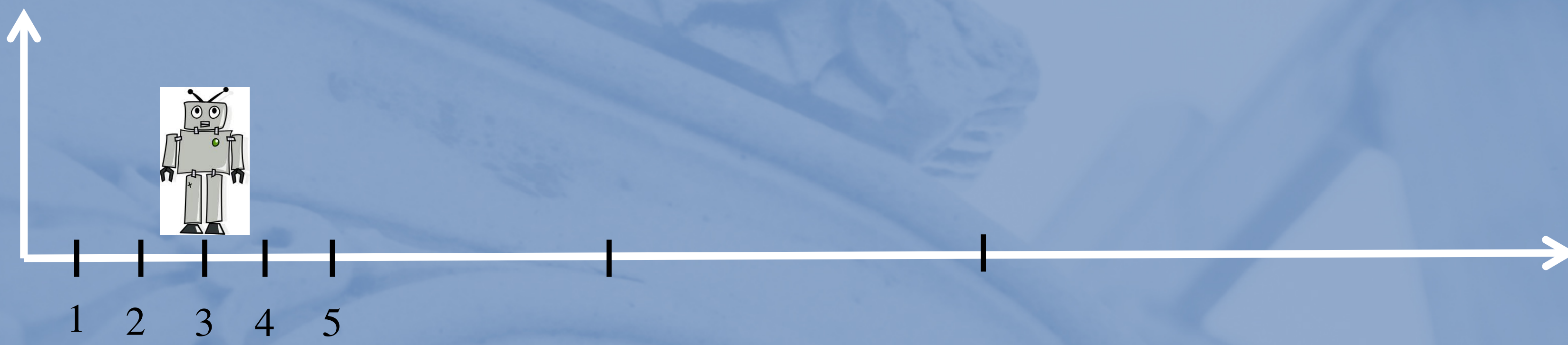
CSE 468/568: Robotics Algorithms

Extended Kalman Filter and Particle Filters

Karthik Dantu
kdantu@buffalo.edu

Some slides adopted from USC, Thrun book, ETH, and others

1-D Kalman Filter



$$x_t = x_{t-1} + u_t + \varepsilon_t$$
$$z_t = x_t + \delta_t$$

$$\varepsilon_t = N(\mu_1, \sigma_1^2)$$
$$\delta_t = N(\mu_2, \sigma_2^2)$$

- State transition linear in motion variables
- Measurement linear in state
- Motion and measurement noise are Gaussian

Example: Simple 1D Linear System

Given: u=0

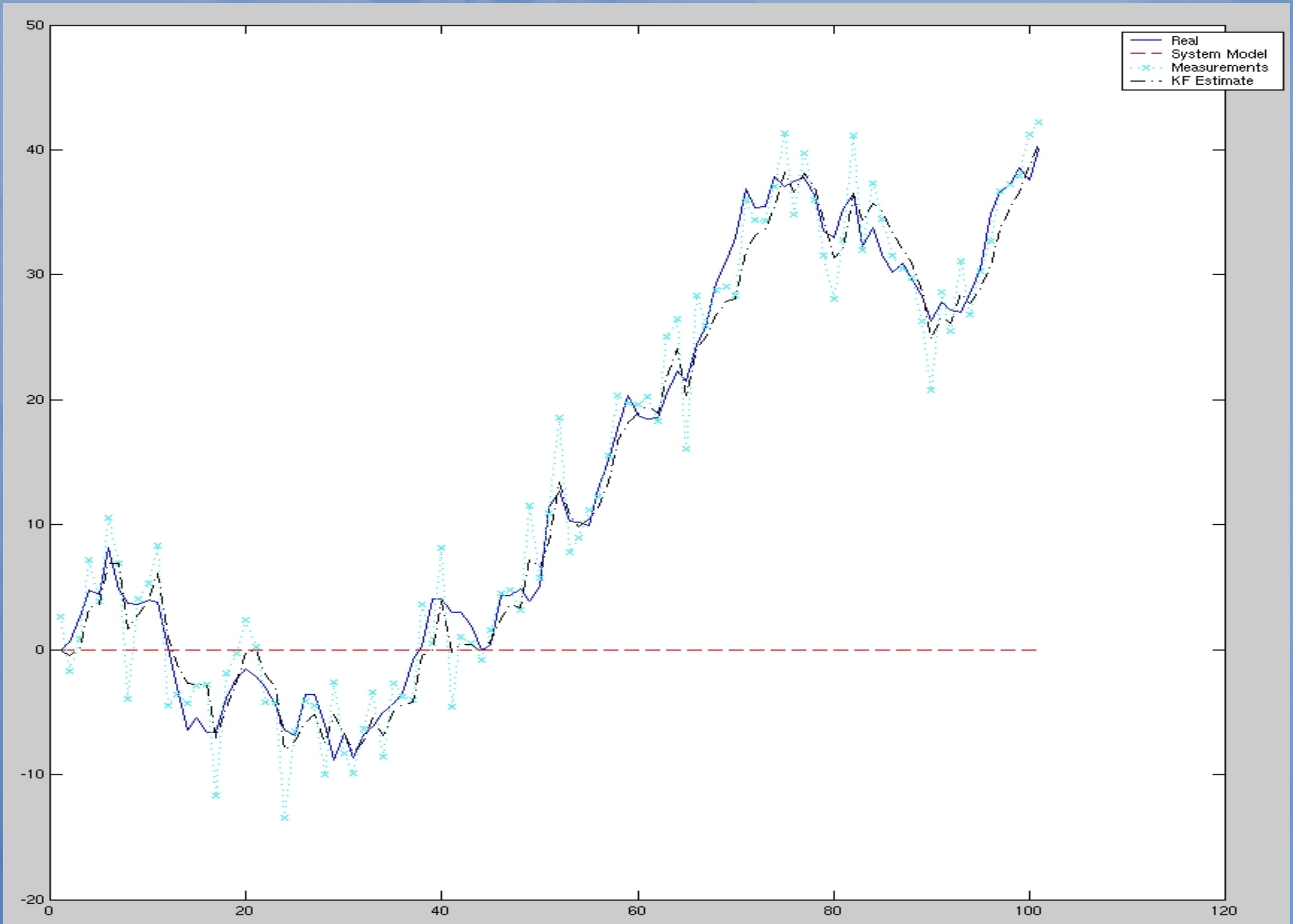
Initial state estimate = 0

Linear System:

