# Markov Localization & Bayes Filtering

with

Kalman Filters

Discrete Filters

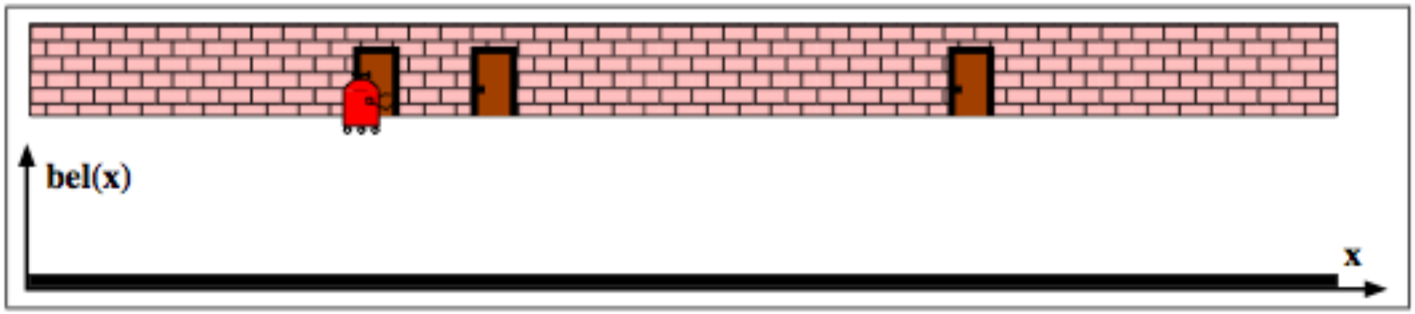
Particle Filters

Slides adapted from Thrun et al.,

Probabilistic Robotics

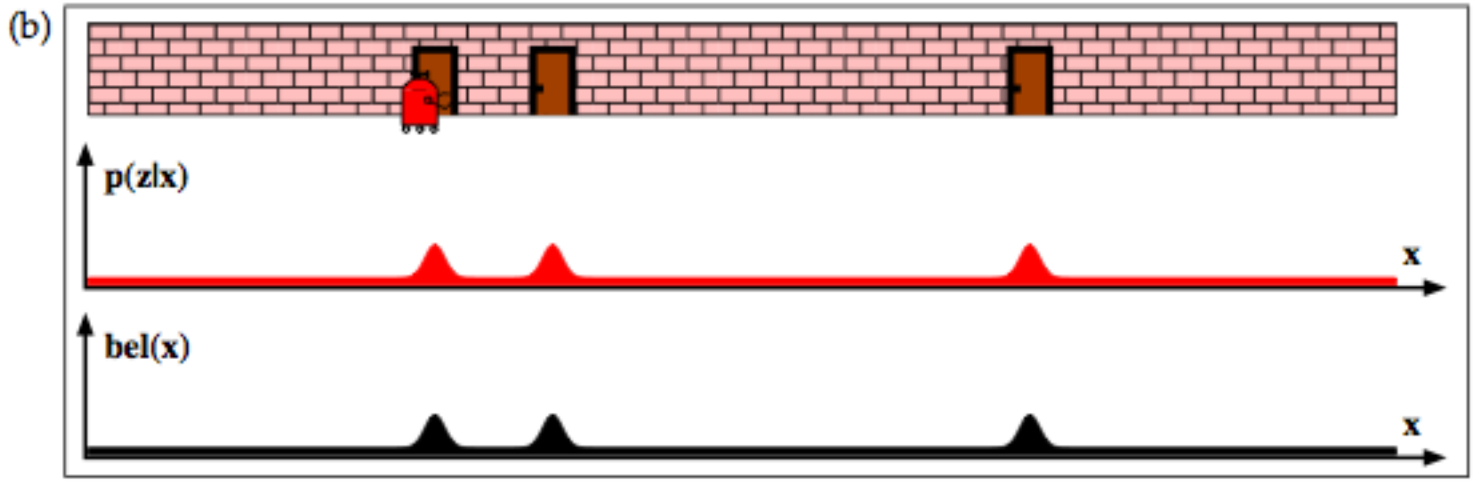
# Markov Localization

(a)




The robot doesn't know where it is. Thus, a reasonable initial believe of it's position is a uniform distribution.

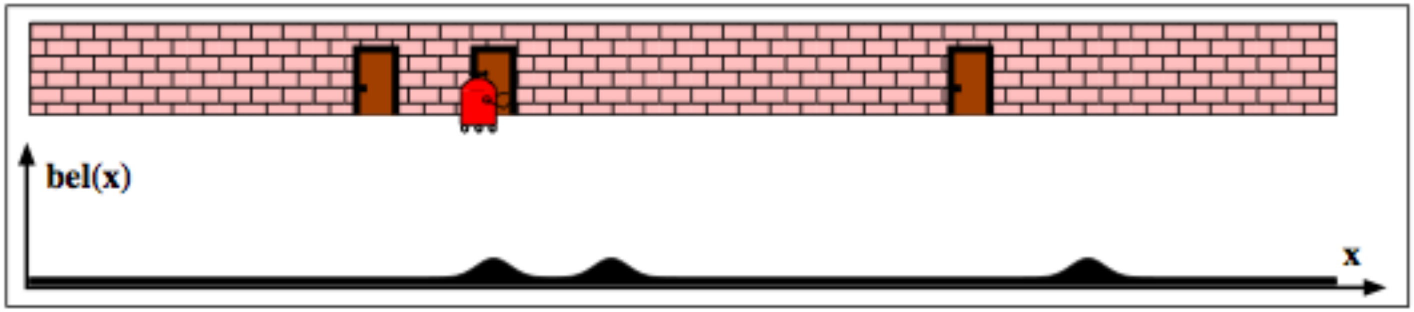
# Markov Localization




A sensor reading is made (USE SENSOR MODEL) indicating a door at certain locations (USE MAP). This sensor reading should be integrated with prior believe to update our believe (USE BAYES).

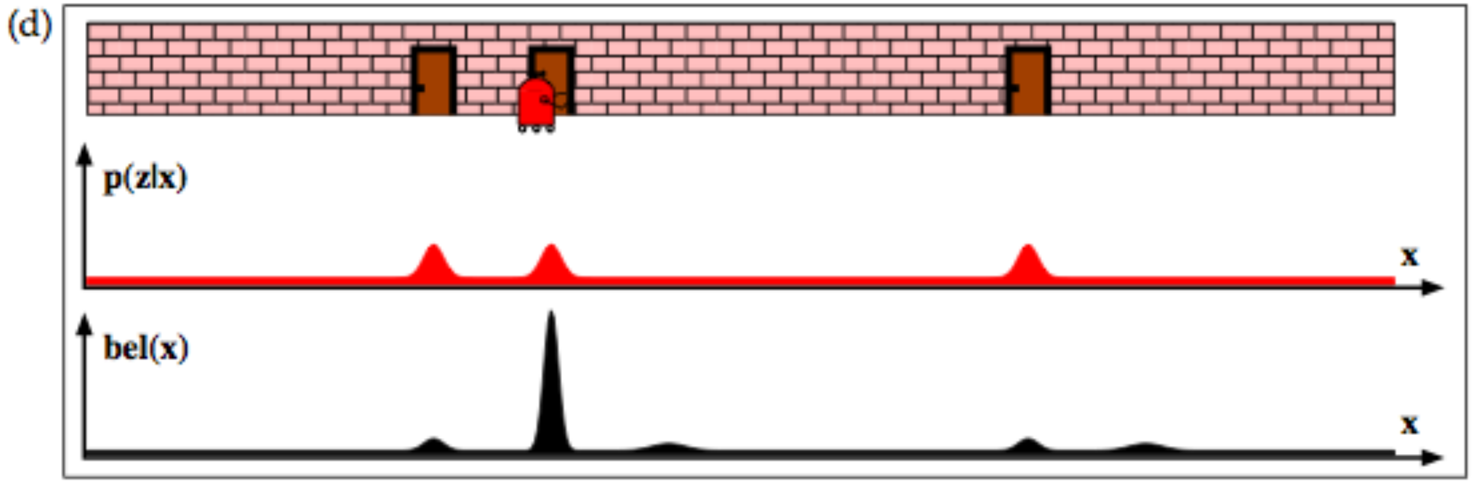
# Markov Localization

(c)



