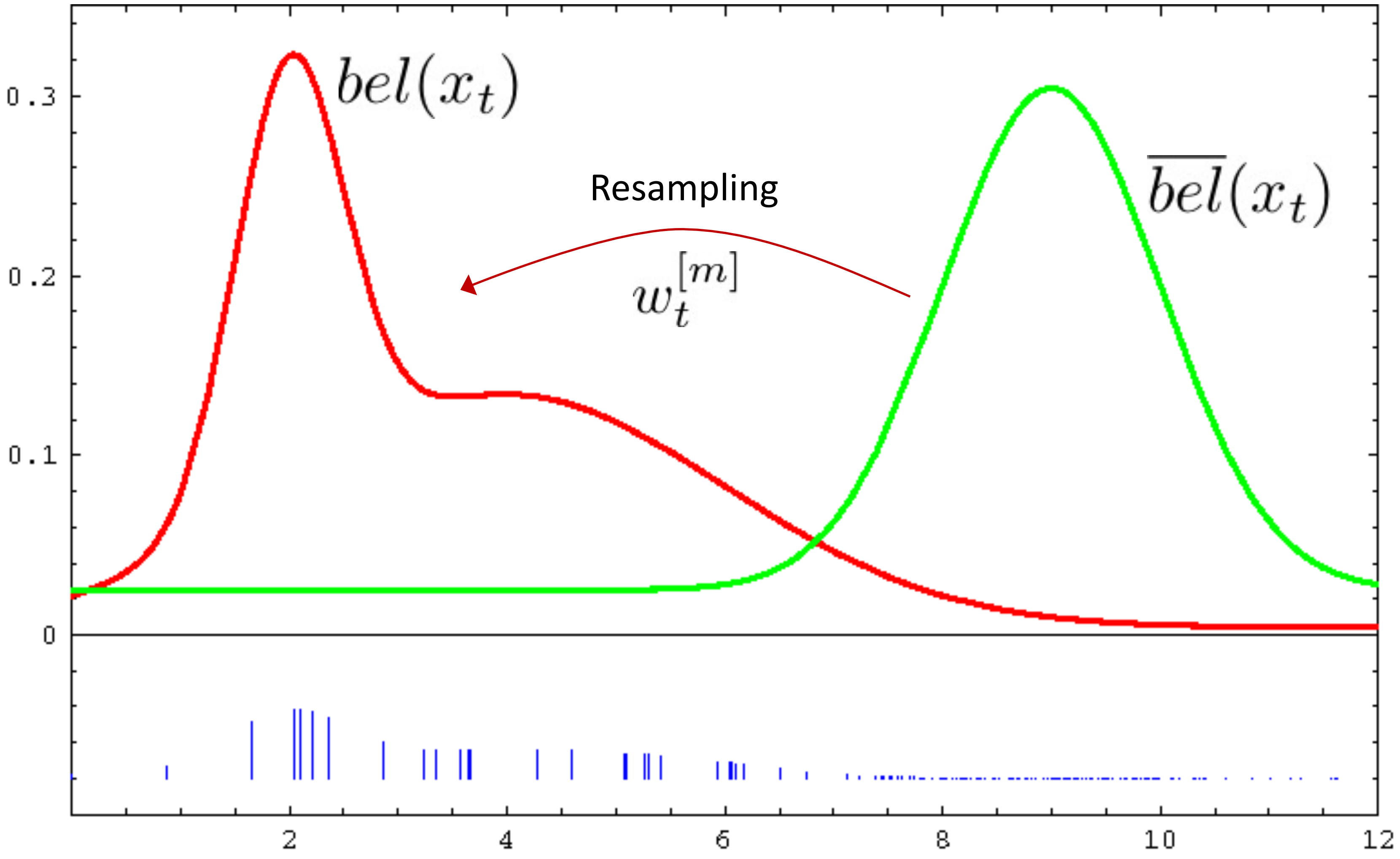
  $w_t^{[m]} = p(z_t \mid x_t^{[m]})$;
  $\overline{\mathcal{X}}_t = \overline{\mathcal{X}}_t \cup \left(x_t^{[m]}, w_t^{[m]}\right)$;

end

Correction: $bel(x_t)$

for $m = 1$ **to** M **do**
  Draw i with probability $\propto w_t^{[i]}$;
  Add $x_t^{[i]}$ to $\mathcal{X}_t$;
end
Return $\underbrace{\mathcal{X}_t}_{bel(x_t)}$