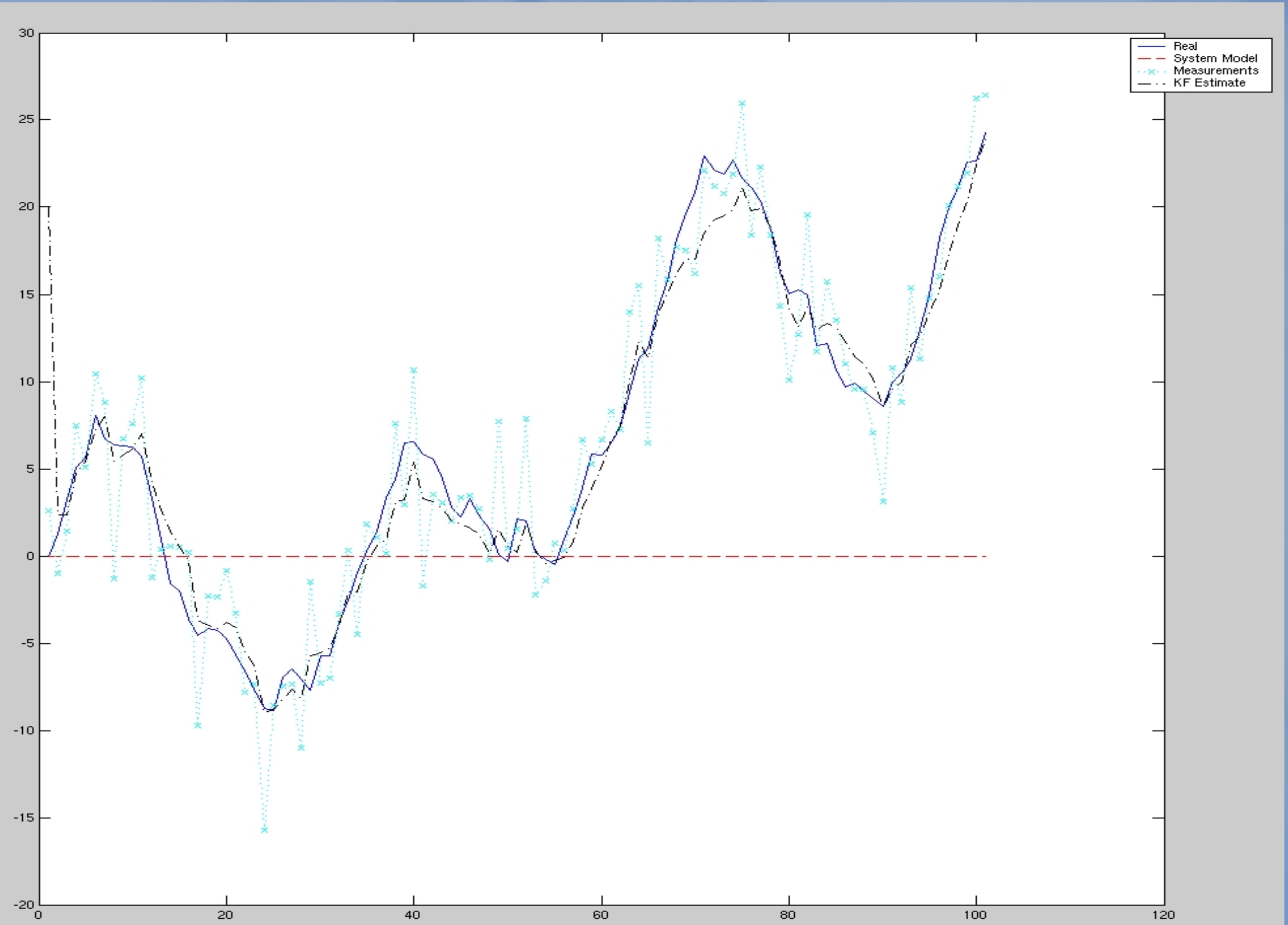
$$x_{t+1} = x_t + w_t$$

$$z_{t+1} = x_{t+1} + n_{t+1}$$

State Estimate



State Estimate With Bad Initialization



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

Linearity Assumption

- Most systems of interest not linear
- To model such systems, a linear process model needs to be generated out of the non-linear dynamics
- Extended Kalman Filter is a method by which state propagation equations and sensor models can be linearized about the current estimate
- Linearization increases state error residual since it is not the best estimate

