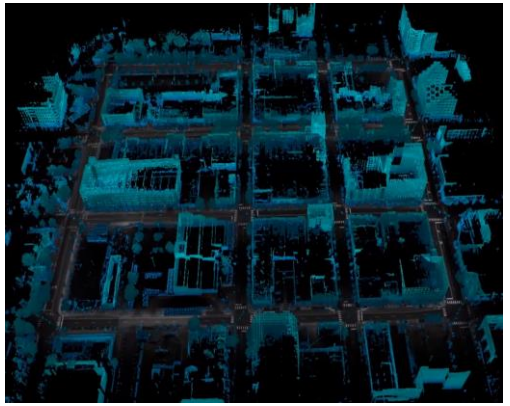
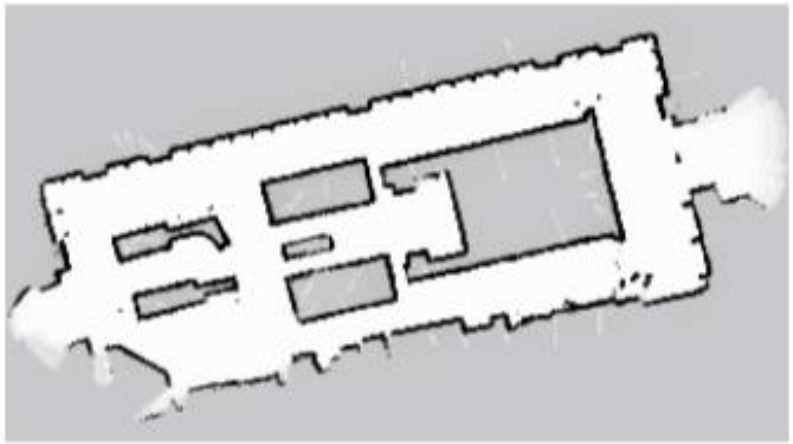
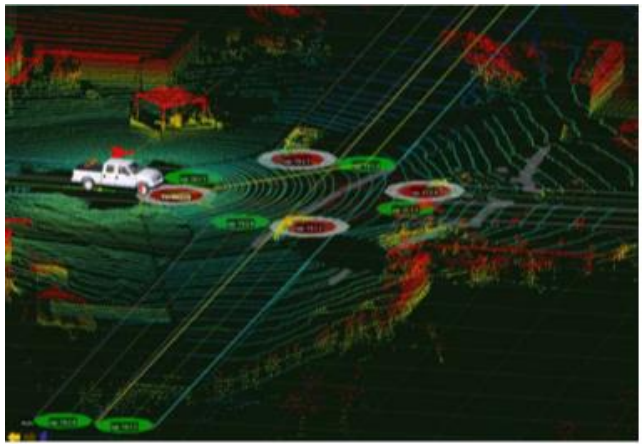
BYU Electrical & Computer Engineering





LOCALIZATION & PARTICLE FILTERS

ECEN 633: Robotic Localization and Mapping

Agenda

- Localization
- Localization Example
- Particle Filters Introduction
- Particle Filtering Derivation
  - Monte Carlo Integration
  - Importance Sampling
  - Sequential Importance Sampling
  - Degeneracy
  - Resampling

Localization

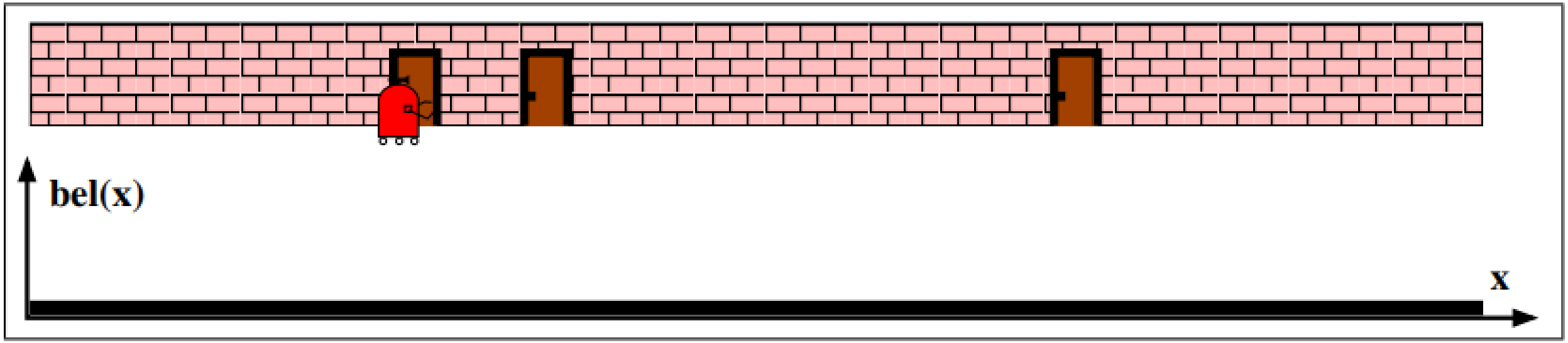
- Given:
  - A Map $\mathbf{m}$
  - Measurements $\mathbf{z}_0, \cdots, \mathbf{z}_T$

- **Find:**
  - The pose of the robot at either:
    - time t : $\mathbf{x}_t$

      or
    - all times from $t = 0, \cdots, T$: $\mathbf{x}_0, \cdots, \mathbf{x}_T$

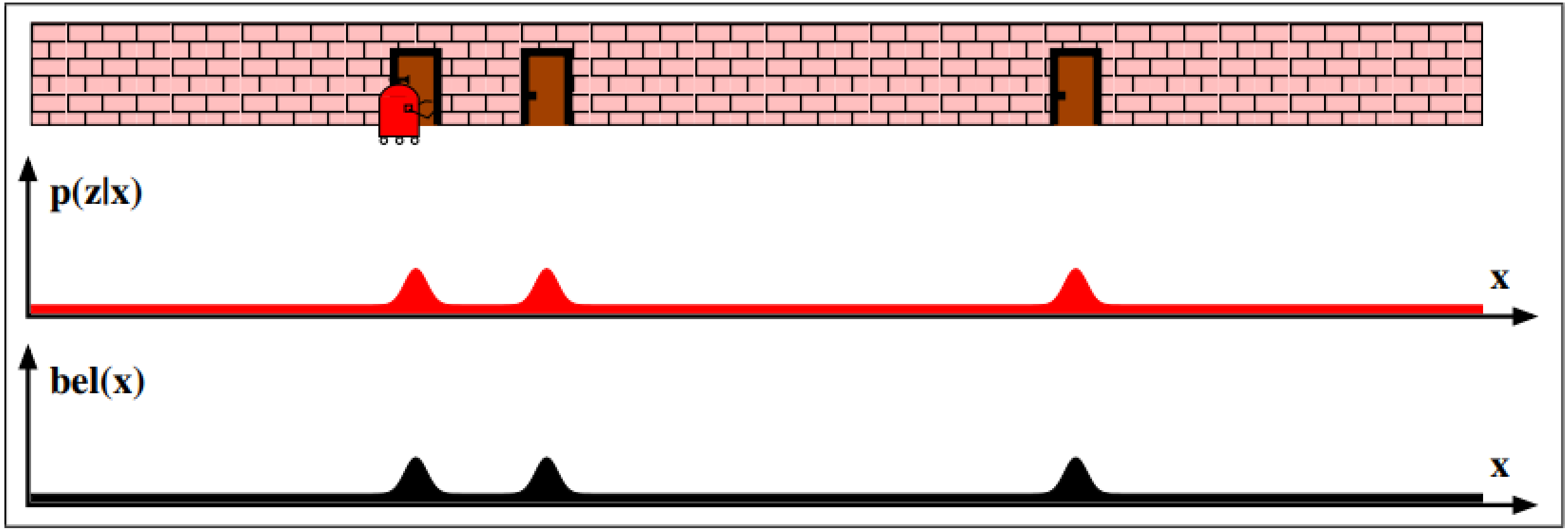
Localization Example

Initial Belief (Prior)