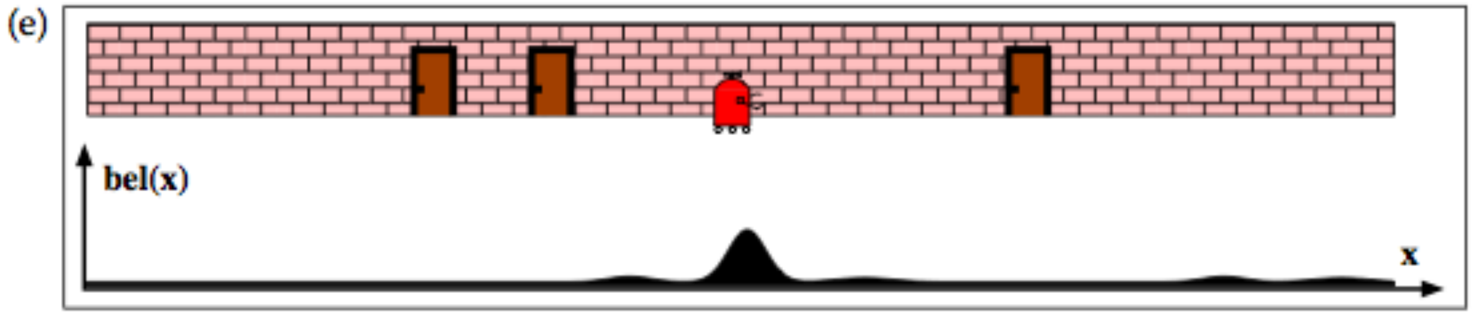

The robot is moving (USE MOTION MODEL) which adds noise.

# Markov Localization




A new sensor reading (USE SENSOR MODEL) indicates a door at certain locations (USE MAP). This sensor reading should be integrated with prior believe to update our believe (USE BAYES).

# Markov Localization




The robot is moving (USE MOTION MODEL) which adds noise. …

# Recursive Bayesian Updating

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x, z_1, \ldots, z_{n-1})\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

**Markov assumption**: $z_n$ is independent of $z_1, \ldots, z_{n-1}$ if we know *x.*

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

$$= \eta\, P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})$$

$$= \eta_{1 \ldots n} \prod_{i=1 \ldots n} P(z_i \mid x)\, P(x)$$

# Putting oberservations and actions together: Bayes Filters

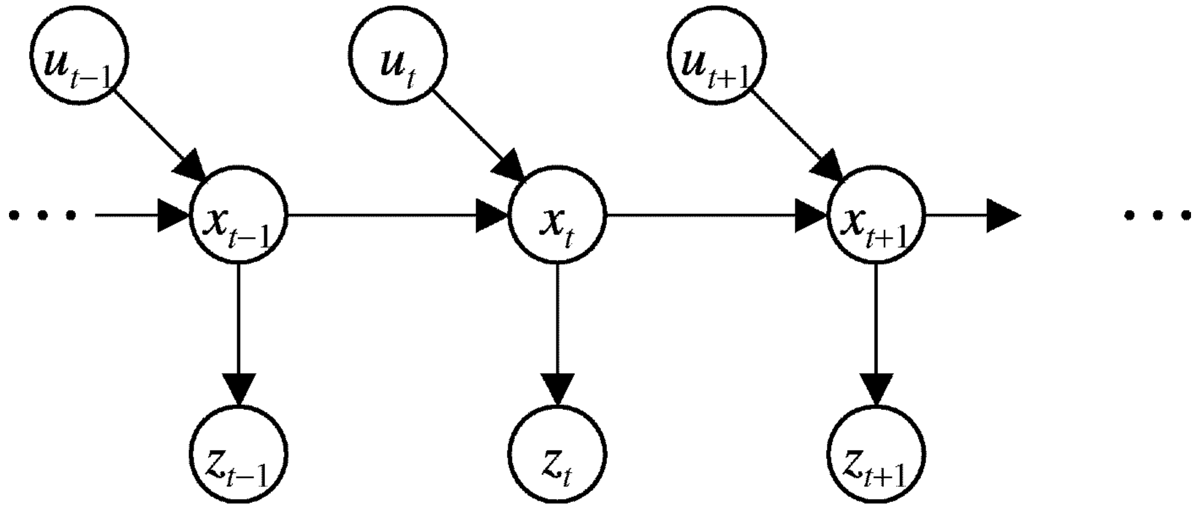
- **Given:**
  - Stream of observations *z* and action data *u:*

$$d_t = \{u_1, z_1 \ldots, u_t, z_t\}$$

  - Sensor model *P(z|x).*
  - Action model *P(x|u,x')*.
  - Prior probability of the system state *P(x).*
- **Wanted:**
  - Estimate of the state *X* of a dynamical system.
  - The posterior of the state is also called **Belief**:

$$Bel(x_t) = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

# Graphical Representation and Markov Assumption




$$p(z_t \mid x_{0:t}, z_{1:t}, u_{1:t}) = p(z_t \mid x_t)$$

$$p(x_t \mid x_{1:t-1}, z_{1:t}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

Underlying Assumptions

- Static world
- Independent noise
- Perfect model, no approximation errors

# Bayes Filters

$z$ = observation
$u$ = action
$x$ = state

$$\boxed{Bel(x_t)} = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

**Bayes** $$= \eta \; P(z_t \mid x_t, u_1, z_1, \ldots, u_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

**Markov** $$= \eta \; P(z_t \mid x_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

**Total prob.** $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \; dx_{t-1}$$

**Markov** $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \; dx_{t-1}$$

**Markov** $$= \eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) P(x_{t-1} \mid u_1, z_1, \ldots, z_{t-1}) \; dx_{t-1}$$

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

- Prediction

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

- Correction

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

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

Algorithm **Bayes_filter**( *Bel(x),d* ):

$\eta$=0

If *d* is a perceptual data item *z* then

  For all *x* do

    $Bel'(x) = P(z \mid x) Bel(x)$

    $\eta = \eta + Bel'(x)$

  For all *x* do

    $Bel'(x) = \eta^{-1} Bel'(x)$

Else if *d* is an action data item *u* then

  For all *x* do

    $Bel'(x) = \int P(x \mid u, x') \; Bel(x') \; dx'$

Return *Bel'(x)*

# Kalman Filter Algorithm

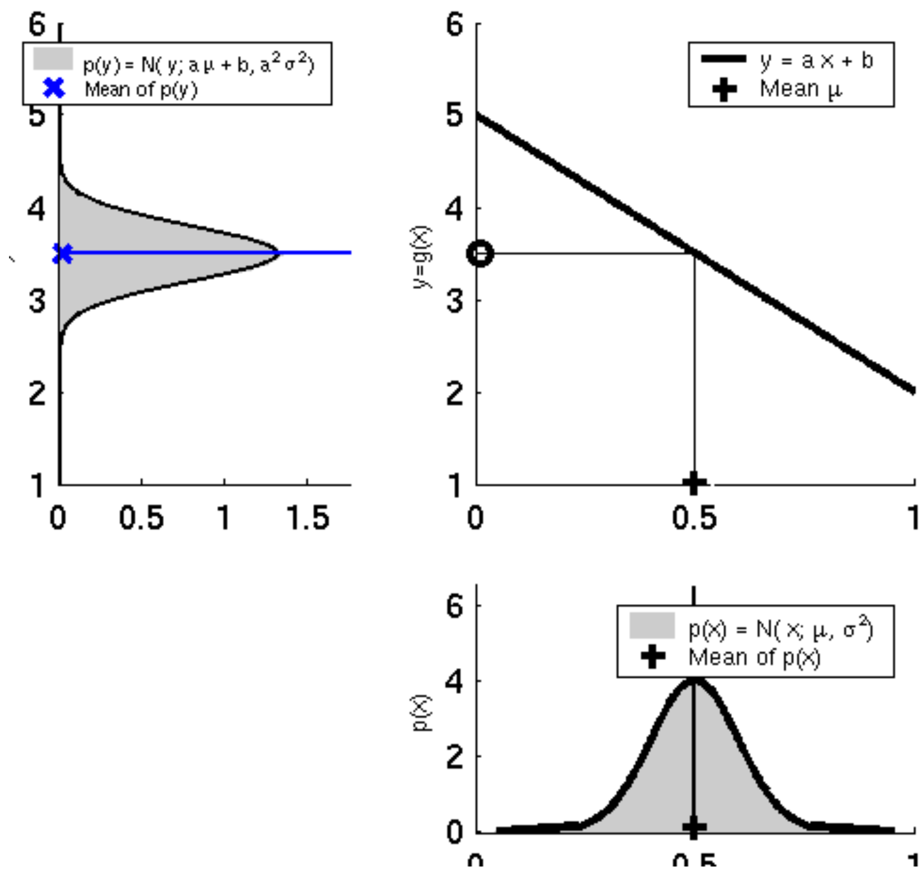
1. Algorithm **Kalman_filter**( $\mu_{t-1}$, $\Sigma_{t-1}$, $u_t$, $z_t$):