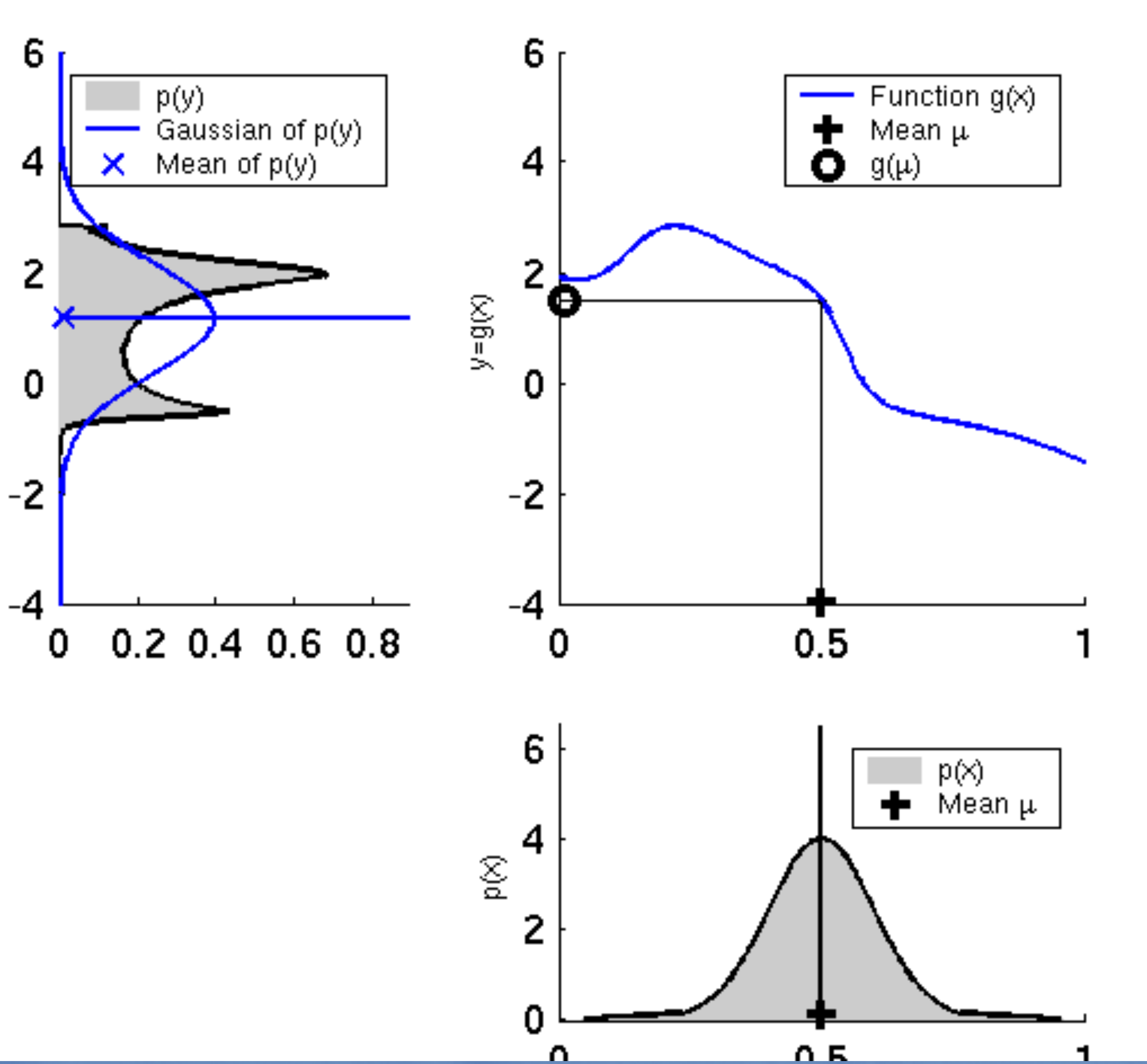
Nonlinear Dynamic Systems

- Realistic robots involve non-linear functions

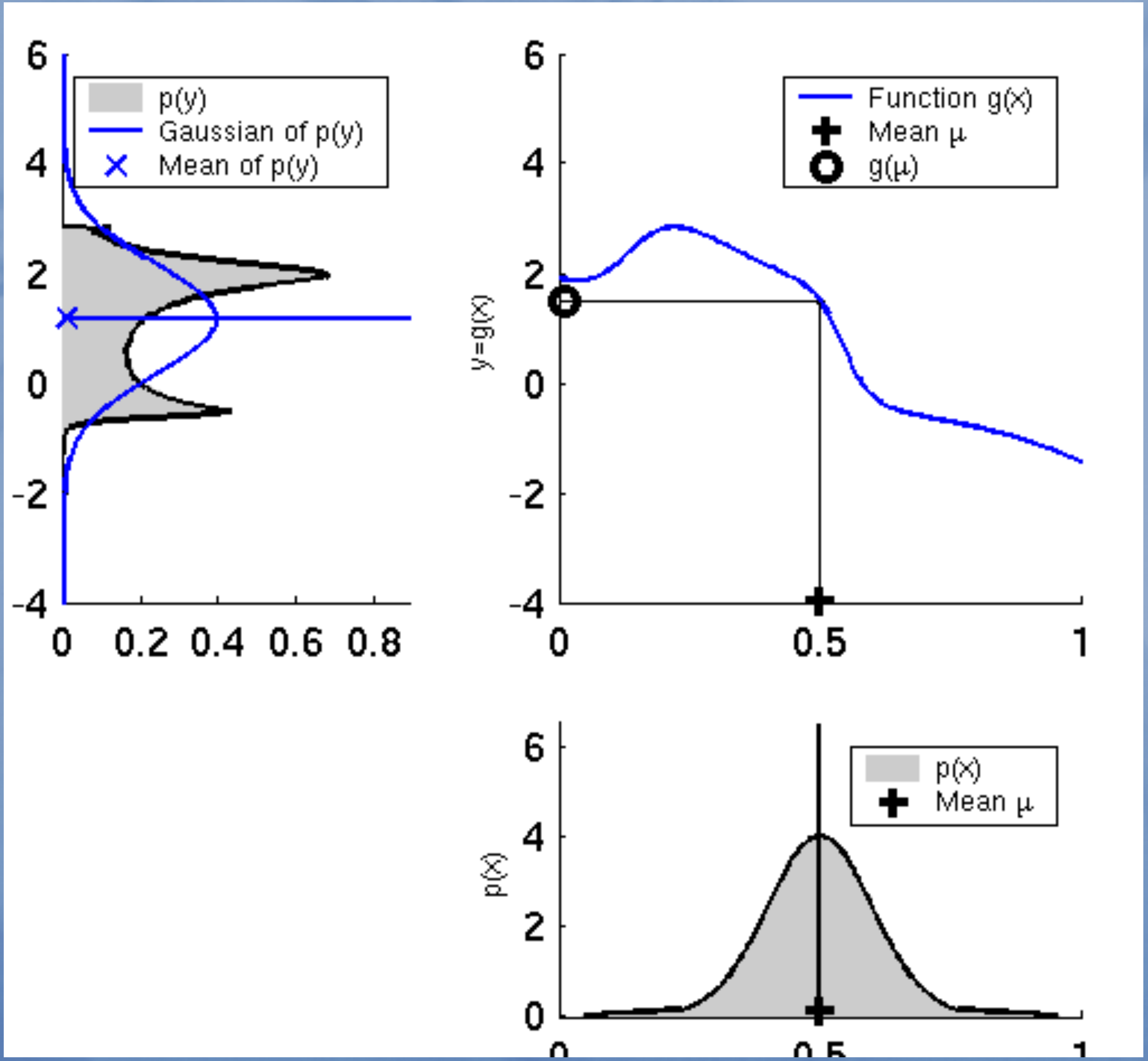