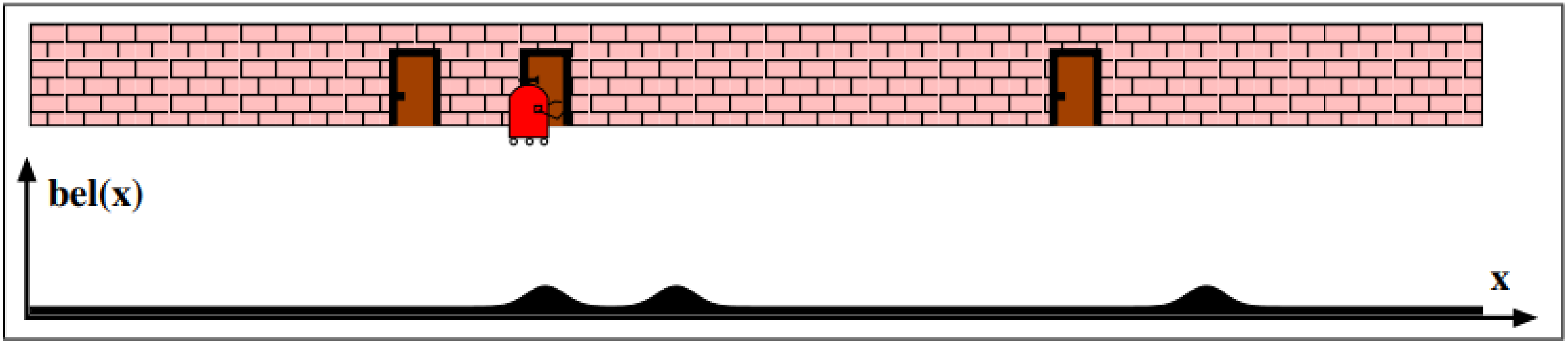
Localization Example

Robot Senses Door



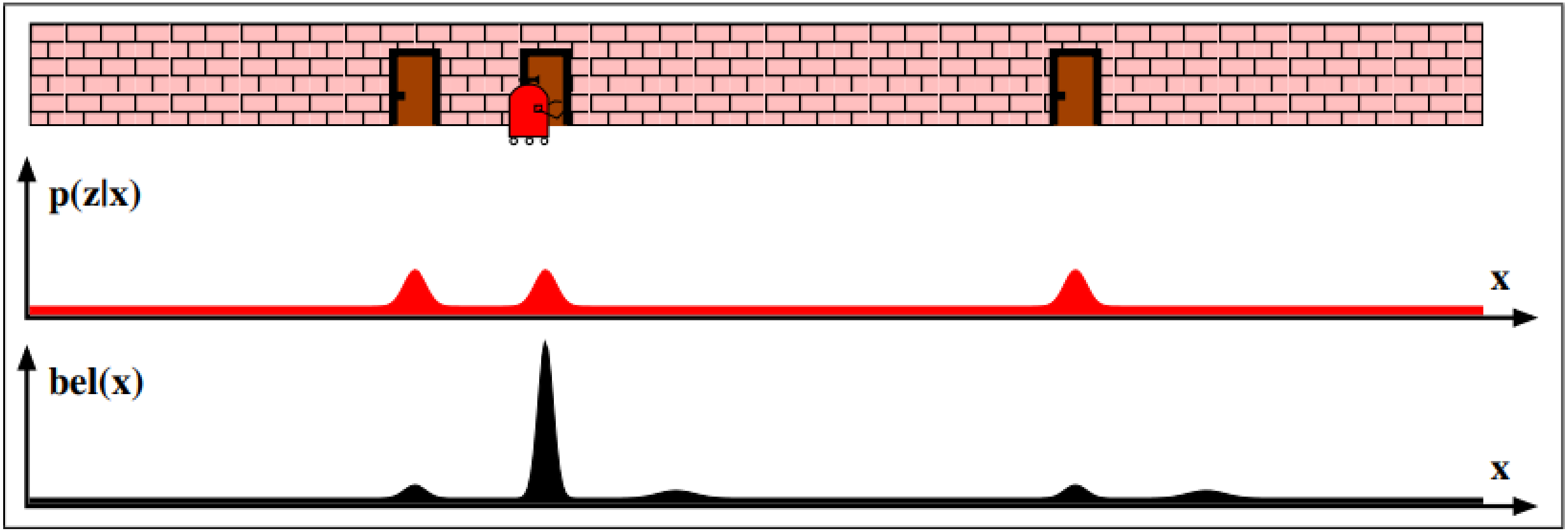
Localization Example

Robot Moves Forward



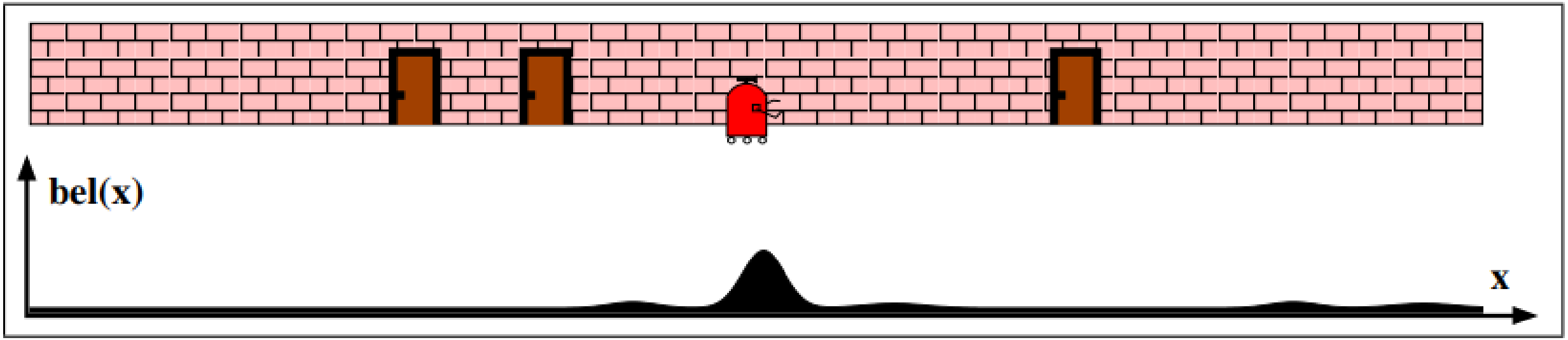
Localization Example

Robot Senses Door Again



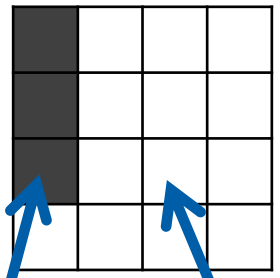
Localization Example

Robot Moves Forward Again



How to Model State/State Uncertainty?

Occupancy Grids

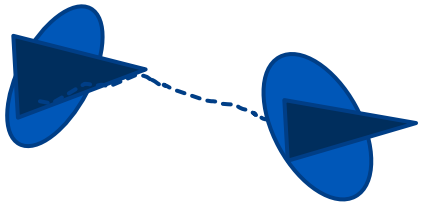


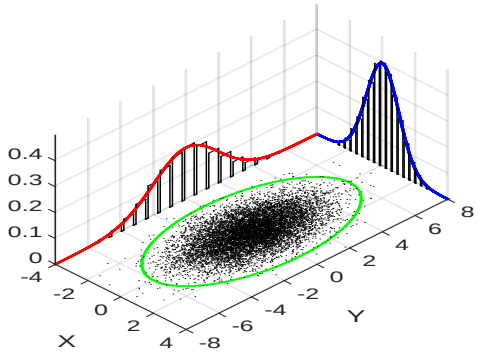
$$p(m_j) \rightarrow 0$$

$$p(m_i) \rightarrow 1$$

Binary RVs

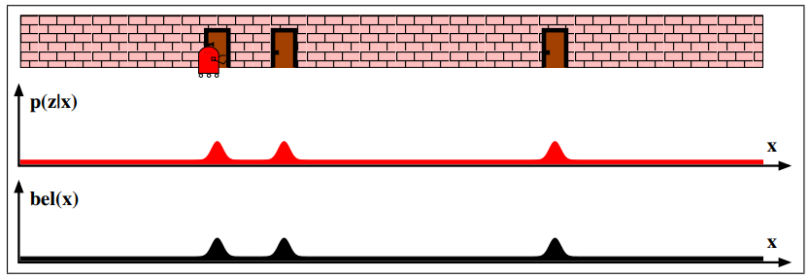
Smith, Self, Cheeseman (Kalman Filtering)

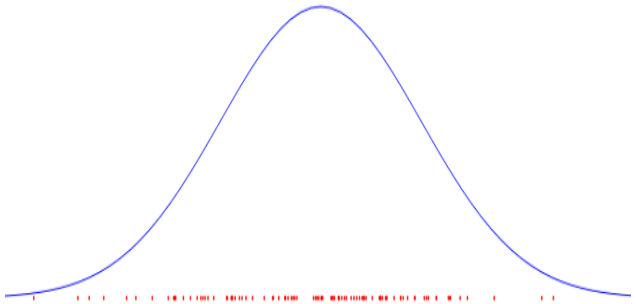




Normally Distributed RVs

Particle Filter





Non-Parametric Particles

State Estimation

Parametric Methods

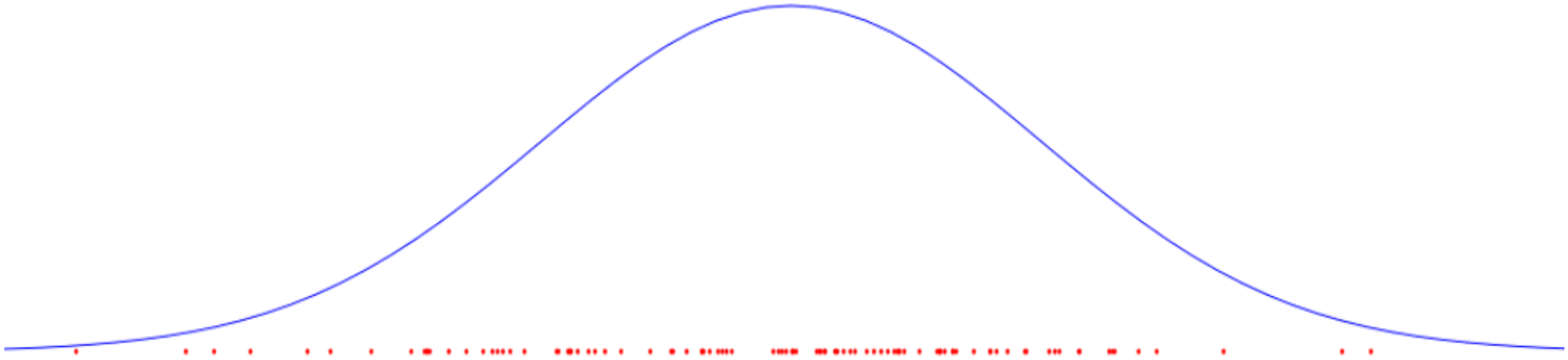
- Model state using a parametric distribution (e.g. normal distribution or binary random variable)
- Can give optimal estimate if assumptions hold
- Examples:
  - Kalman Filter
  - EKF
  - Factor Graph Optimization

Non-parametric Methods

- Do not assume a particular model
- Allows tracking of arbitrary distributions
- Often use particles or kernels to represent underlying distribution

Particle Filters

- Instead of a *parametric* description of state (and uncertainty), use a set of *state samples*. The distribution of these particles represents the posterior distribution.
- Can represent arbitrary PDFs, not just Gaussians.