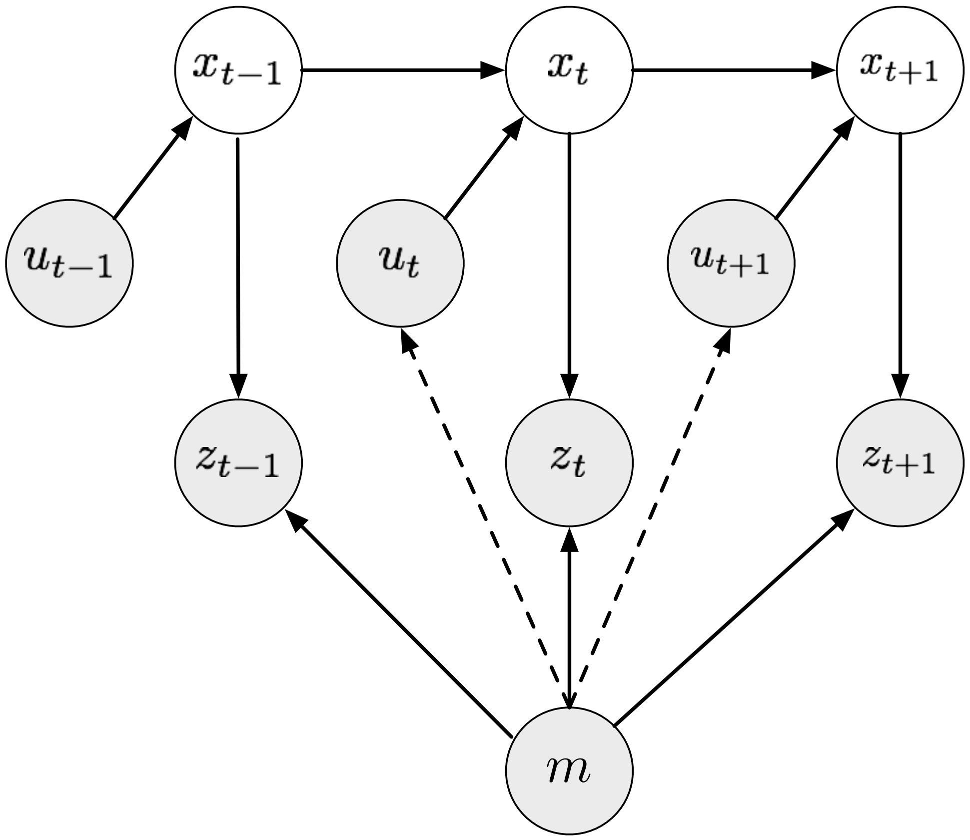
More on resampling



Mobile robot localization

- Problem: determine pose of a robot relative to a *given* map

- Localization can be interpreted as the problem of establishing correspondence between the map coordinate system and the robot's local coordinate frame
- This process requires integration of data over time



Local versus global localization

- Position tracking assumes that the initial pose is known -> *local* problem well-addressed via Gaussian filters
- In global localization, the initial pose is unknown -> *global* problem best addressed via non-parametric, multi-hypothesis filters
- In kidnapped robot localization, initial pose is unknown and during operation robot can be "kidnapped" and "teleported" to some other location -> *global* problem best addressed via non-parametric, multi-hypothesis filters

Static versus dynamic environments

- Static environments are environments were the only variable quantity is the pose of the robot
- Dynamic environments possess objects (e.g., people) other than the robot whose locations change over time -> addressed via either state augmentation or outlier rejection

Passive versus active localization

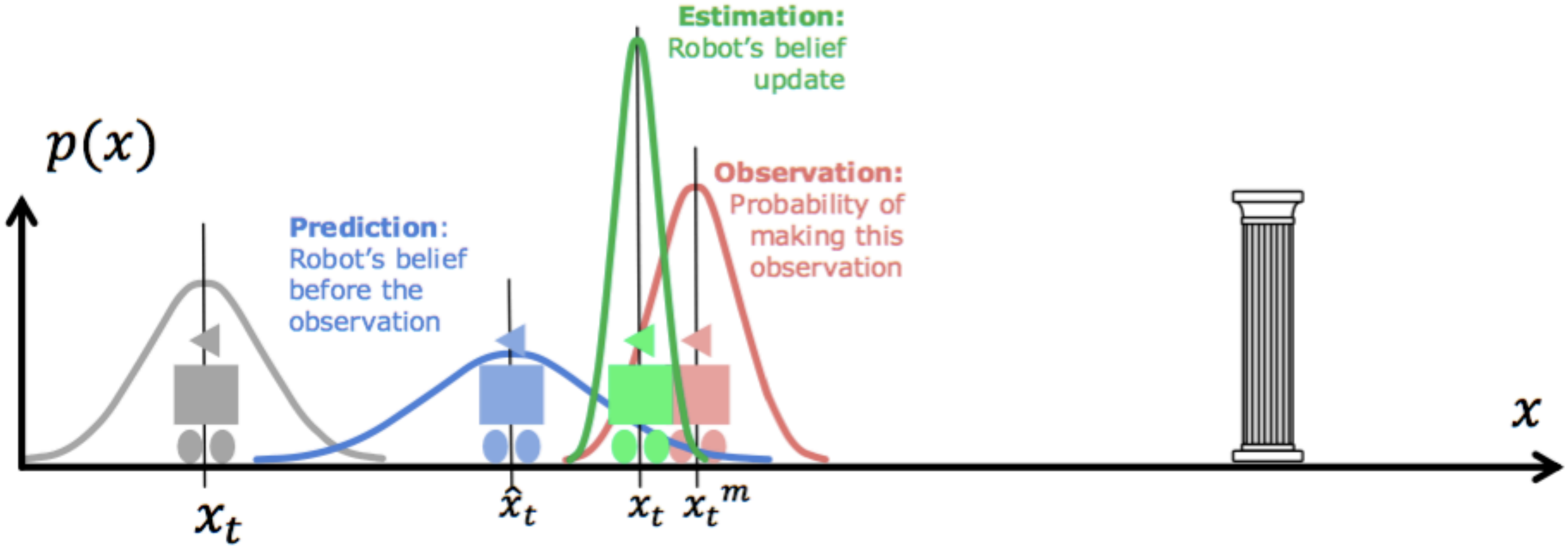
- In passive localization, localization module only *observes* the robot; i.e., robot's motion is not aimed at facilitating localization
- In active localization, robot's actions are aimed at minimizing the localization error
- Hybrid approaches are possible

Single-robot versus multi-robot

- In single-robot localization, a single, individual robot is involved in the localization process
- In multi-robot localization, a team of robots is engaged with localization, possibly cooperatively (or even adversarially!)

In this class we will focus on local & global, static (or quasi-static), passive, single-robot localization problems

Next time



$p(x)$
Prediction:
Robot's belief
before the
observation
Estimation:
Robot's belief
update
Observation:
Probability of
making this
observation
x
x_t
$\hat{x}_t$
x_t
$x_t{}^m$