$$x_t = g(u_t, x_{t-1})$$

$$z_t = h(x_t)$$

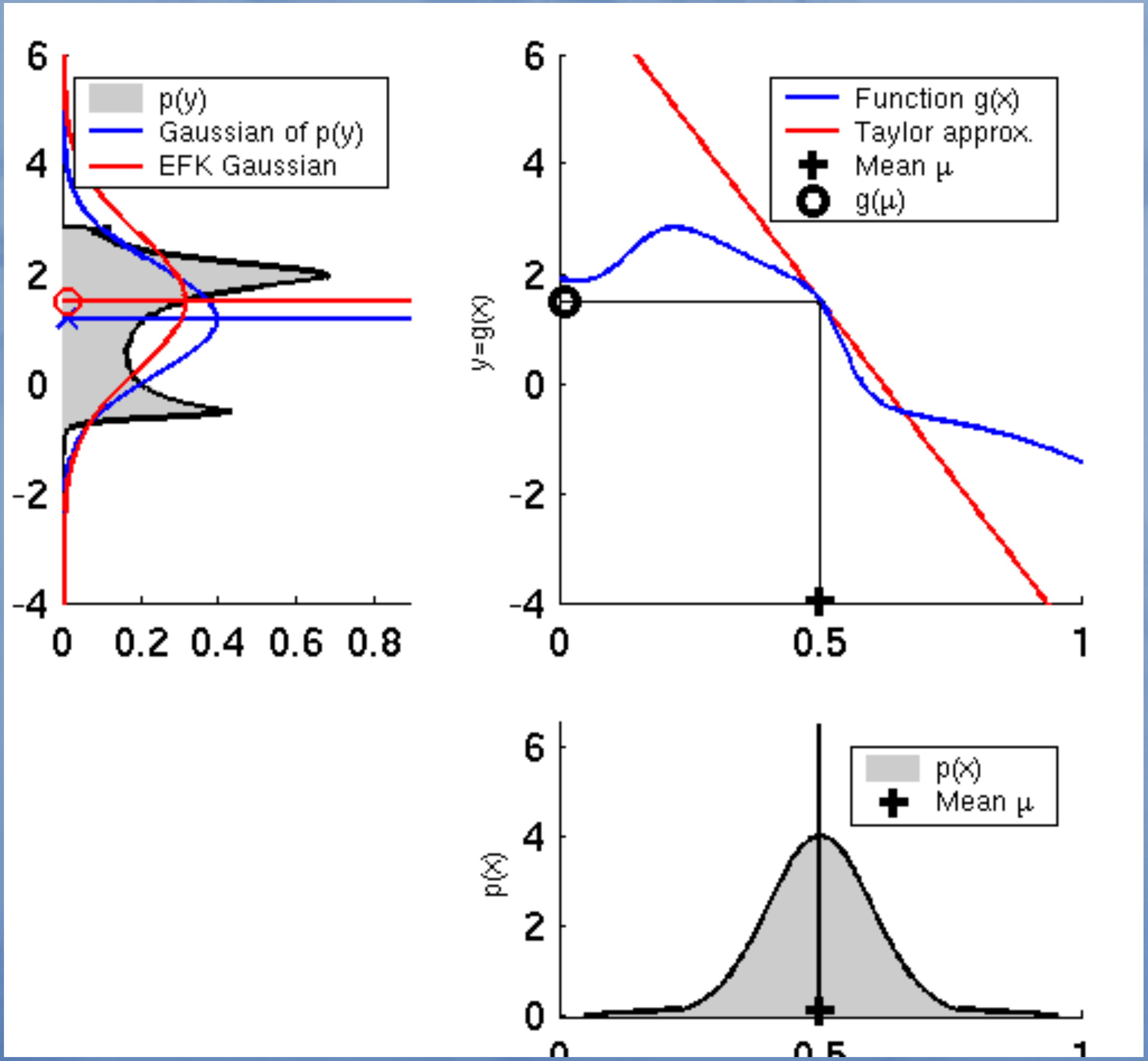
Linearity Assumption



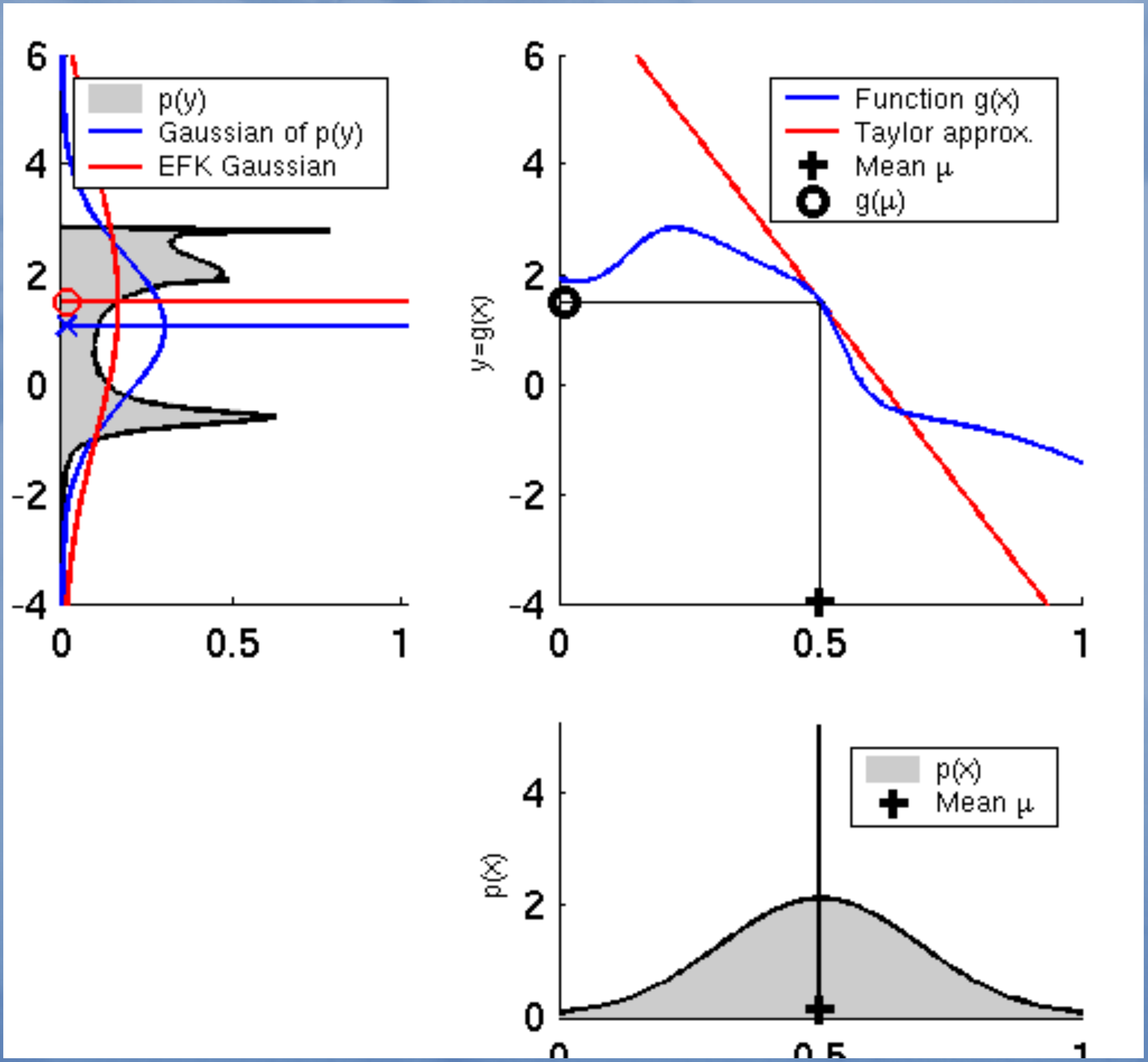