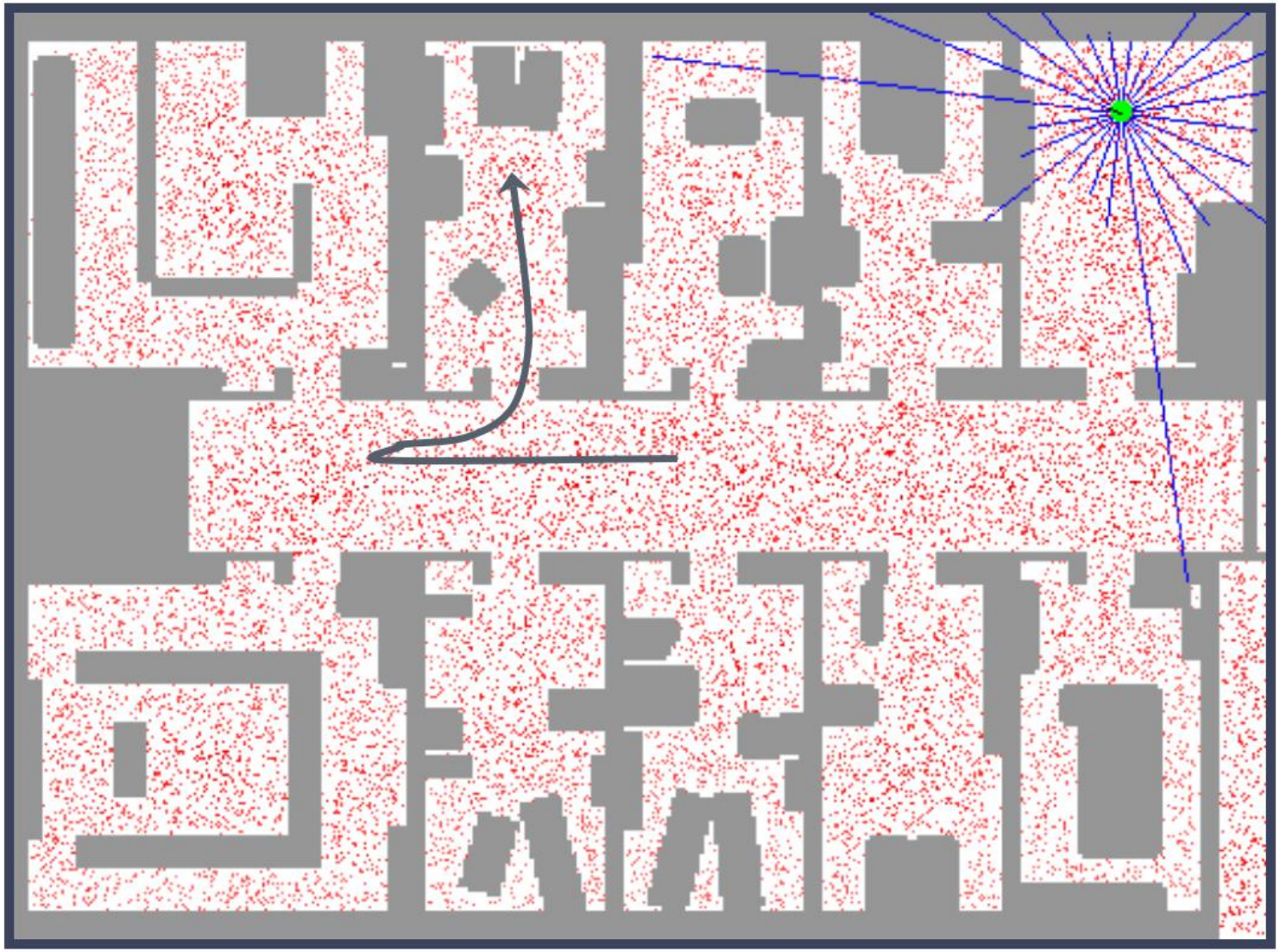
Particle Filters

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

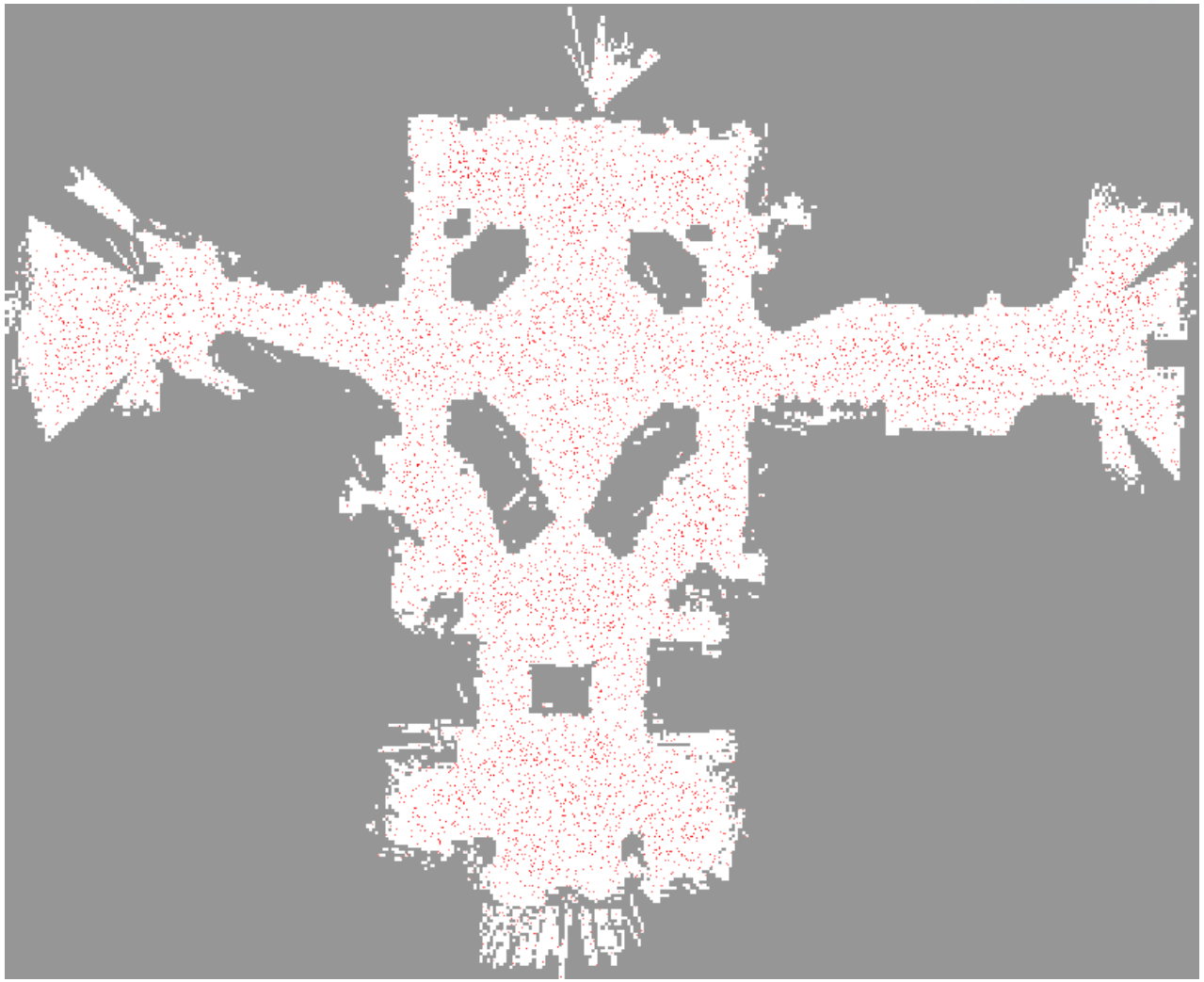
- Sequential Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter, Point-Mass (a.k.a. "particle filter)

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

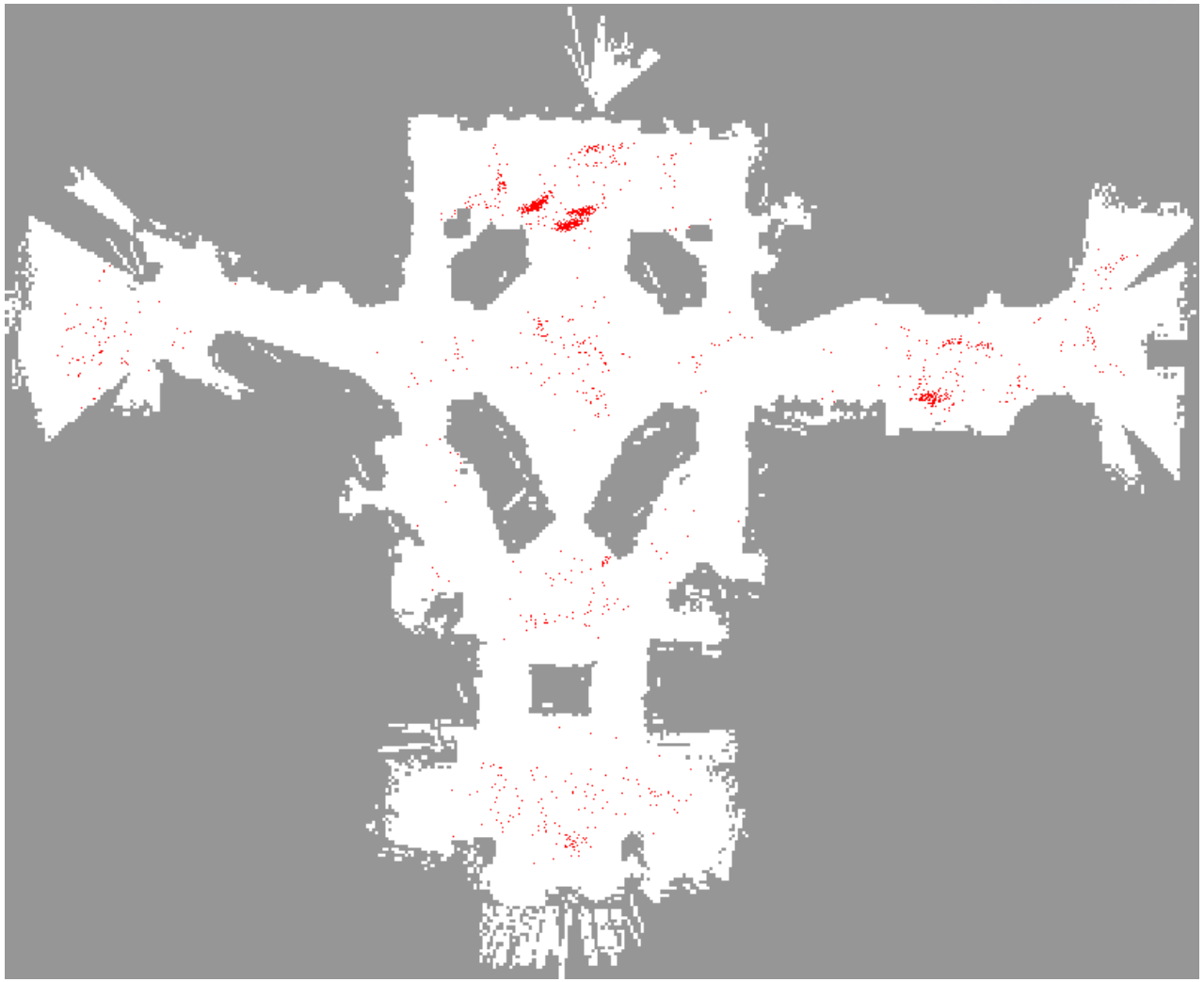
Sample-based Localization (sonar)