2. Prediction:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

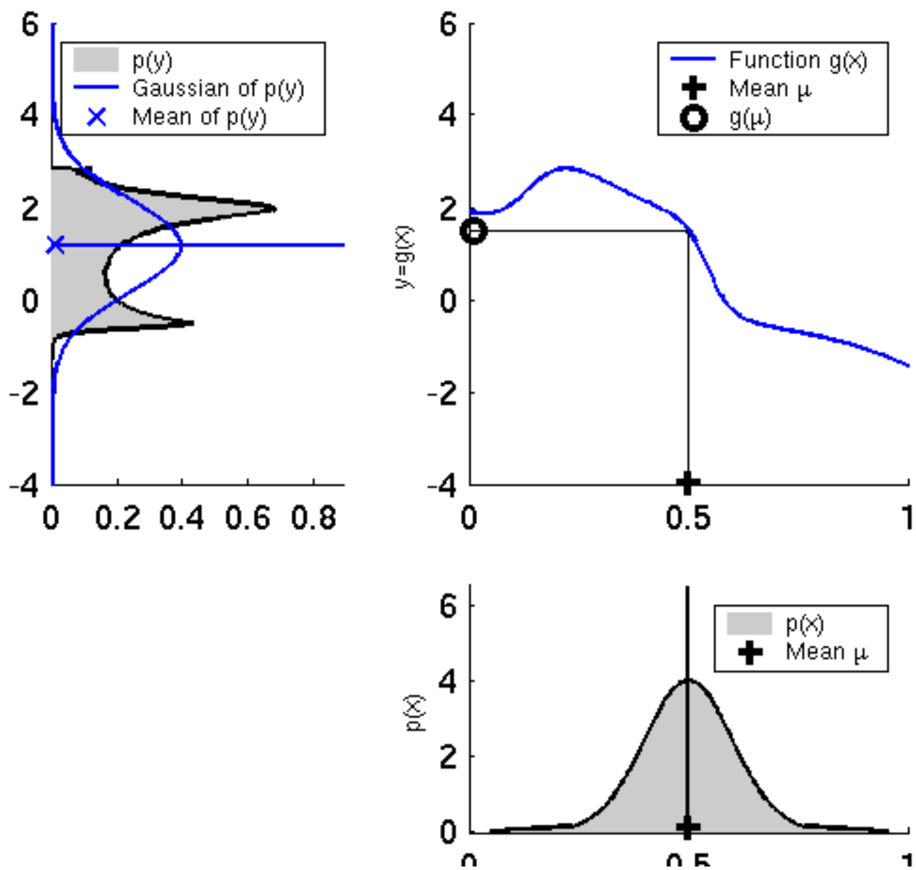
5. Correction:
6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