Non-Linear Function



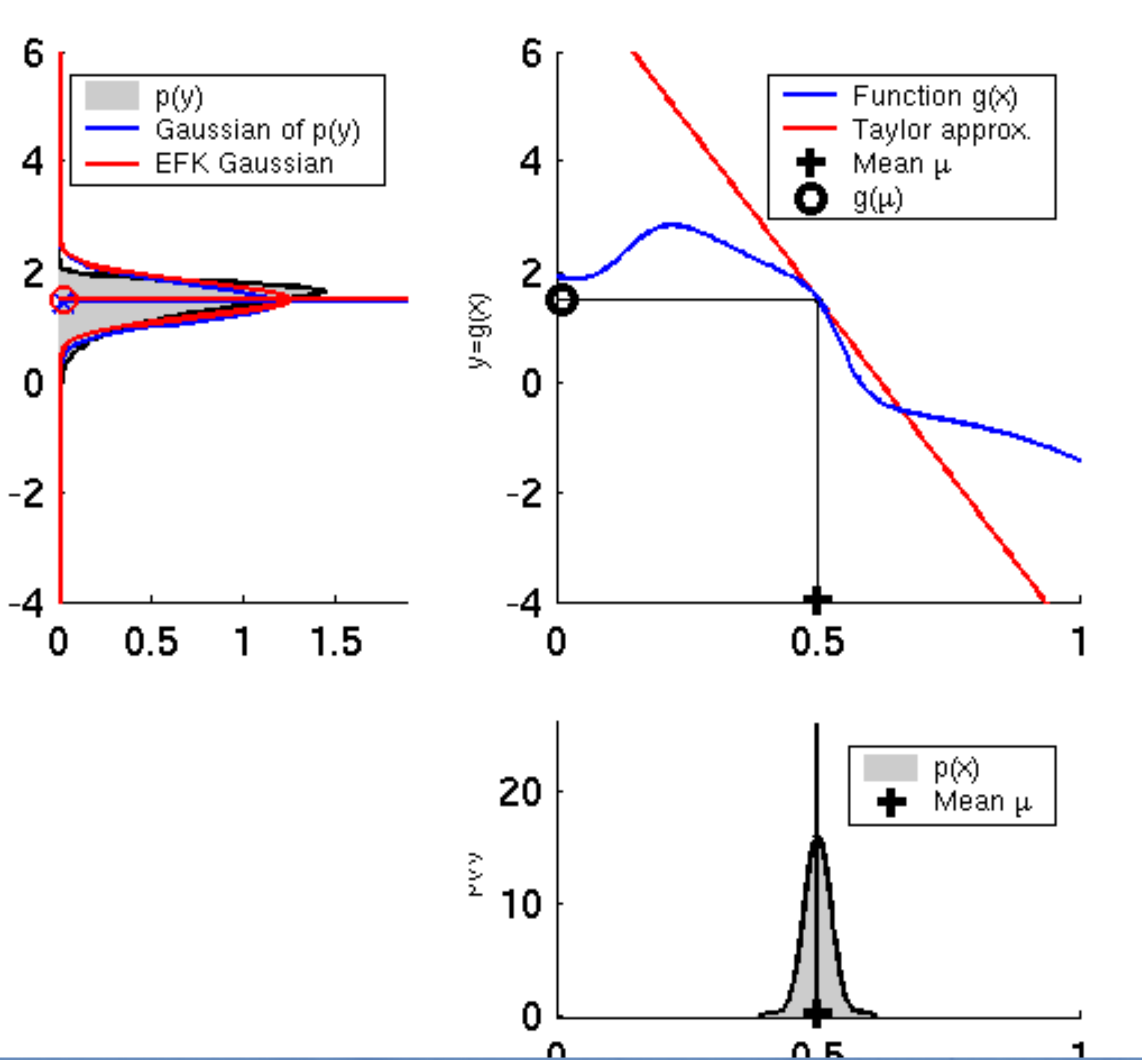
EKF Linearization (1)



EKF Linearization (2)



EKF Linearization (3)