Example: Start

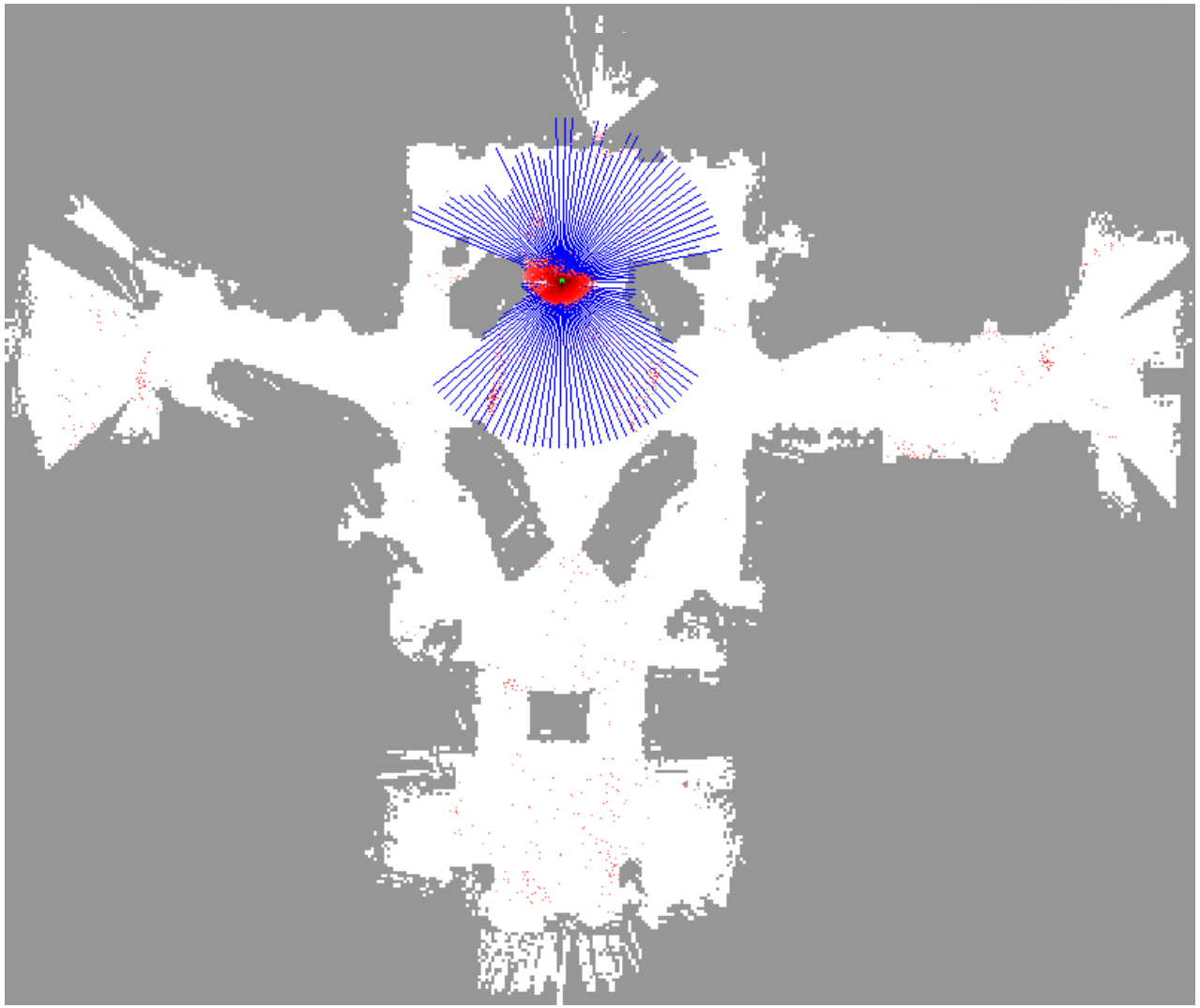


Example: Resample According to Measurements

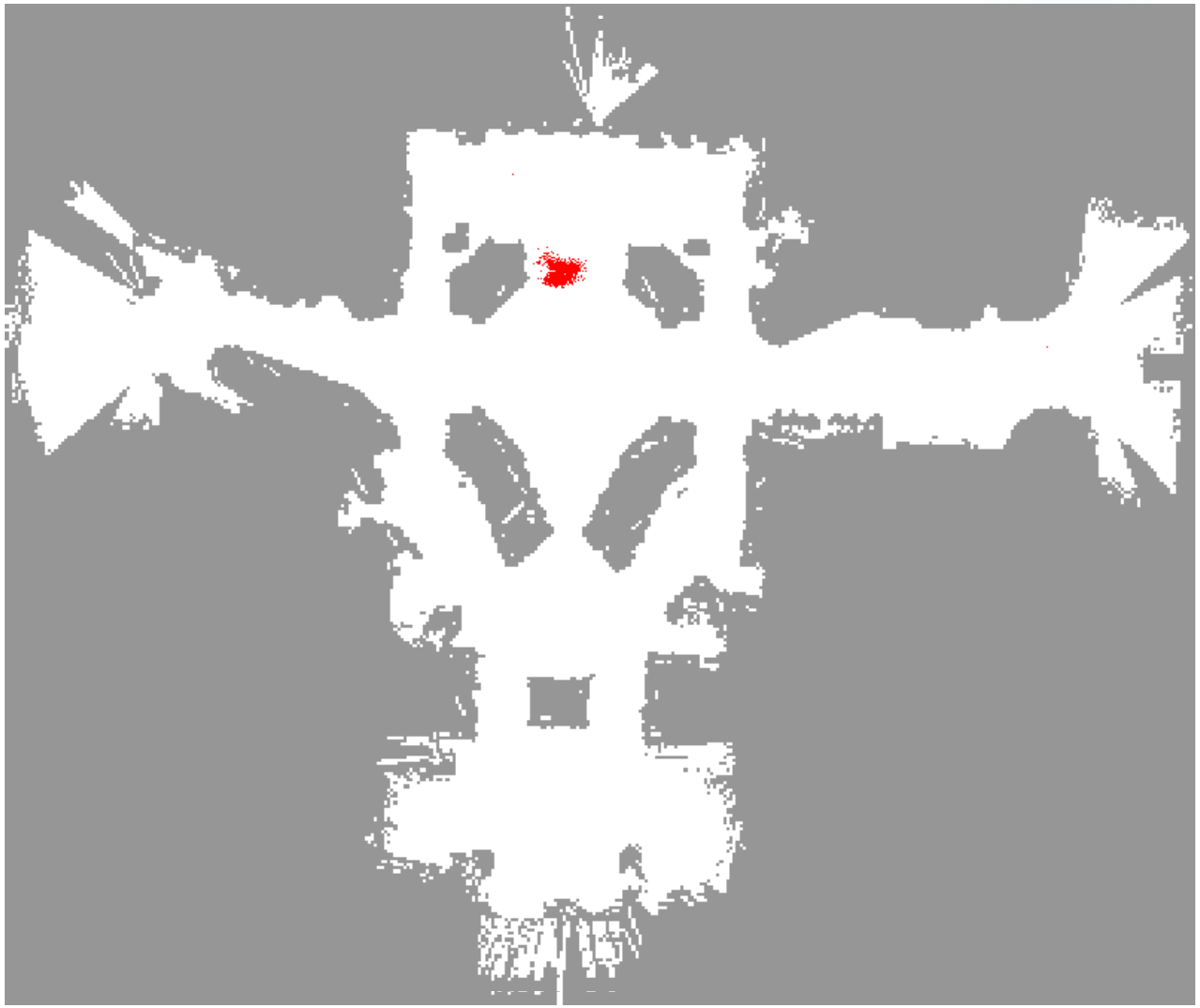


Example: Propagate with Motion Model

Example: Get Measurements



Example: Resample



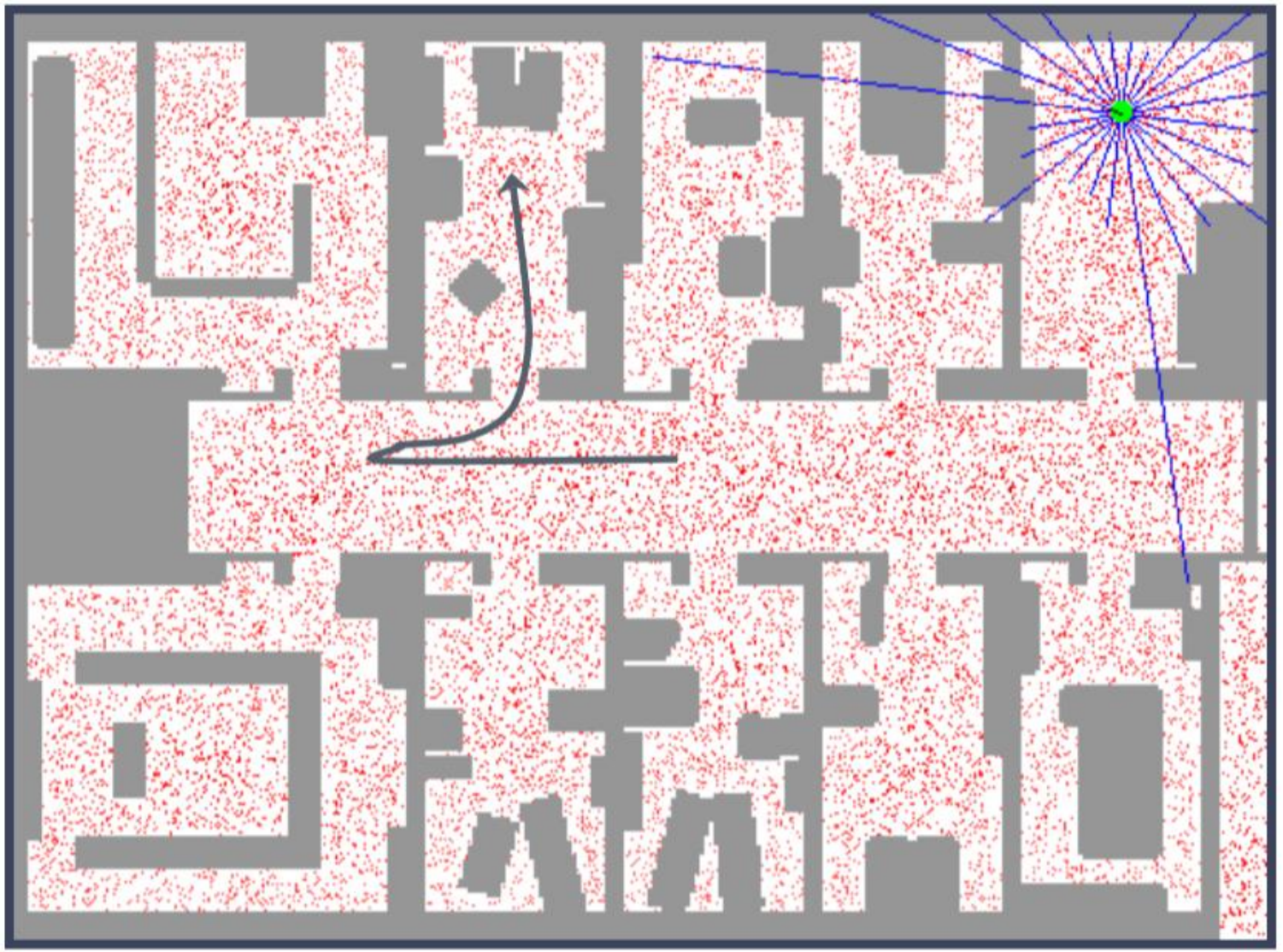
Sequential Monte Carlo (SMC): Brief History

- Basic Idea of SMC around since the 1950s
- Explored through 60's and 70's, but largely overlooked and ignored:
  - Modest computational power at the time
  - Vanilla Sequential Importance Sampling (SIS) leads to degeneracy over time
- The major contribution to development of the SMC method was the inclusion of the resampling step [Neil Gordon et al, 1993]

Particle Filter Derivation

- Monte Carlo Integration
- Importance Sampling
- Sequential Importance Sampling

Particle Filters - Sequential Monte Carlo (SMC)



Roots of SMC are in MC Integration

- Let I be the result of a multivariate integral, where g is some arbitrary function.

$$I = \int g(\mathbf{x}) d\mathbf{x}$$

- Thought experiment
  - Imagine discretizing and evaluating I numerically. What is the complexity?