9. Return $\mu_t$, $\Sigma_t$

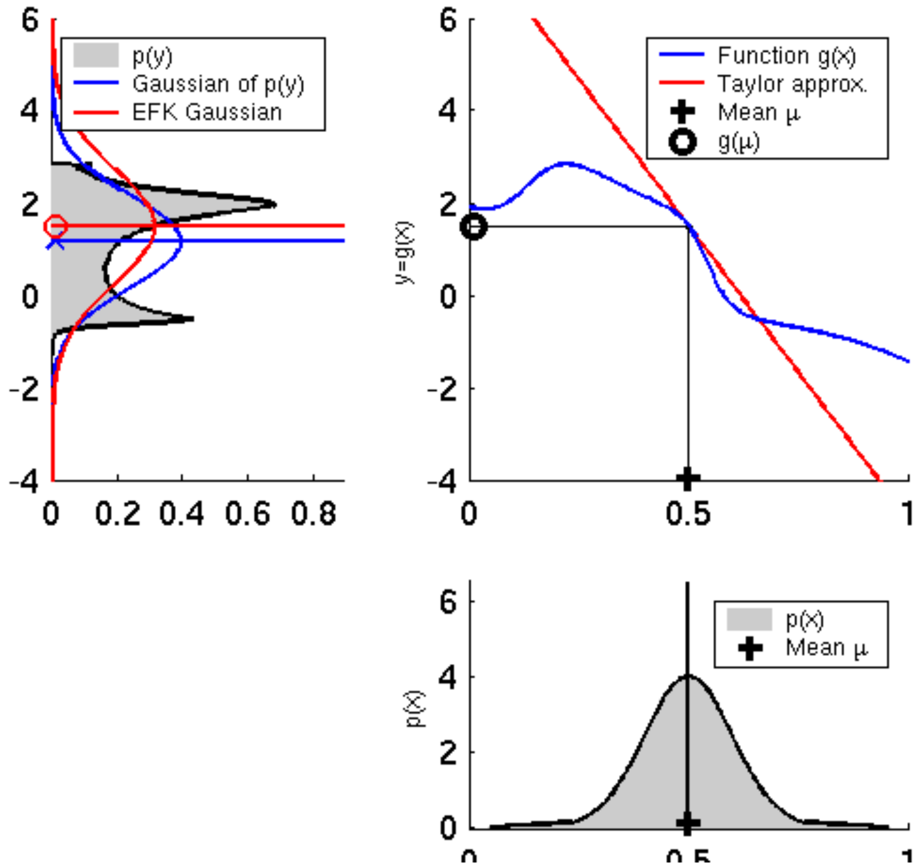
# Linearity Assumption Revisited



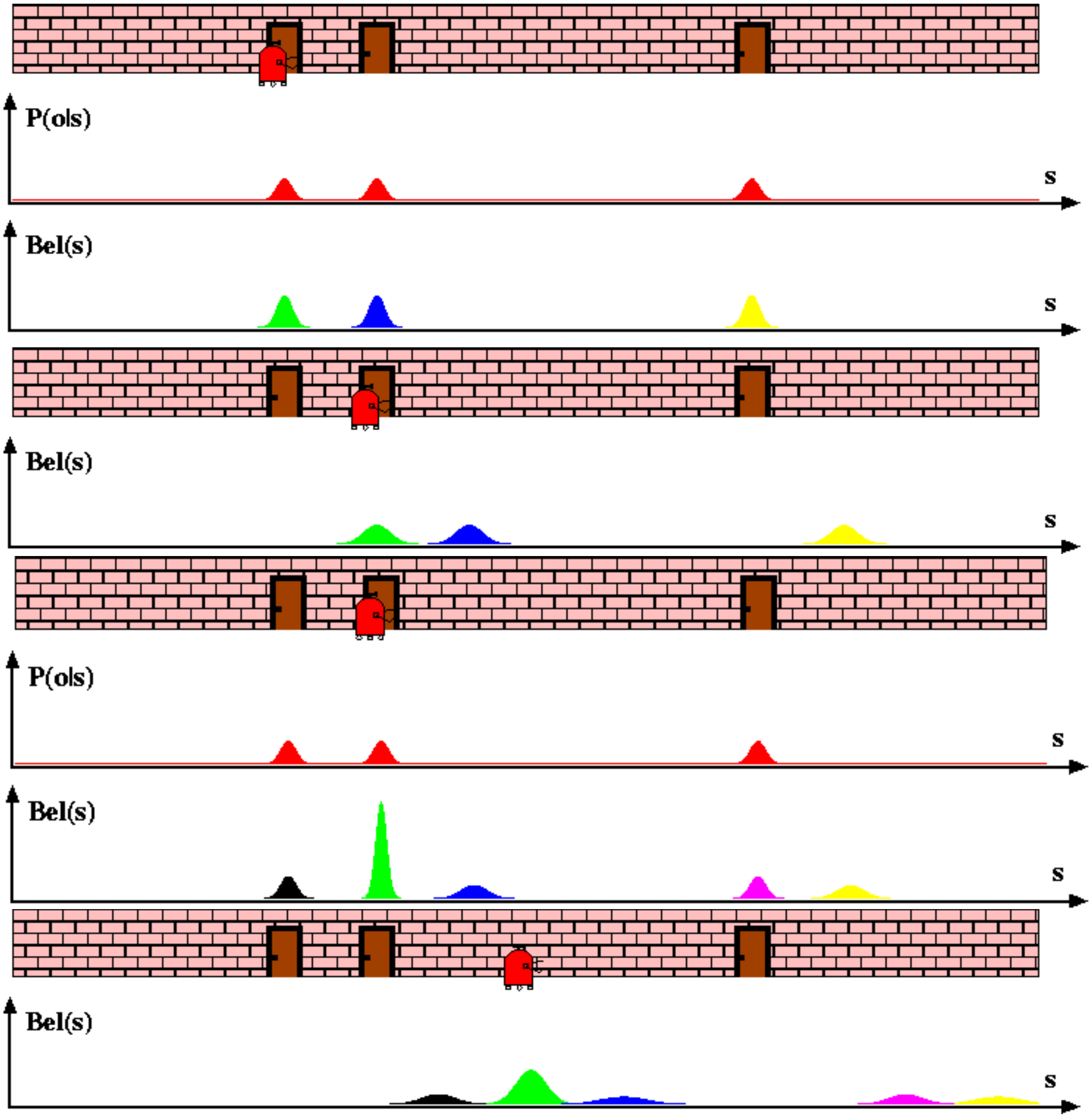
# Non-linear Function



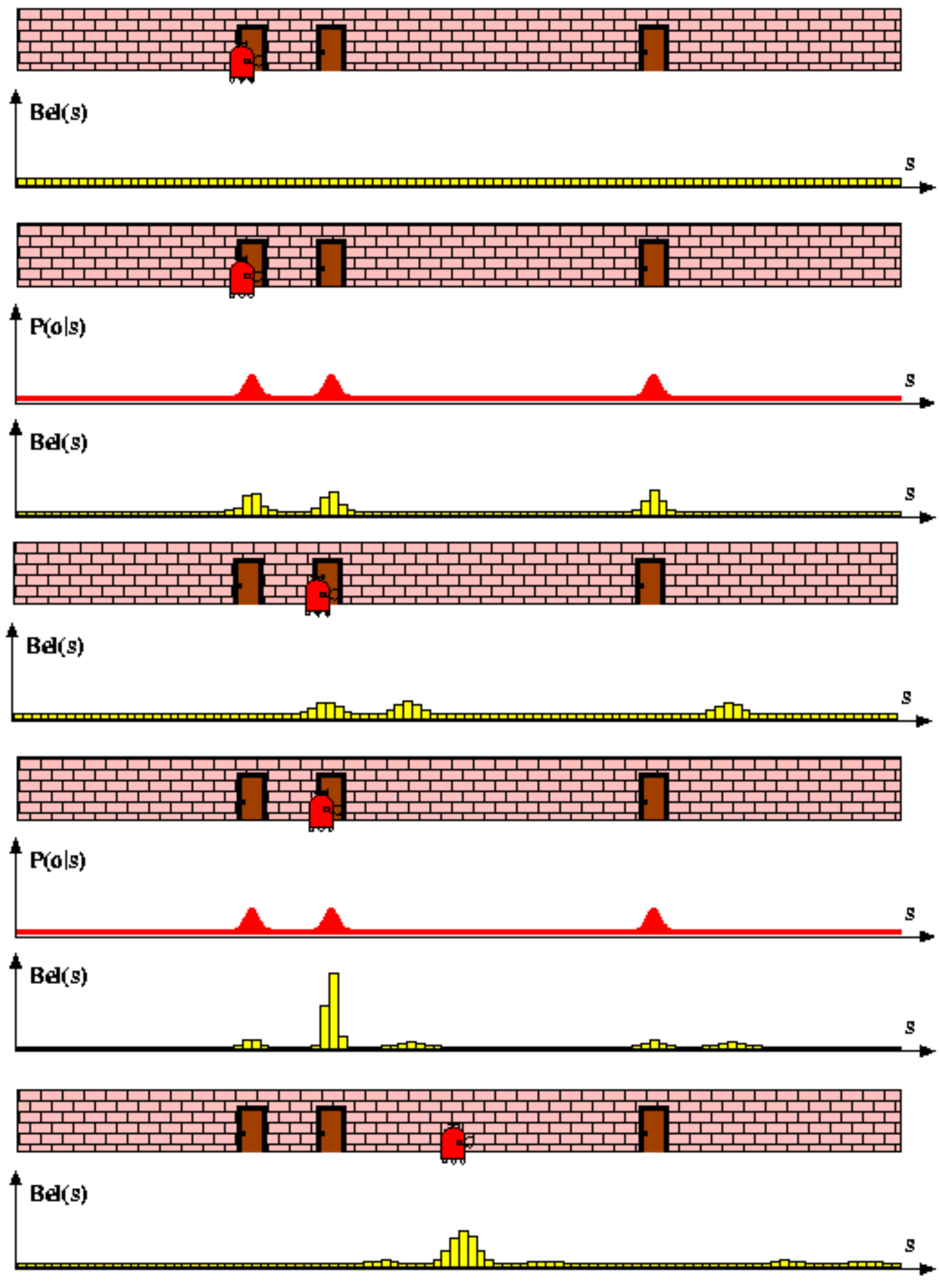
# EKF Linearization (1)