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

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

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

EKF Summary

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

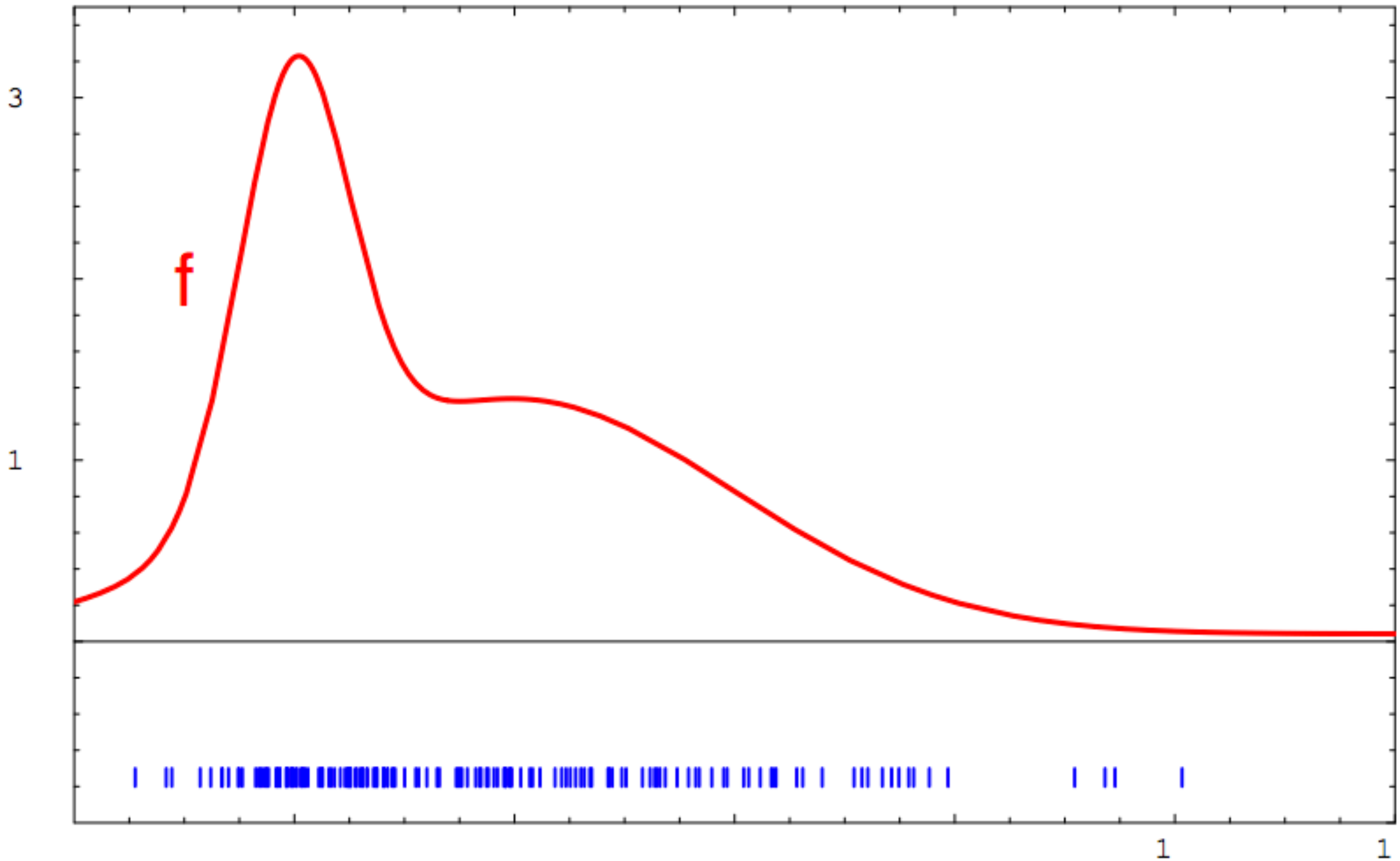
  $$O(k^{2.376} + n^2)$$

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

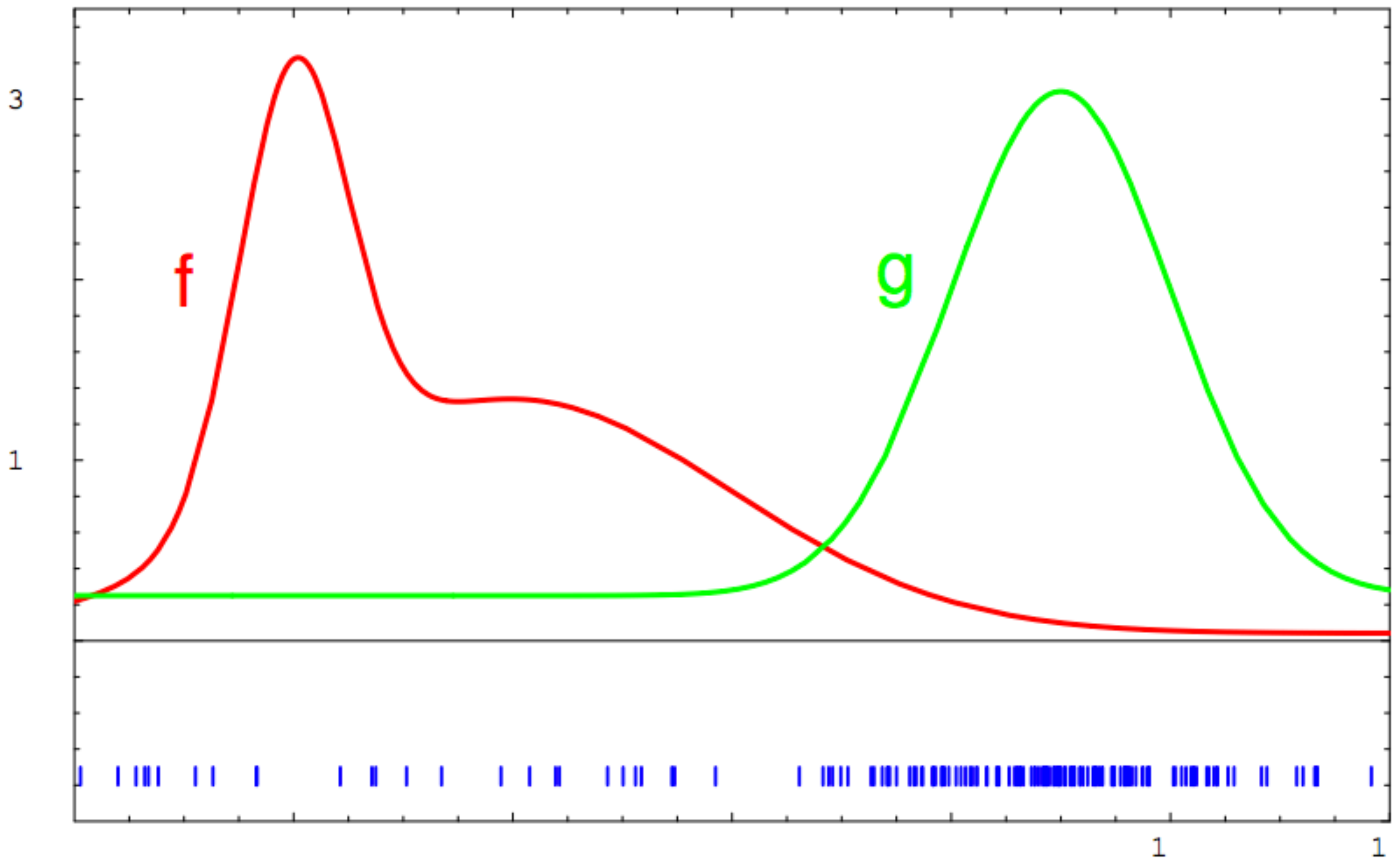
Particle Filters

- Represent belief by random samples
- Estimation of non-Gaussian, nonlinear processes
- Inherently represents multiple hypotheses
- Storage overhead: typically state is represented using one to few thousands of particles
- Compare that to mean, variance in a Kalman filter
- Computing correspondingly expensive as well because we have to move all the state forward