MC Integration

- Suppose we can factorize $g(\mathbf{x})$

$$g(\mathbf{x}) = f(\mathbf{x})\pi(\mathbf{x})$$

- Such that $\pi(\mathbf{x})$ can be interpreted as a pdf

$$\pi(\mathbf{x}) \geq 0 \quad \text{and} \quad \int \pi(\mathbf{x}) d\mathbf{x} = 1$$

- Draw N>>1 i.i.d samples from $\pi(\mathbf{x})$, then

$$I = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} = E_\pi[f(\mathbf{x})] \approx I_N = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i)$$

Why 1/N?

Because samples already distributed according to $\pi(\mathbf{x})$

MC Integration Continued …

- If the samples $\mathbf{x}^i$ are i.i.d., then I_N is the unbiased estimate of the integral I.
- According to the law of large numbers I_N will almost surely converge to I.
- If the variance of $f(\mathbf{x})$ is finite, i.e.

$$\sigma^2 = \int (f(\mathbf{x}) - I)^2 \pi(\mathbf{x}) d\mathbf{x}$$

then the CLT holds and the estimation error converges in distribution:

$$\lim_{N \to \infty} \sqrt{N}(I_N - I) \sim \mathcal{N}(0, \sigma^2)$$

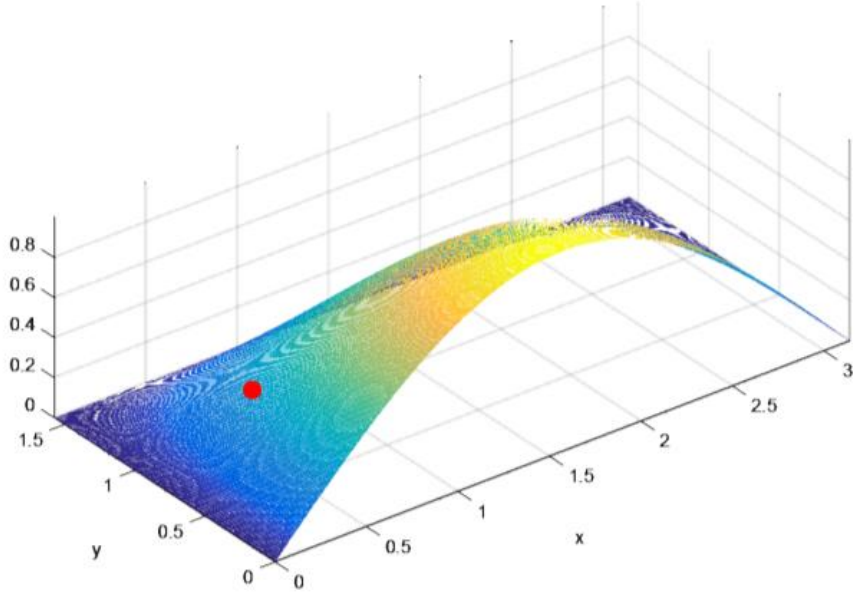
MC Integration …

- Punch Line
  - The error $e=I_N-I$ is on the order of $O(N^{-½})$
  - The rate of convergence is **independent** of the **dimension** of the integrand n_x !

Example

Integrate $g(\mathbf{x}) = \sin(x)\cos(y)$ over domain $0<x<\pi$ and $0<y<\pi/2$

$$I = \int_0^{\pi/2} \int_0^{\pi} \sin(x)\cos(y)dxdy = -\cos(x)|_0^{\pi} \sin(y)|_0^{\pi/2} = 2$$



Example continued

- Equivalent MC Integral

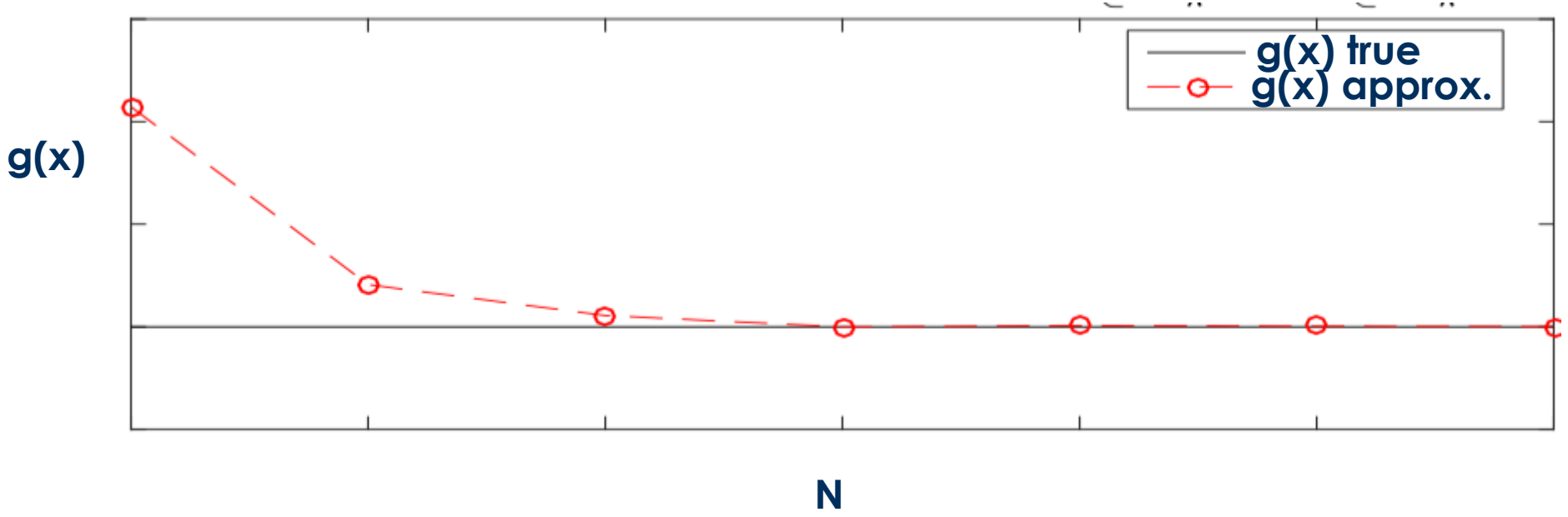
$$g(\mathbf{x}) \equiv f(\mathbf{x})\pi(\mathbf{x}) = \underbrace{\left(\frac{\pi^2}{2}\sin(x)\cos(y)\right)}_{f(\mathbf{x})}\underbrace{\left(\frac{1}{\frac{\pi^2}{2}}\right)}_{\pi(\mathbf{x})}$$

- Draw $\mathbf{x}^i$ i.i.d. samples from $\pi(\mathbf{x})$ and compute $f(\mathbf{x}^i)$, compute mean of $f(\mathbf{x})$ for N=10, 100, 1000, 10000, ..., 10^7

- Expect std of error $\text{std}(e) = \frac{\sqrt{2.0881}}{\sqrt{N}}$

$$\sigma^2 = \int \left(\frac{\pi^2}{2}\sin(x)\cos(y) - 2\right)^2 \pi(\mathbf{x})d\mathbf{x} = \frac{\pi^4}{16} - 4 \approx 2.0881$$

It works!



Importance Sampling

- Ideally we want to sample from $\pi(\mathbf{x})$ and estimate I. But suppose we can only *generate* samples from density $q(\mathbf{x})$, which is similar to $\pi(\mathbf{x})$