# Multi-hypothesis Tracking



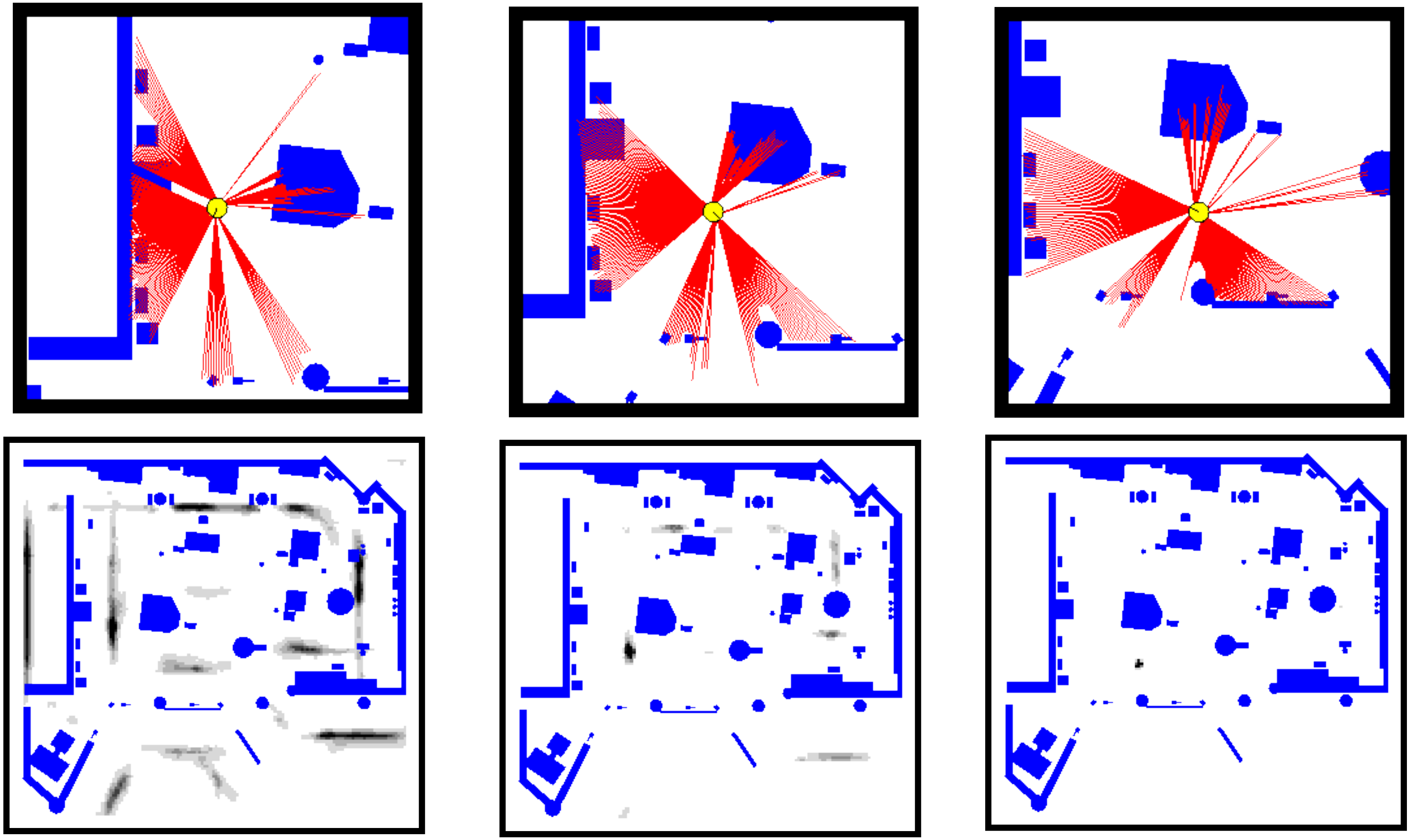
# Piecewise Constant



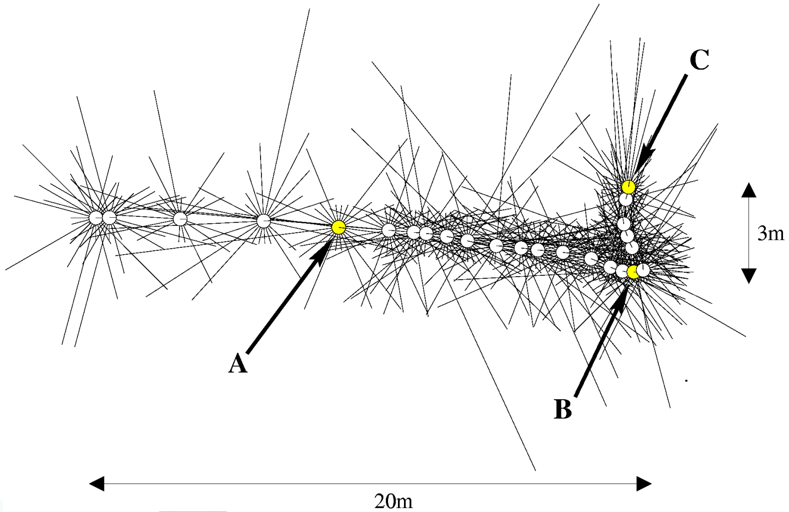
# Discrete Bayes Filter Algorithm

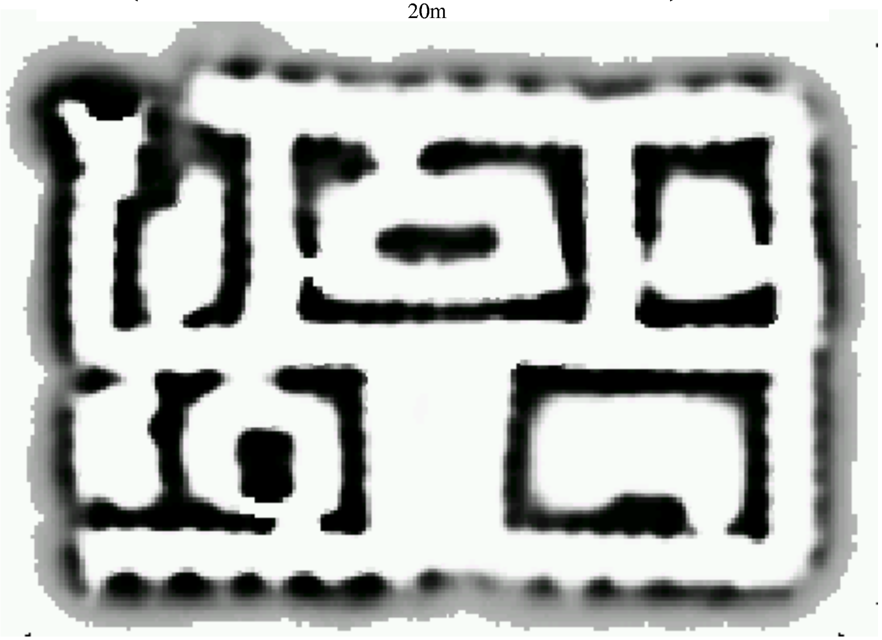
1. Algorithm **Discrete_Bayes_filter**( *Bel(x),d* ):
2. $\eta$=0
3. If *d* is a perceptual data item *z* then
4. For all *x* do
5. $$Bel'(x) = P(z \mid x) Bel(x)$$
6. $$\eta = \eta + Bel'(x)$$
7. For all *x* do
8. $$Bel'(x) = \eta^{-1} Bel'(x)$$
9. Else if *d* is an action data item *u* then
10. For all *x* do
11. $$Bel'(x) = \sum_{x'} P(x \mid u, x')\, Bel(x')$$
12. Return *Bel'(x)*