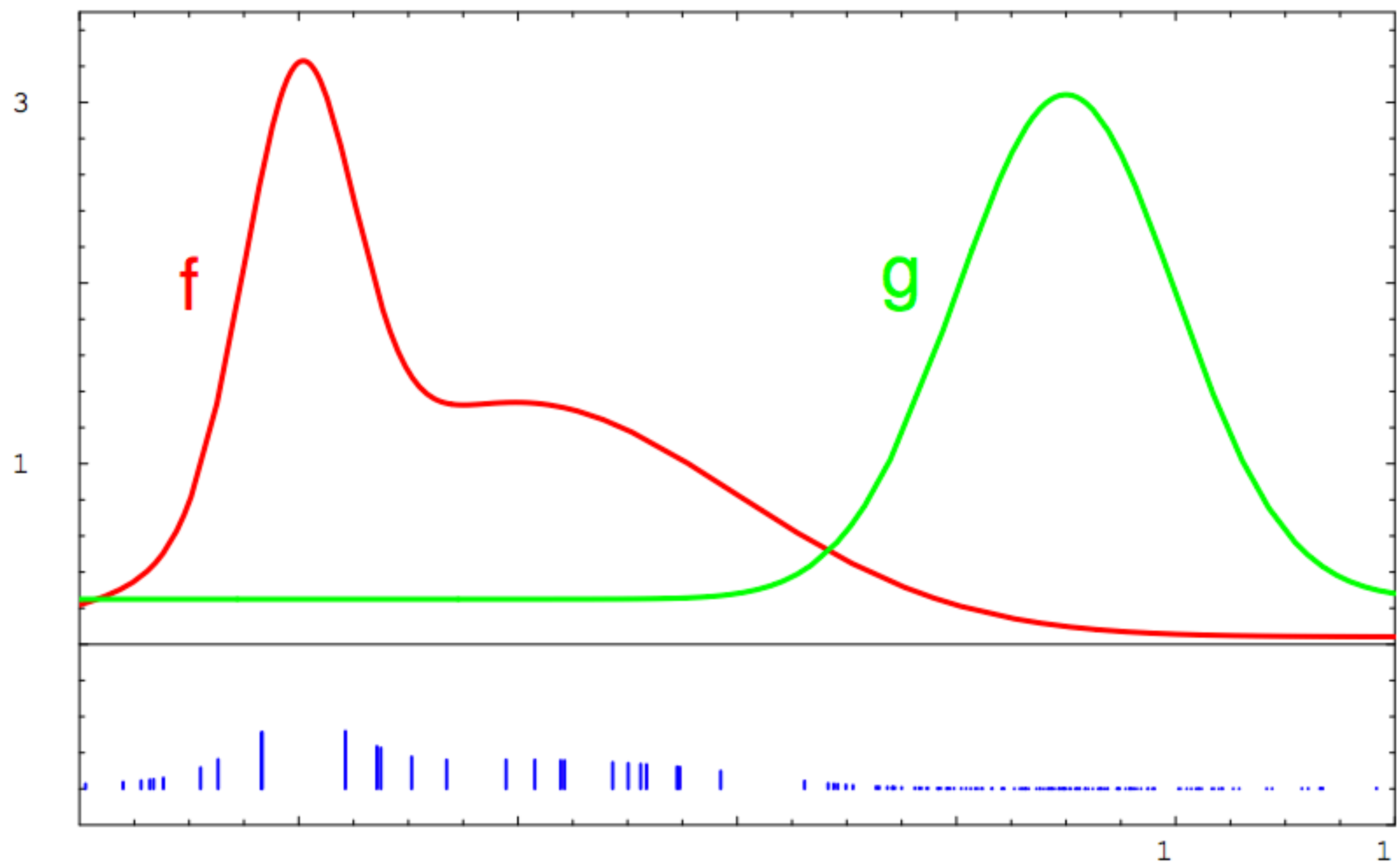
Importance Sampling



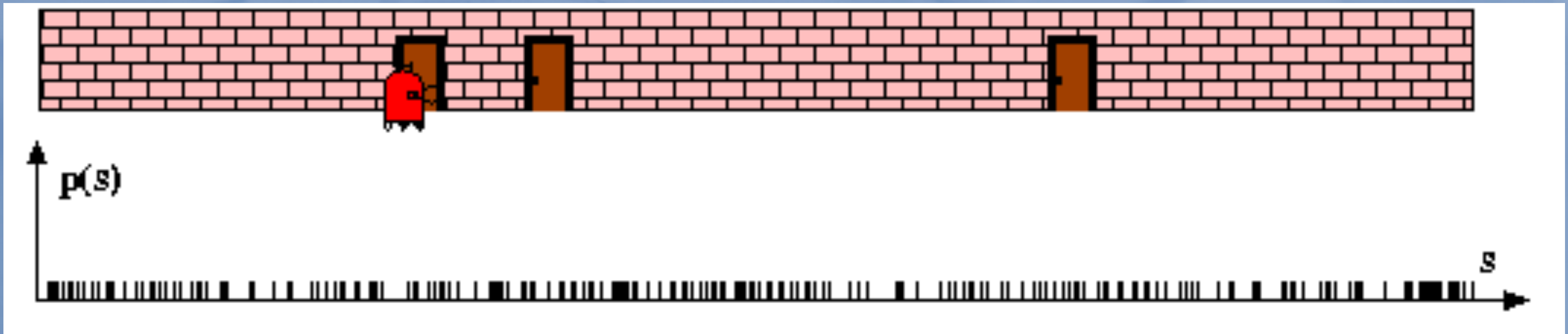
Importance Sampling (2)



Importance Sampling (3)