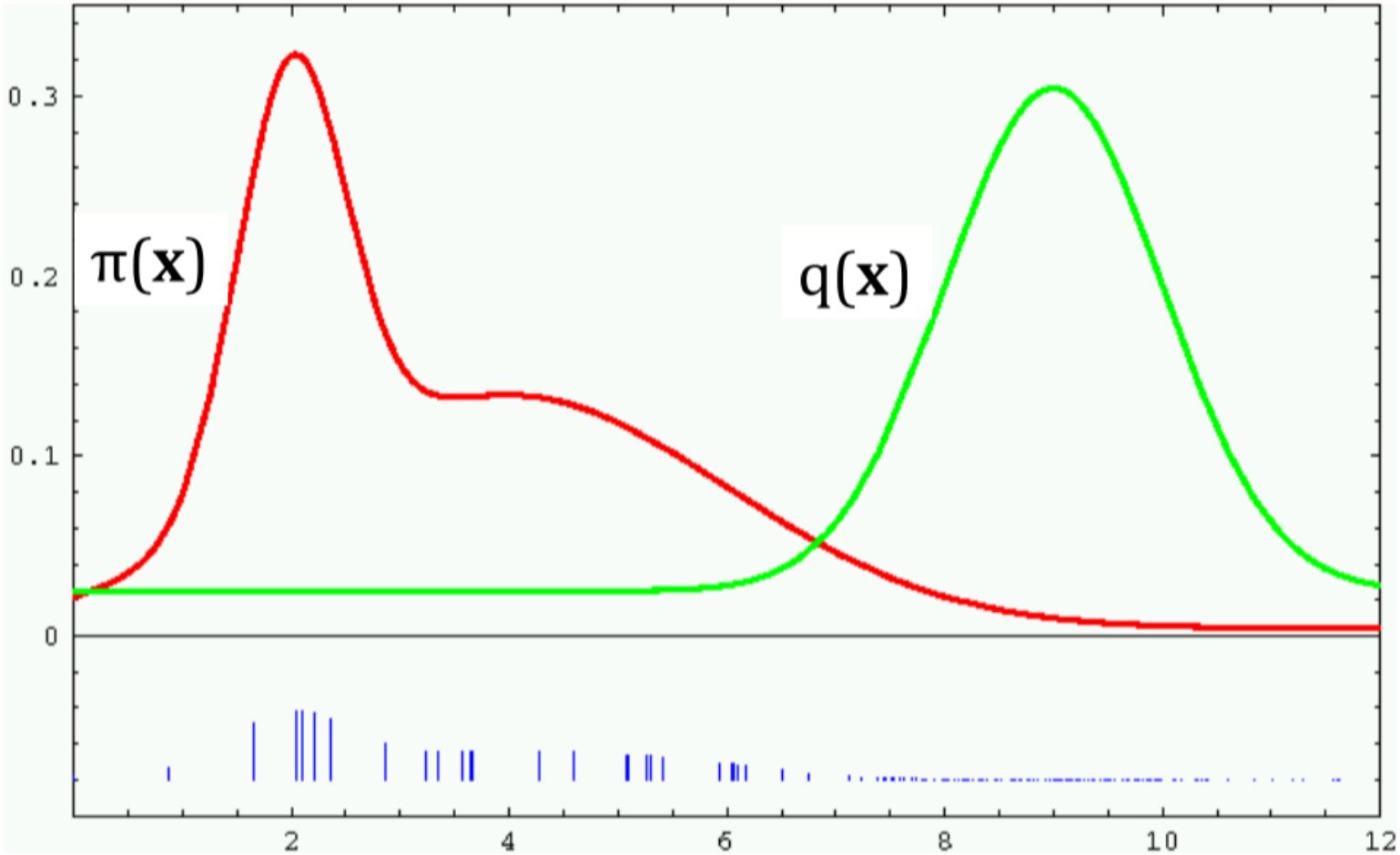
$$\underbrace{\pi(\mathbf{x}) > 0 \Rightarrow q(\mathbf{x}) > 0}_{\text{i.e., same region of support}} \quad \forall \quad \mathbf{x} \in \Re^{n_x}$$

- $q(\mathbf{x})$ is call the importance or proposal density

$$I = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\underbrace{\frac{\pi(\mathbf{x})}{q(\mathbf{x})}}_{\text{Importance weight}}q(\mathbf{x})d\mathbf{x}$$

$$\tilde{w}(\mathbf{x}) = \frac{\text{target dist}}{\text{proposal dist}}$$

Importance Sampling



Weight samples: $\tilde{w}(\mathbf{x}) = \dfrac{\text{target dist}}{\text{proposal dist}} = \dfrac{\pi(\mathbf{x})}{q(\mathbf{x})}$

MC Integration with Importance Sampling

- Draw $N>>1$ i.i.d. samples $\mathbf{x}^i \sim q(\mathbf{x})$

- Compute $$I_N = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i)\tilde{w}(\mathbf{x}^i)$$

- If desired density $\pi(\mathbf{x})$ is known only up to a proportionality constant then what?

$$I_N = \frac{\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i)\tilde{w}(\mathbf{x}^i)}{\frac{1}{N}\sum_{j=1}^{N} \tilde{w}(\mathbf{x}^j)} = \sum_{i=1}^{N} f(\mathbf{x}^i)w(\mathbf{x}^i)$$

Normalized Importance weight → $$w(\mathbf{x}^i) = \frac{\tilde{w}(\mathbf{x}^i)}{\sum_{j=1}^{N}\tilde{w}(\mathbf{w}^i)}$$

Bayesian Inference

- MC can be applied in a Bayesian framework where $\pi(\mathbf{x})$ is the sought after posterior density.

- To develop, let us introduce:
  - The sequence of all target states up to time k

$$\underbrace{\mathbf{X}_k}_{\text{PF Tutorial (Gordon et al)}} = \{\mathbf{x}_j; j = 0, \ldots, k\} \equiv \underbrace{\mathbf{x}_{0:k}}_{\text{ProbRob}}$$

  - The sequence of all measurements up to time k

$$\mathbf{Z}_k = \{\mathbf{z}_j; j = 1, \ldots, k\} \equiv \mathbf{z}_{1:k}$$

Posterior over State Trajectory

- Joint-posterior density at time k: $p(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k})$

- Discrete approximation:

$$p(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i)$$

Dirac delta fcn

- Normalized weights, w_k^i, chosen using principle of importance sampling.

$$\mathbf{x}_{0:k}^i \sim q(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k}) \qquad w_k^i \propto \frac{p(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k})} \qquad \sum_{i=1}^{N} w_k^i = 1$$

Recursive Factorization

- Sampling
  - Suppose at time step k-1 we have samples approximating
    $p(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1})$
  - With reception of $\mathbf{z}_k$ at time k, we wish to approximate
    $p(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k})$
  - If importance density is *chosen* to factorize as
    $q(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k}) \equiv q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})\, q(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1})$
  - Samples $\mathbf{x}^i_{0:k} \sim q(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k})$ can be obtained by augmenting existing samples $\mathbf{x}^i_{0:k-1} \sim q(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1})$ with the new state $\mathbf{x}^i_k \sim q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})$

Recursive Factorization

- Weight Update
  - Target distribution

$$
\begin{aligned}
p(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k}) &= \frac{p(\mathbf{z}_k \mid \mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})\, p(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1})\, p(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})} \\
&= \frac{p(\mathbf{z}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, p(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})} \\
&\propto p(\mathbf{z}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, p(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1})
\end{aligned}
$$

  - Hence, importance weights become:

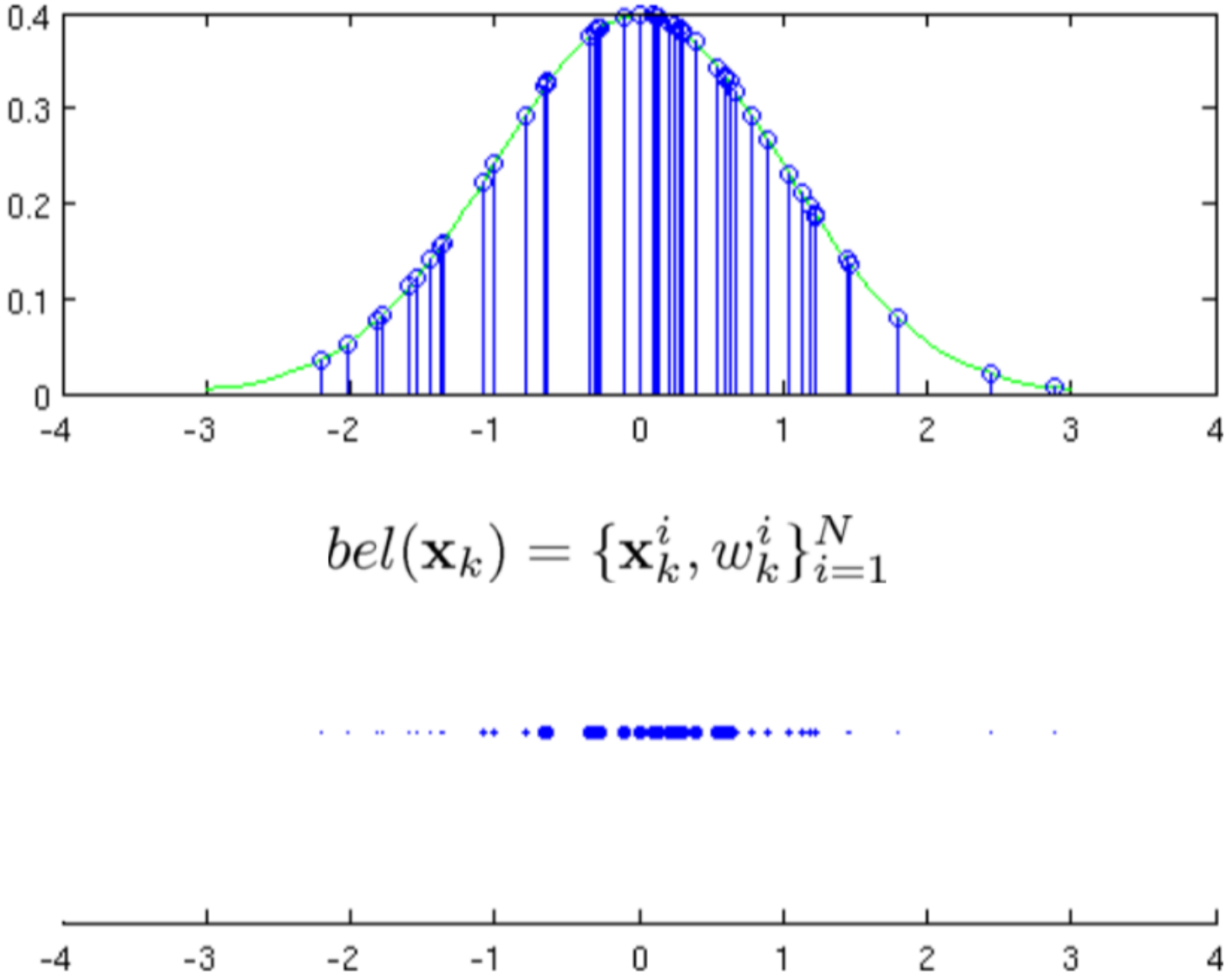
$$
w_k^i \propto \frac{p(\mathbf{x}_{0:k}^i \mid \mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}^i \mid \mathbf{z}_{1:k})} \propto \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^i)\, p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i \mid \mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k})} w_{k-1}^i
$$

Filtering Posterior

- If $q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) = q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_k)$ then importance density only depends on $\mathbf{x}_{k-1}$, and $\mathbf{z}_k$
  - i.e., Markov
- Hence $w_k^i \propto \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^i)\, p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k)} w_{k-1}^i$
- In such scenarios, only need to store $\mathbf{x}_k{}^i$'s
  - Can silently "ignore" sample sequence of state trajectory, $\mathbf{x}_{0:k-1}{}^i$, and history of observations, $\mathbf{z}_{1:k-1}{}^i$
- Filtered posterior approximation becomes

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i)$$

SIS Belief



$$bel(\mathbf{x}_k) = \{\mathbf{x}_k^i, w_k^i\}_{i=1}^N$$

Sequential Importance Sampling (SIS) Algorithm

Table 3.1

Filtering via SIS

$[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N] = \text{SIS}\,[\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^N, \mathbf{z}_k]$