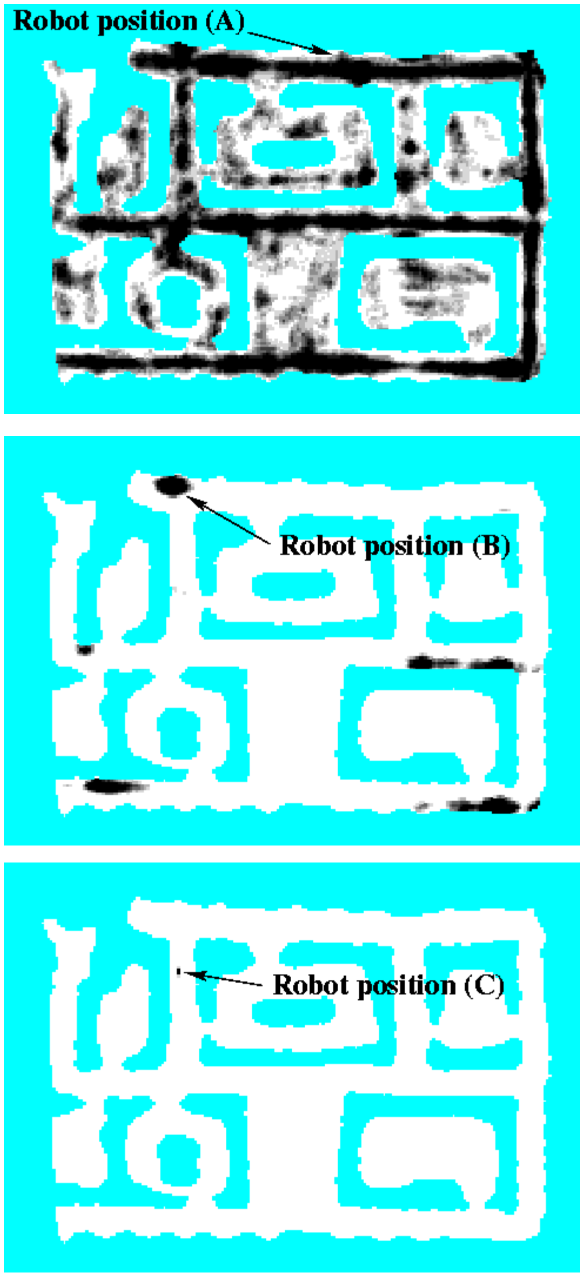
# Grid-based Localization



# Sonars and Occupancy Grid Map



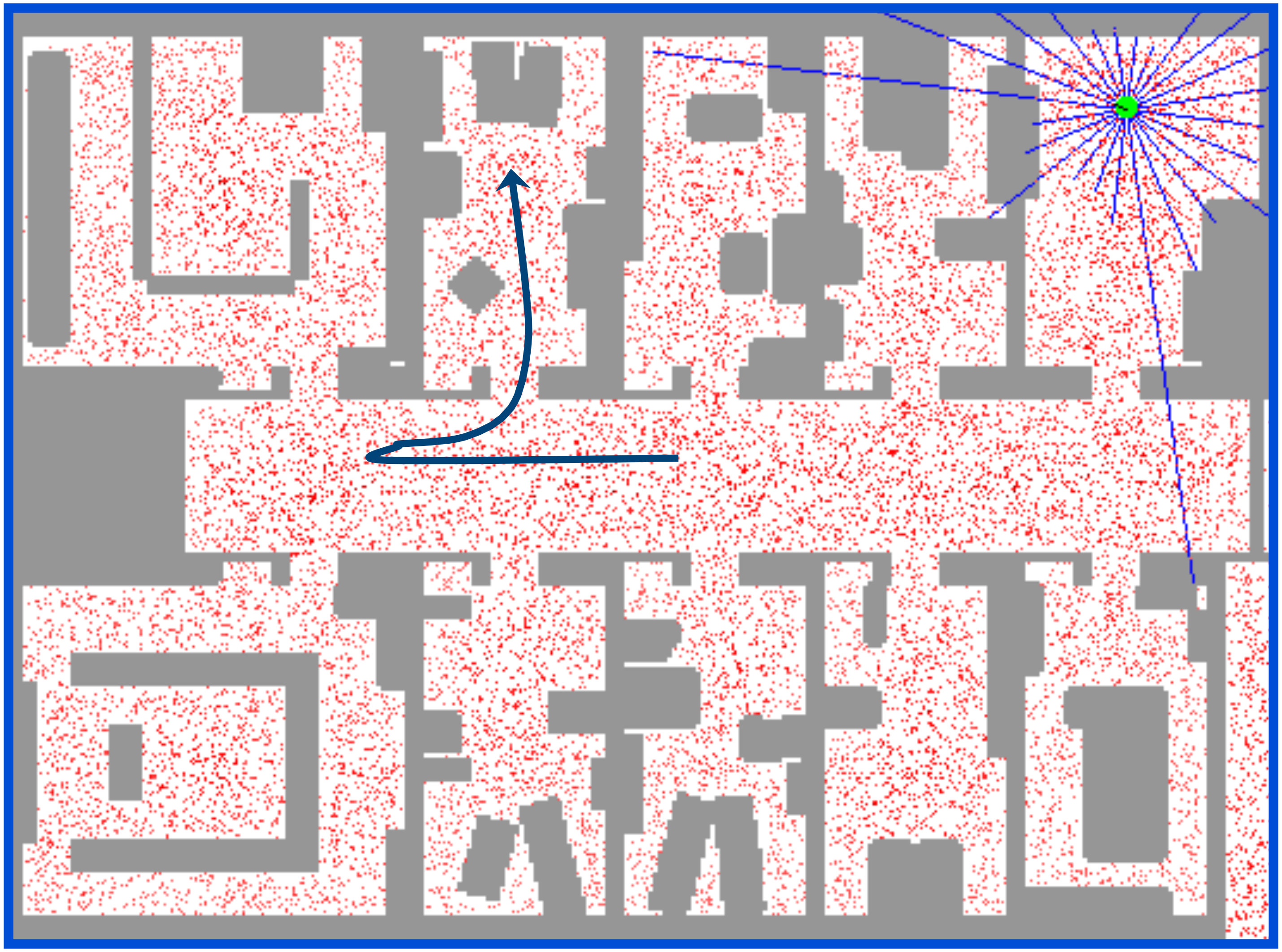


# Probabilistic Robotics

## **Bayes Filter Implementations**

Particle filters

# Sample-based Localization (sonar)



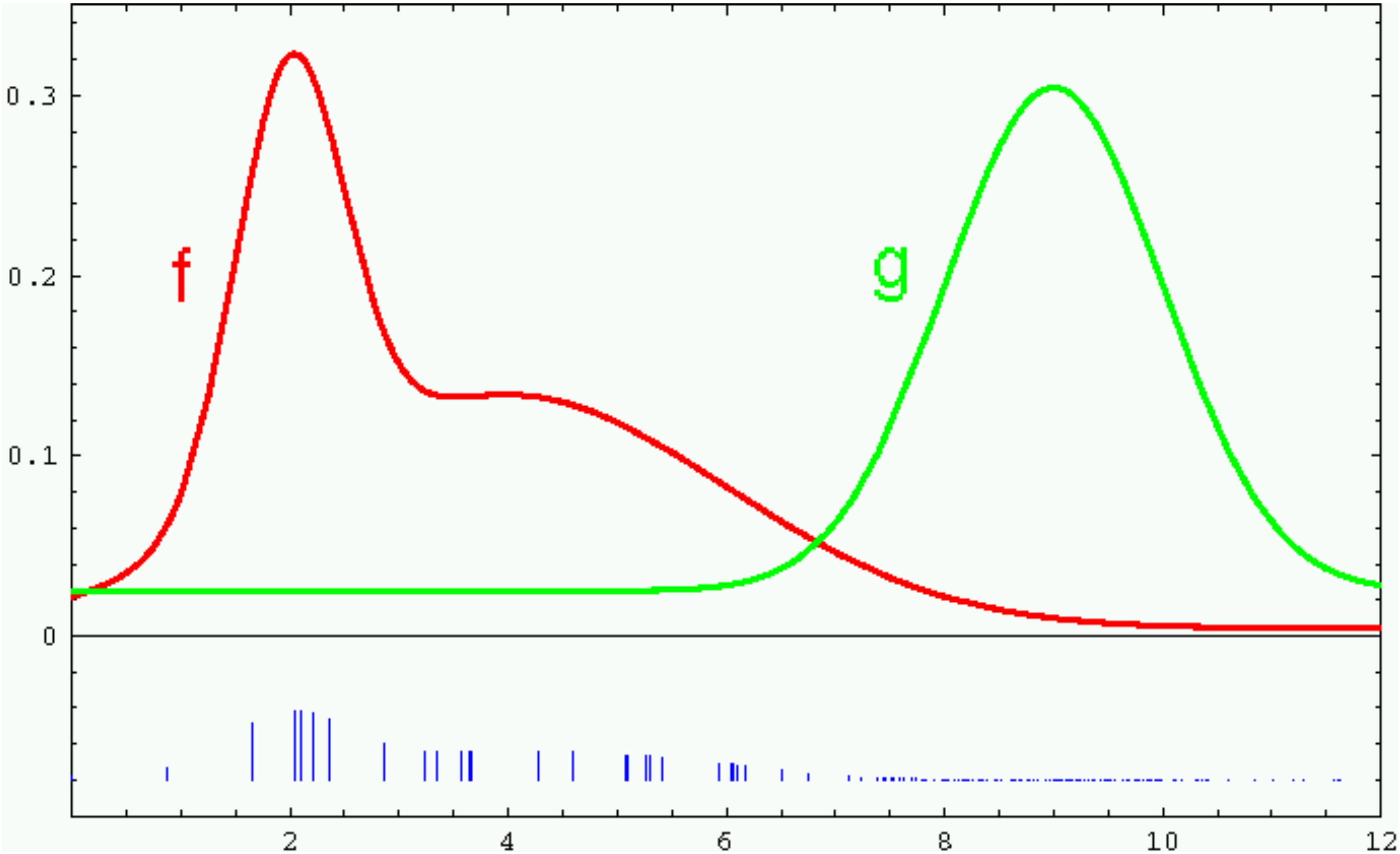
# Particle Filters

- Represent belief by random samples
- Estimation of non-Gaussian, nonlinear processes

- Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter, Particle filter

- Filtering: [Rubin, 88], [Gordon et al., 93], [Kitagawa 96]
- Computer vision: [Isard and Blake 96, 98]
- Dynamic Bayesian Networks: [Kanazawa et al., 95]d