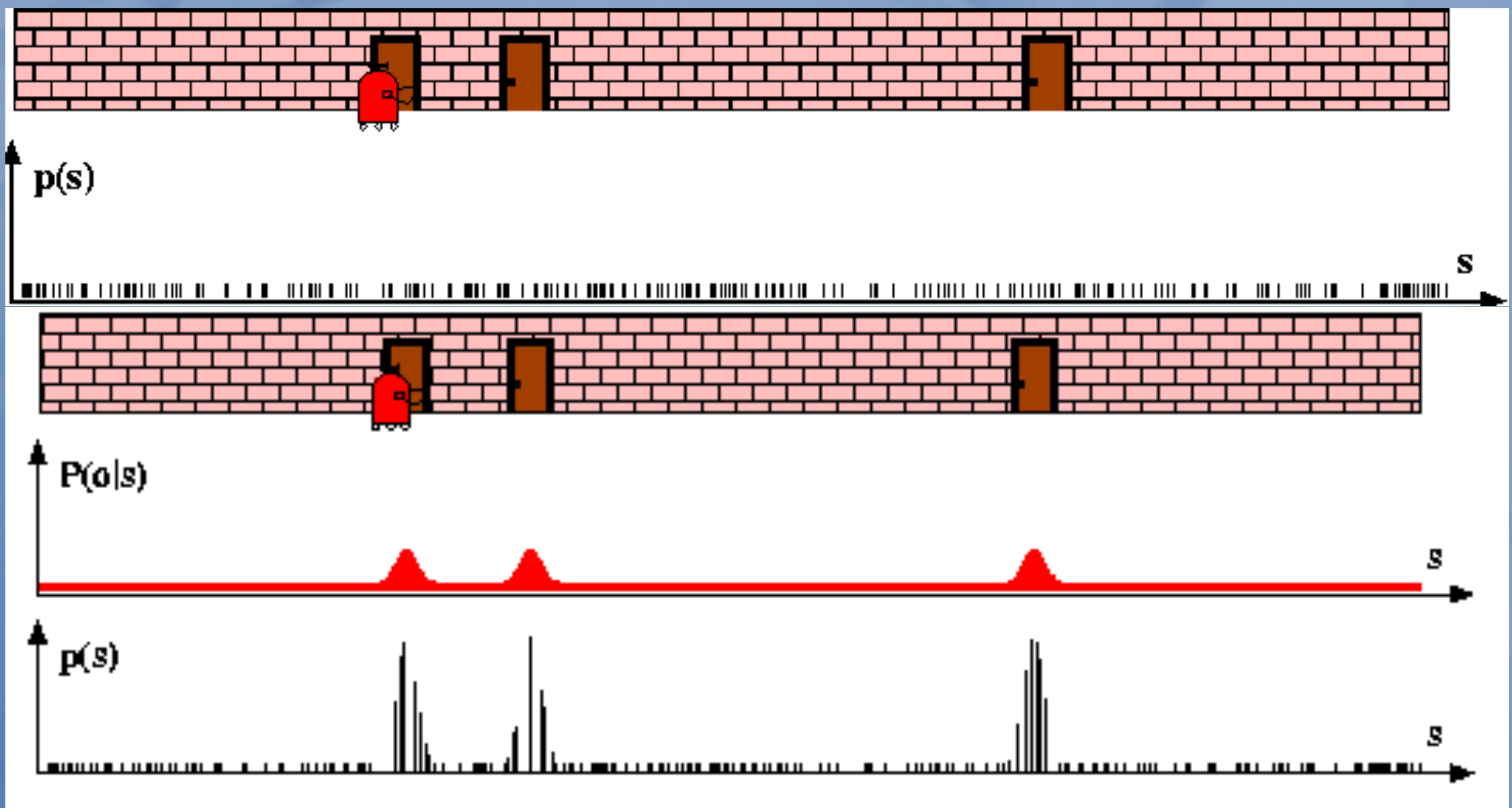
Particle Filter Example



Sensor Information: Importance Sampling

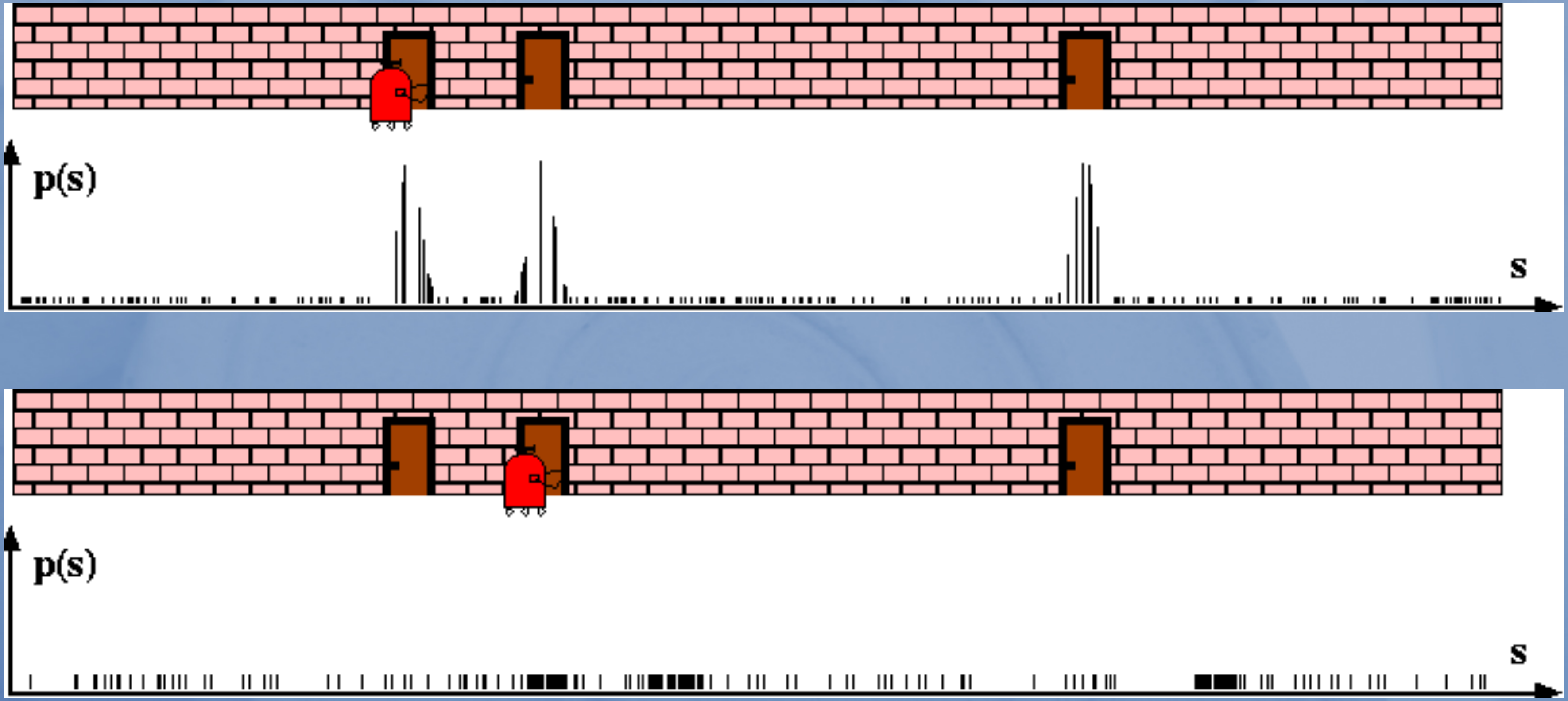
$$Bel(x) \quad \leftarrow \quad \alpha\ p(z \mid x)\ Bel^{-}(x)$$

$$w \quad \leftarrow \quad \frac{\alpha\ p(z \mid x)\ Bel^{-}(x)}{Bel^{-}(x)} \quad = \quad \alpha\ p(z \mid x)$$



Robot Motion

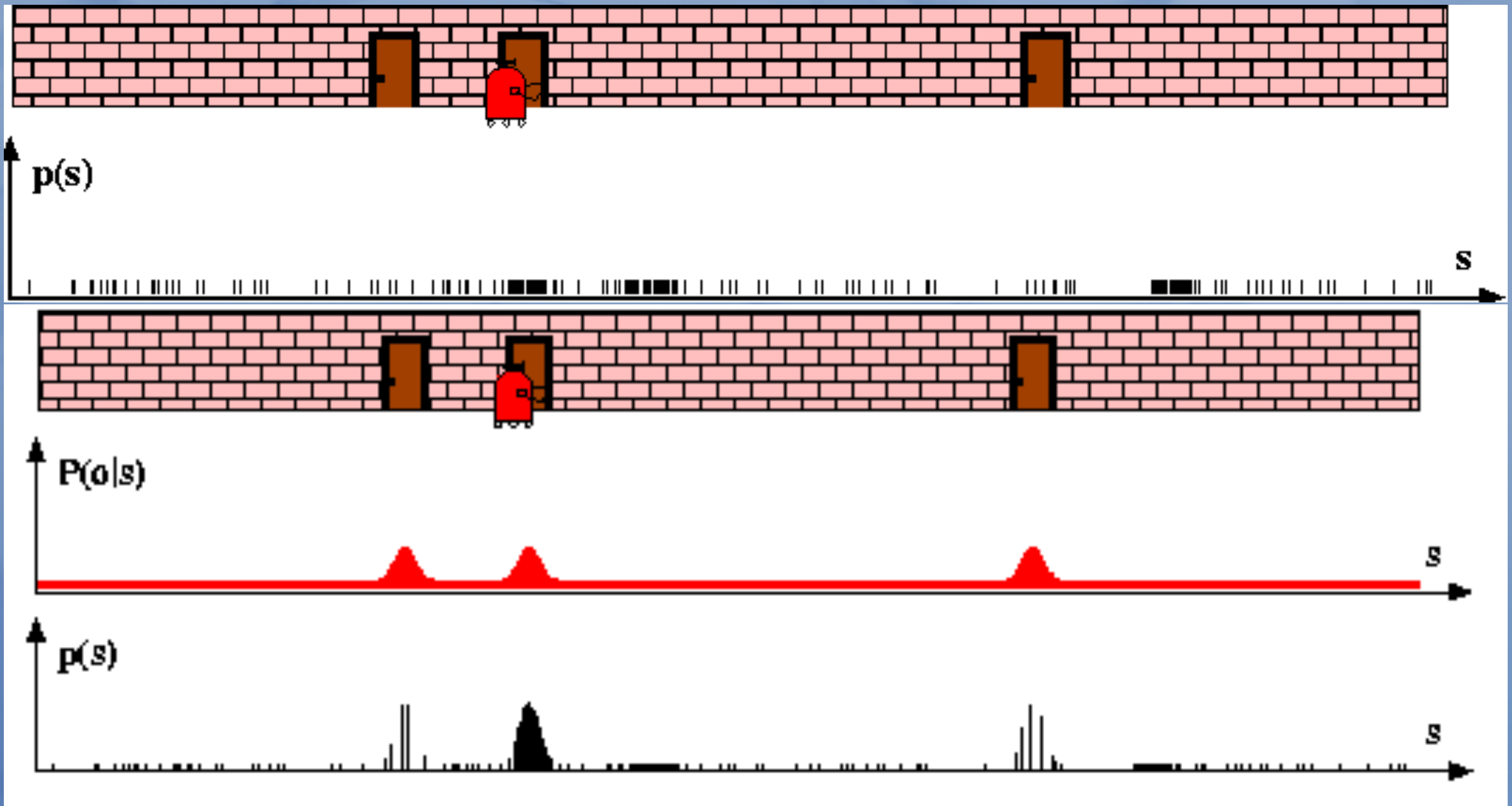
$$Bel^{-}(x) \leftarrow \int p(x \mid u, x')\, Bel(x')\, \mathrm{d}x'$$



Sensor Information: Importance Sampling

$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^-(x)$$

$$w \leftarrow \frac{\alpha\, p(z \mid x)\, Bel^-(x)}{Bel^-(x)} = \alpha\, p(z \mid x)$$



Robot Motion

$$Bel^{-}(x) \quad \leftarrow \quad \int p(x \mid u, x')\, Bel(x')\ \mathrm{d}x'$$