- FOR $i = 1 : N$
  - Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$
  - Evaluate the importance weights up to a normalizing constant according to (3.16)

$$\tilde{w}_k^i = w_{k-1}^i \frac{p\left(\mathbf{z}_k|\mathbf{x}_k^i\right) p\left(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i\right)}{q\left(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k\right)}$$

- END FOR
- Calculate total weight: $t = \text{SUM}\,[\{\tilde{w}_k^i\}_{i=1}^N]$
- FOR $i = 1 : N$
  - Normalize: $w_k^i = t^{-1}\tilde{w}_k^i$
- END FOR

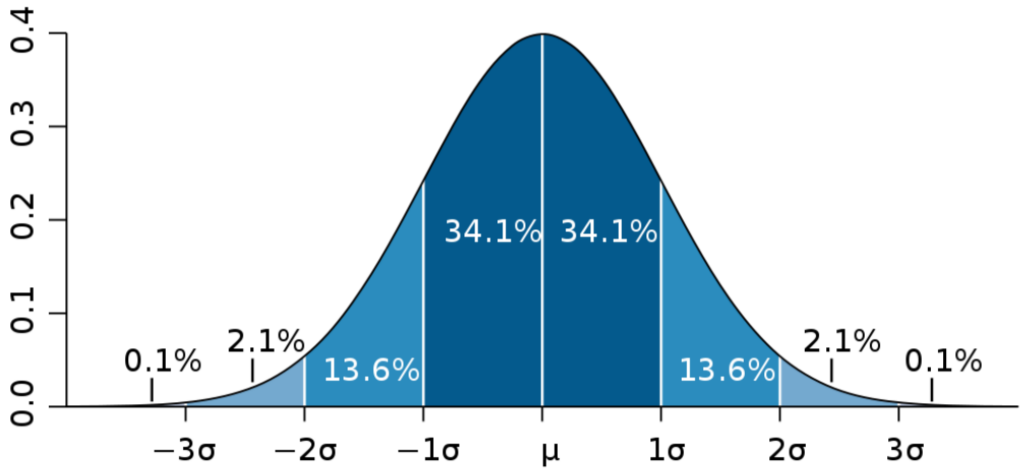
Particle Filtering Continued

- Degeneracy
- Resampling

Gaussian Filters

- The Kalman filter and its variants can only model (unimodal) **Gaussian distributions**

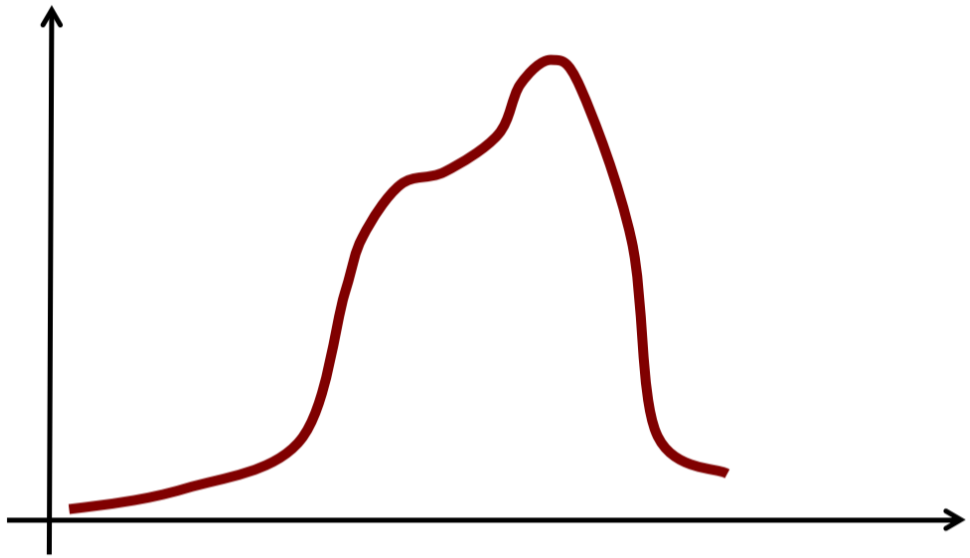
$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu) \right)$$



Courtesy: K. Arras

Motivation

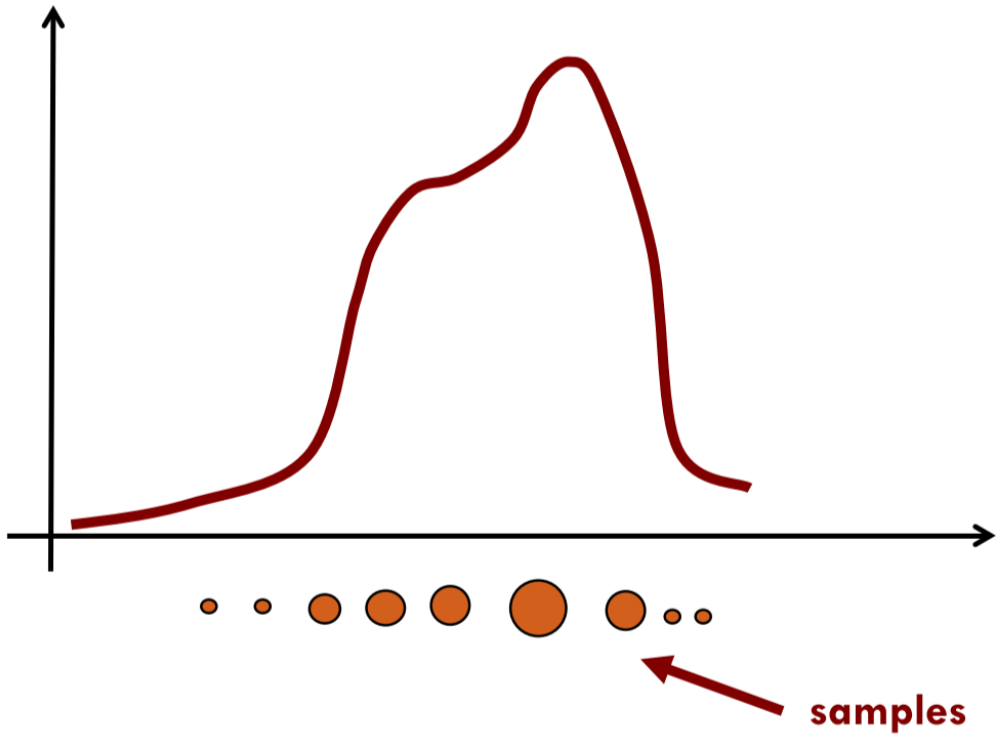
- Goal: approach for dealing with **arbitrary distributions**



Courtesy: Cyrill Stachniss

Key Idea: Samples

- Use **multiple samples** to represent arbitrary distributions



Courtesy: Cyrill Stachniss

Particle Set

- Set of weighted samples

$$\mathcal{X} = \left\{ \left\langle x^{[j]}, w^{[j]} \right\rangle \right\}_{j=1,\ldots,J}$$

state hypothesis

importance weight

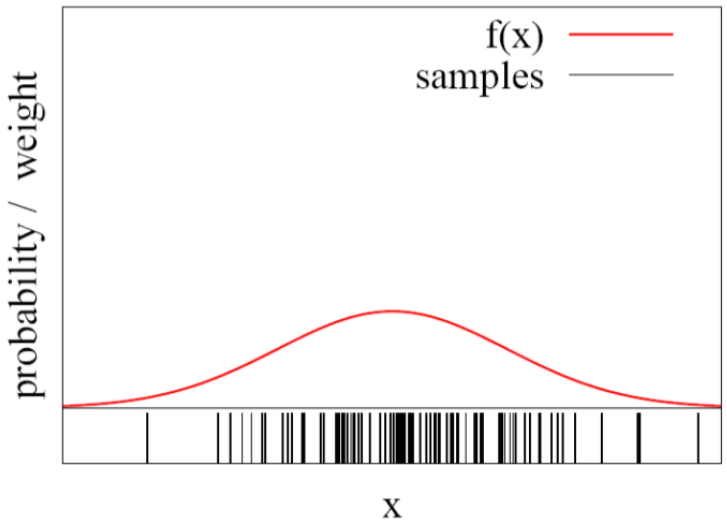
- The samples represent the posterior

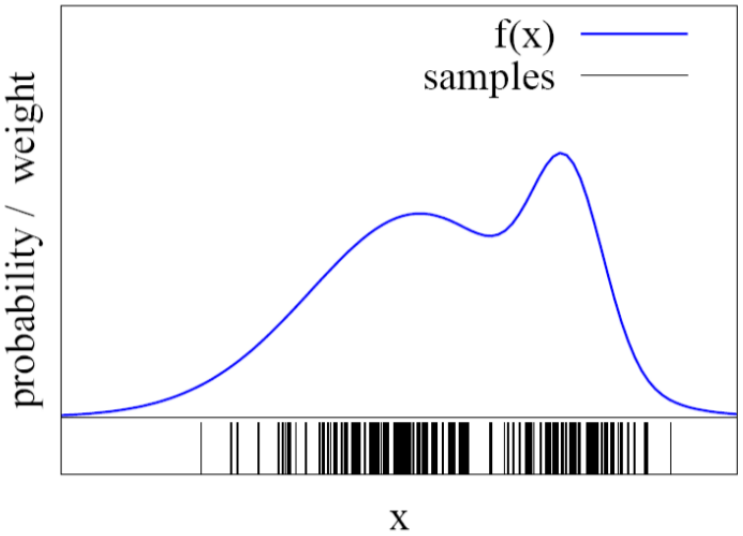
$$p(x) = \sum_{j=1}^{J} w^{[j]} \, \delta_{x^{[j]}}(x)$$

Courtesy: Cyrill Stachniss

Particles for Approximation

- Particles for function approximation





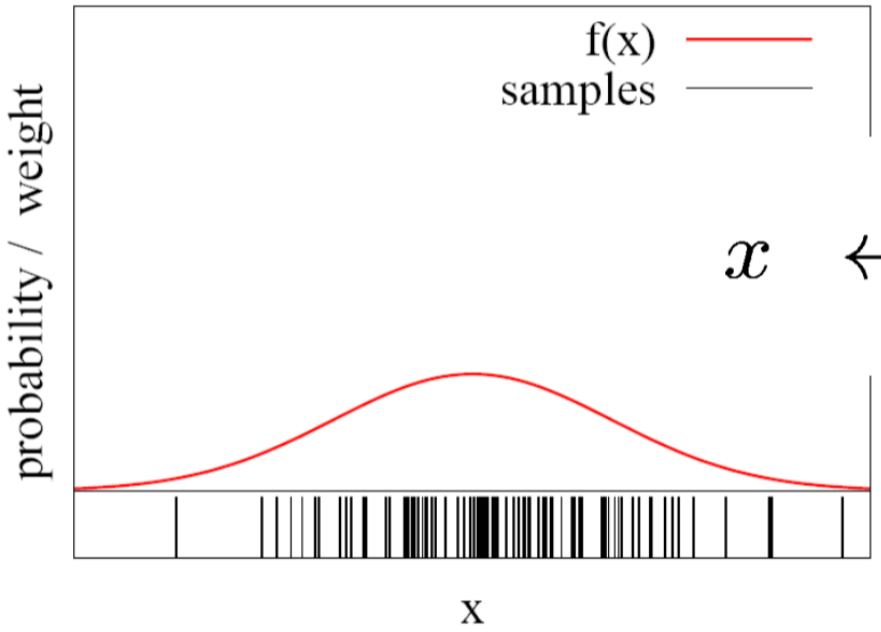
- The more particles fall into a region, the higher the probability of the region

How to obtain such samples?