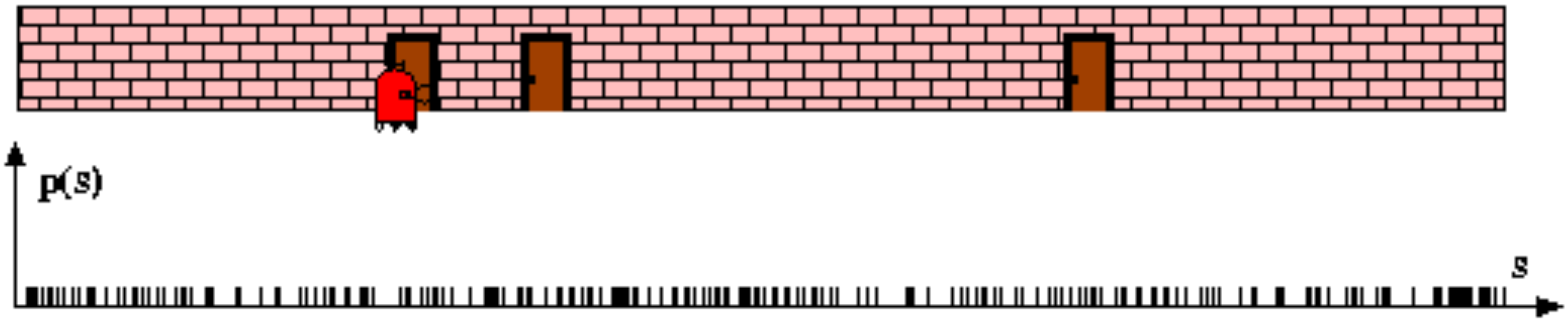
# Importance Sampling




**Weight samples:** $w = f / g$

# Importance Sampling with Resampling: Landmark Detection Example

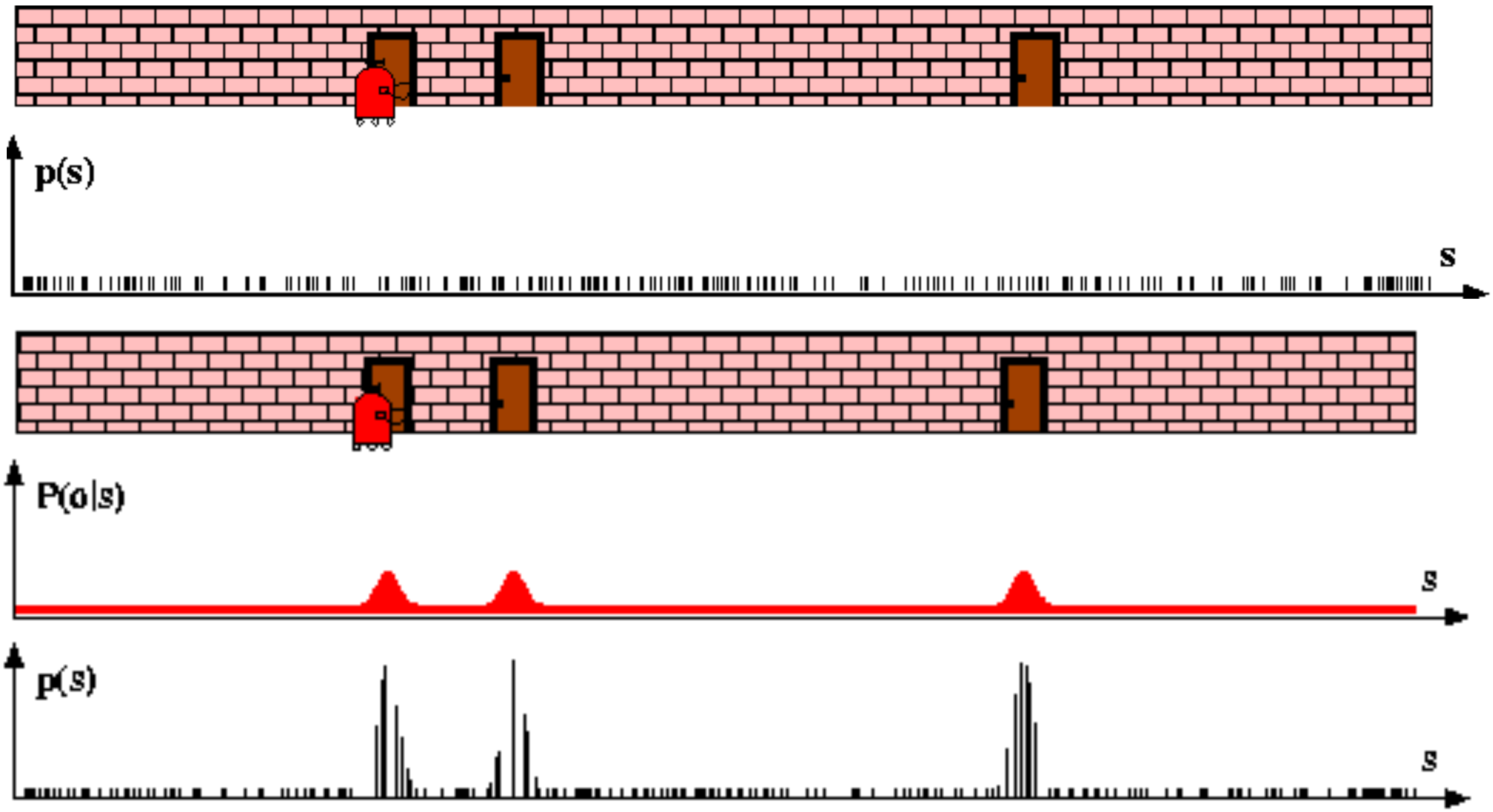
# Particle Filters



# Sensor Information: Importance Sampling

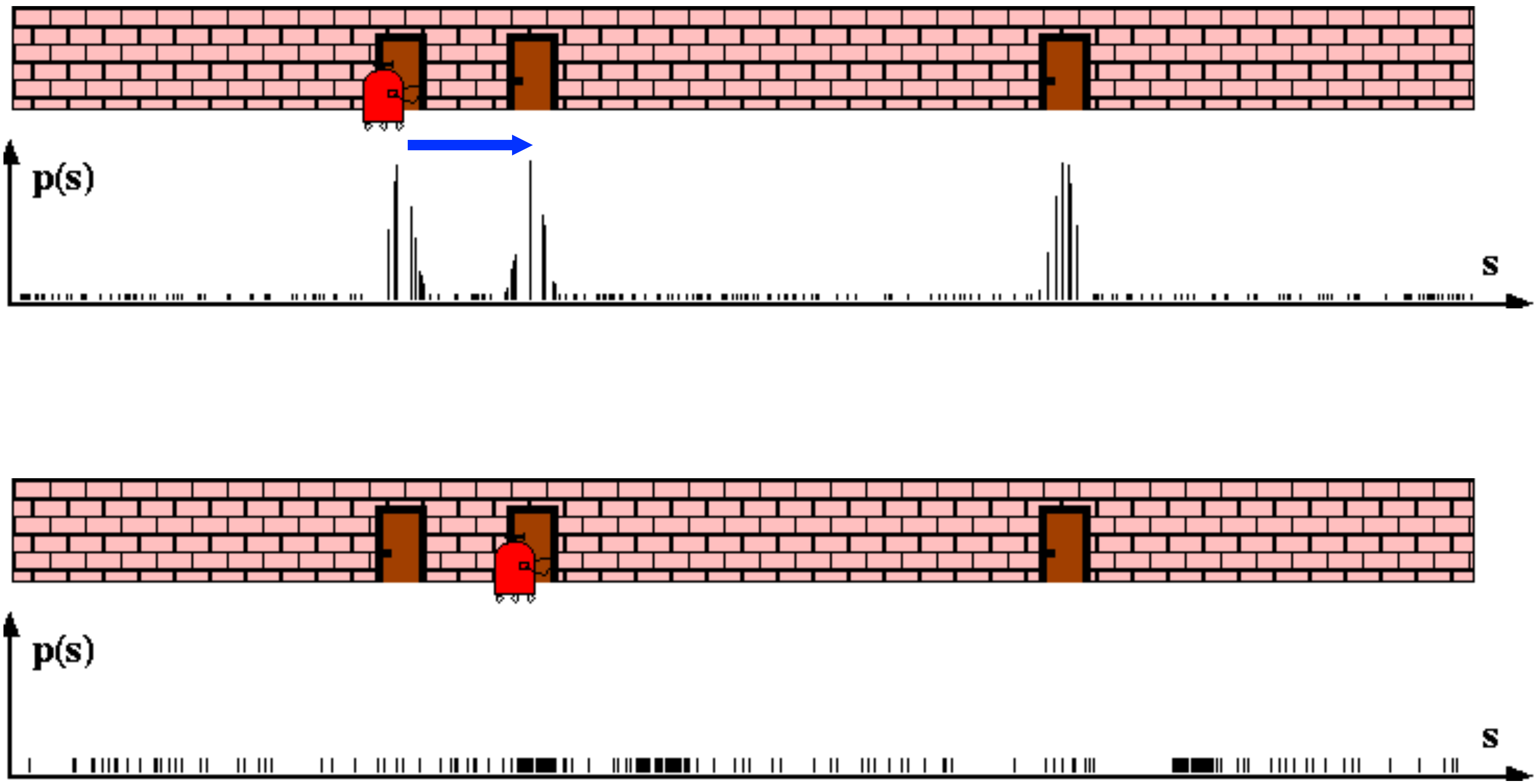
$$Bel(x) \quad \leftarrow \quad \alpha\ p(z \mid x)\ Bel^{-}(x)$$

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



# Robot Motion

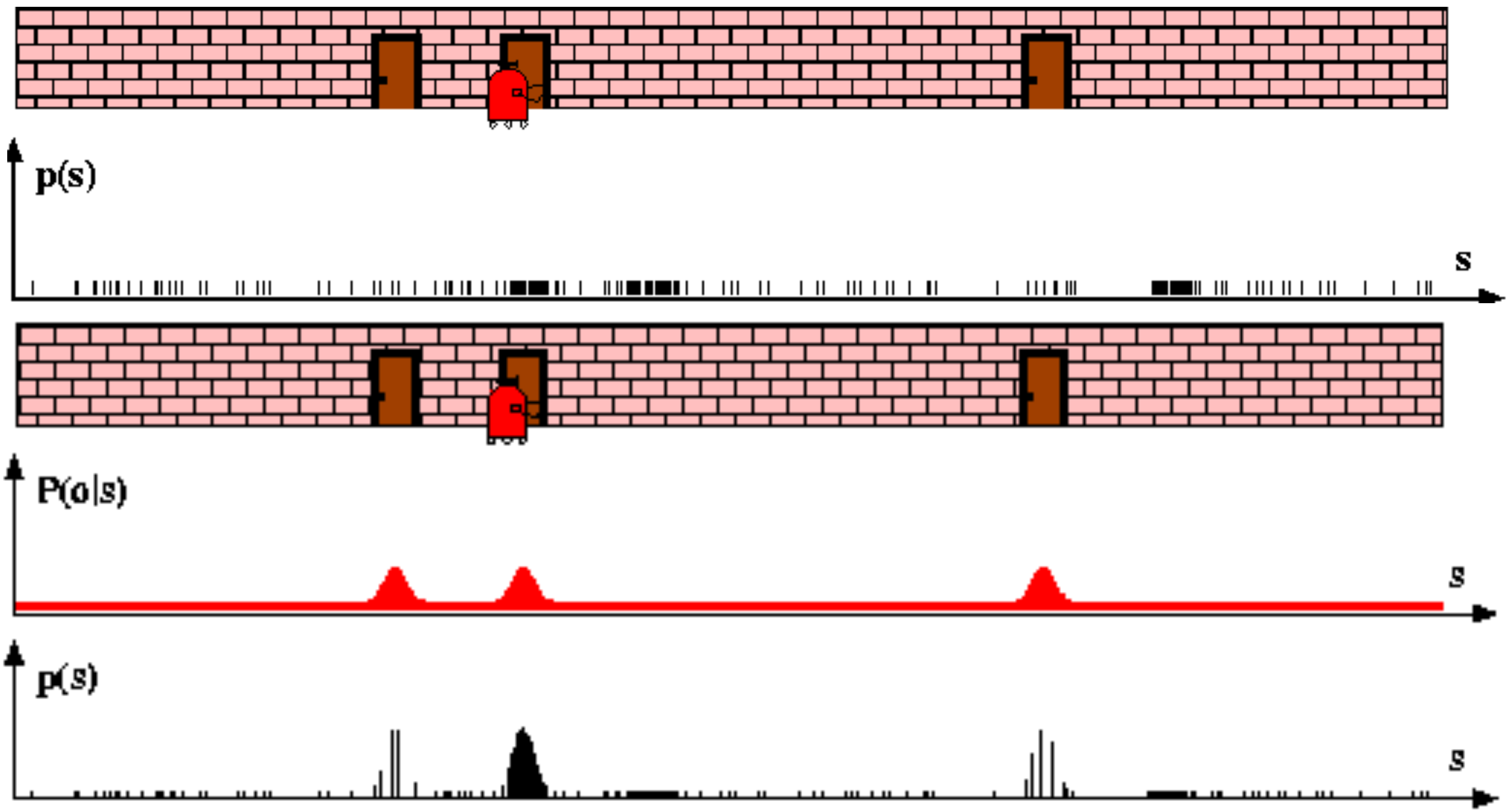
$$Bel^{-}(x) \quad \leftarrow \quad \int p(x \mid u, x')\, Bel(x') \;\mathrm{d}\, x'$$



# Sensor Information: Importance Sampling

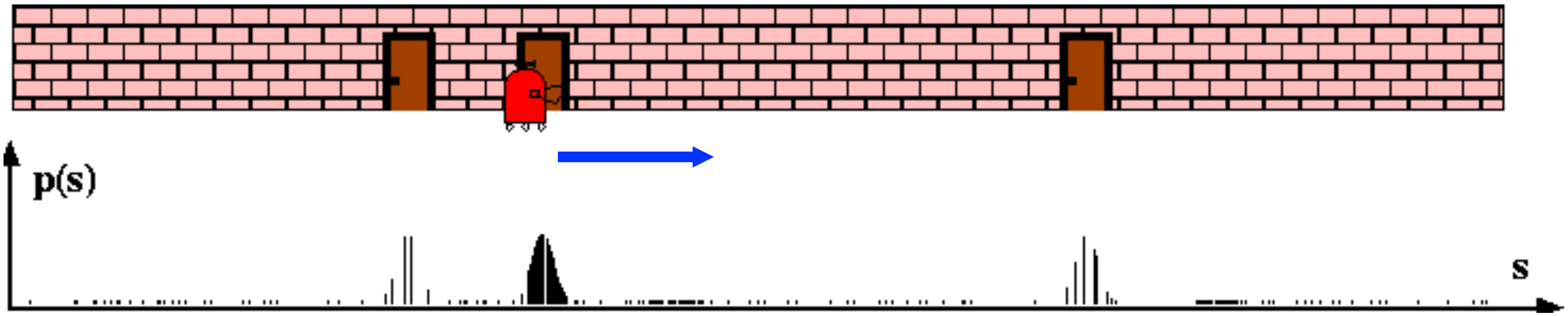
$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

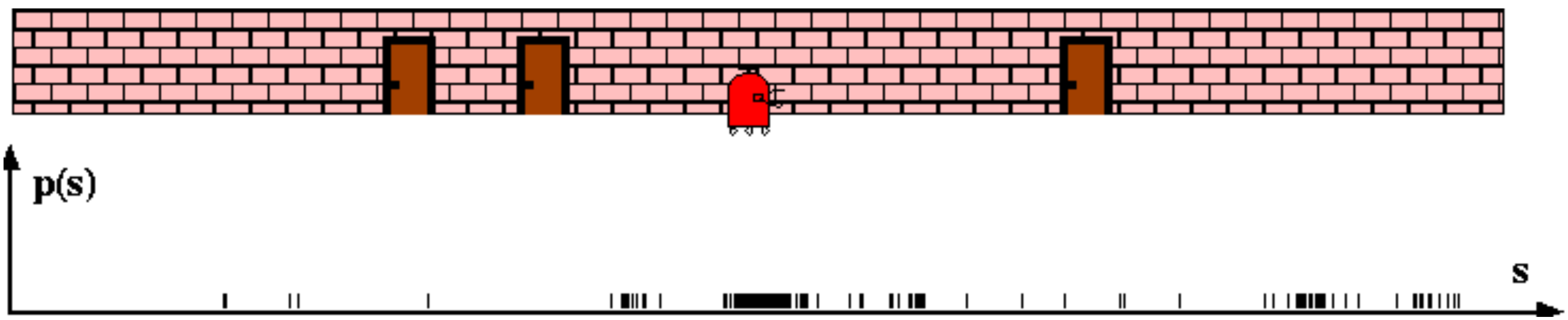
$$w \leftarrow \frac{\alpha\, p(z \mid x)\, Bel^{-}(x)}{Bel^{-}(x)} = \alpha\, p(z \mid x)$$



# Robot Motion

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



# Particle Filter Algorithm

1. Algorithm **particle_filter**( $S_{t-1}$, $u_{t-1}$ $z_t$):
2. $S_t = \emptyset, \quad \eta = 0$
3. **For** $i = 1 \ldots n$ ***Generate new samples***
4. Sample index *j(i)* from the discrete distribution given by $w_{t-1}$
5. Sample $x_t^i$ from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x_{t-1}^{j(i)}$ and $u_{t-1}$
6. $w_t^i = p(z_t \mid x_t^i)$ ***Compute importance weight***
7. $\eta = \eta + w_t^i$ ***Update normalization factor***
8. $S_t = S_t \cup \{< x_t^i, w_t^i >\}$ ***Insert***
9. **For** $i = 1 \ldots n$
10. $w_t^i = w_t^i / \eta$ ***Normalize weights***

# Particle Filter Algorithm

$$Bel(x_t) = \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1})\, dx_{t-1}$$

draw $x^i_{t-1}$ from $Bel(x_{t-1})$

draw $x^i_t$ from $p(x_t \mid x^i_{t-1}, u_{t-1})$

Importance factor for $x^i_t$:

$$\begin{aligned} w^i_t &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{\eta\, p(z_t \mid x_t)\, p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1})}{p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1})} \\ &\propto p(z_t \mid x_t) \end{aligned}$$