Courtesy: Cyrill Stachniss

Closed Form Sampling is Only Possible for a Few Distributions

- Example: Gaussian



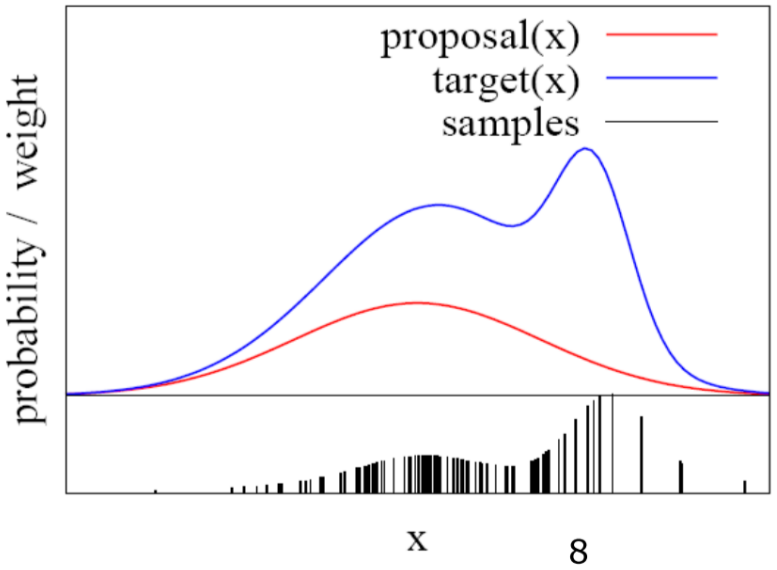
$$x \leftarrow \frac{1}{2}\sum_{i=1}^{12} \text{rand}(-\sigma, \sigma)$$

How to sample from **other** distributions?

Courtesy: Cyrill Stachniss

Importance Sampling Principle

- We can use a different distribution q to generate samples from p
- Account for the "differences between q and p" using a weight $w = p / q$
- target p
- proposal q
- Pre-condition:
  $p(\mathbf{x})>0 \rightarrow q(\mathbf{x})>0$



Courtesy: Cyrill Stachniss

Sequential Importance Sampling (SIS) Algorithm

Table 3.1

Filtering via SIS

$[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N] = \text{SIS}\,[\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^N, \mathbf{z}_k]$

- FOR $i = 1 : N$
  - Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$
  - Evaluate the importance weights up to a normalizing constant according to (3.16)

$$\tilde{w}_k^i = w_{k-1}^i \frac{p\left(\mathbf{z}_k|\mathbf{x}_k^i\right) p\left(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i\right)}{q\left(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k\right)}$$

- END FOR
- Calculate total weight: $t = \text{SUM}\,[\{\tilde{w}_k^i\}_{i=1}^N]$
- FOR $i = 1 : N$
  - Normalize: $w_k^i = t^{-1}\tilde{w}_k^i$
- END FOR

Particle Filter

- Recursive Bayes filter
- Non-parametric approach
- Models the distribution by samples
- Prediction: draw from the proposal
- Correction: weighting by the ratio of target and proposal

The more samples we use, the better the estimate!

Courtesy: Cyrill Stachniss

Degeneracy Problem

- Ideally, the importance density $q(\cdot)$ should be the posterior itself, i.e.,

$$q(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k}) = p(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k})$$

- For the assumed factored form below, it has been shown that the variance of the importance weights can only increase over time†.

$$q(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k}) \equiv q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_k)\, q(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1})$$

- In practical terms, all but one particle will eventually have negligible weight after a fixed number of time steps
  - Effectively, a large computational effort is devoted to updating particles whose contribution to the approximation $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ is almost zero…

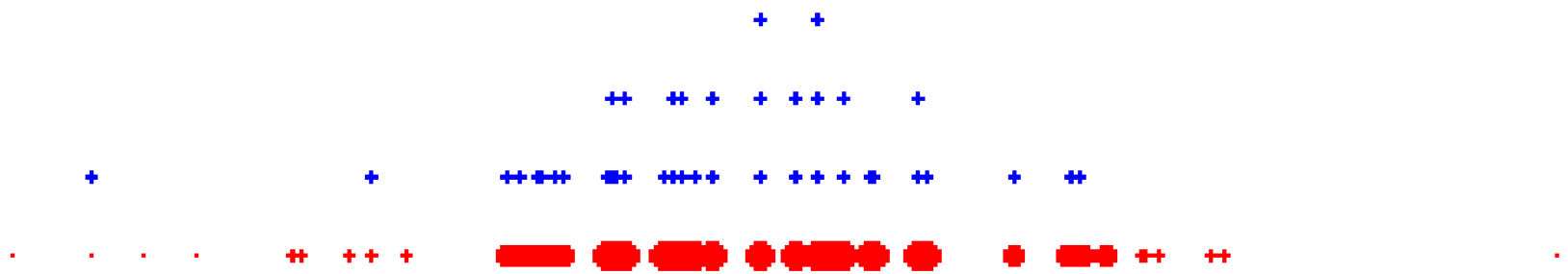
† A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197-208, 2000.

Resampling

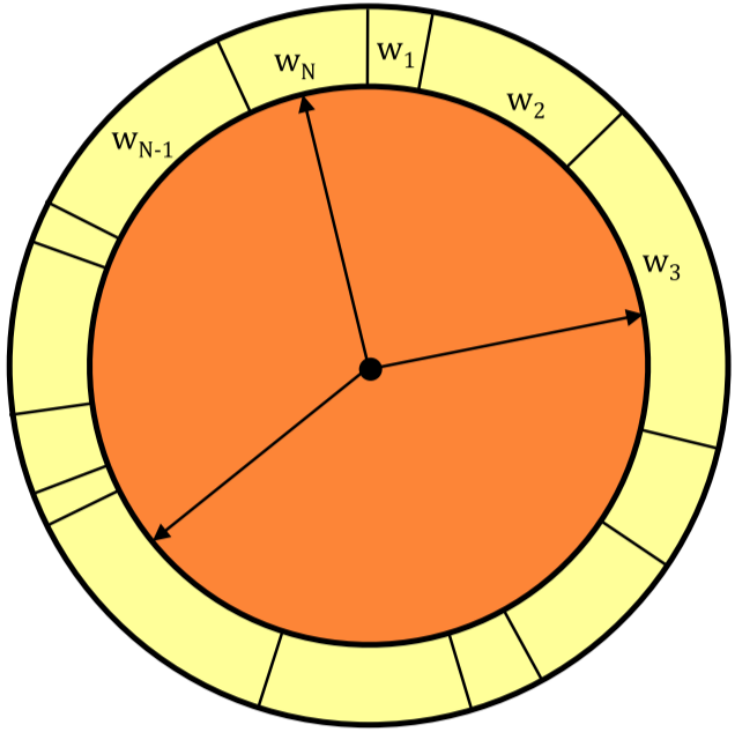
- Resampling eliminates particles with low weights and multiplies particles with high weights

- Maps random measure $\{\mathbf{x}_k^i, w_k^i\}$ to $\{\mathbf{x}_k^{i*}, \underbrace{1/N}_{\text{uniform}}\}$

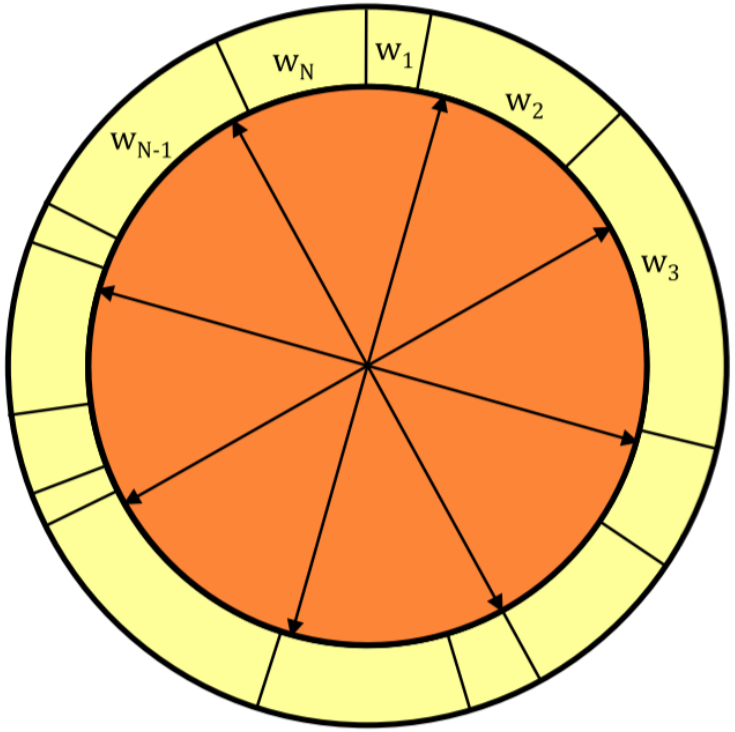
- Sample with replacement such that $\mathrm{P}(\mathbf{x}_k^{i*} = \mathbf{x}_k^i) = w_k^i$



Resampling



- Roulette wheel
- Binary search, $O(N \log N)$



- Stochastic universal sampling
- Systematic resampling
- Linear time complexity, $O(N)$
- Easy to implement, low variance

Low-Variance Resampling Algorithm

Algorithm **systematic_resampling**(S, N):

1. $S' = \emptyset, c_1 = w^1$
2. **For** $i = 2 \ldots N$ — *Generate CDF*
3. $c_i = c_{i-1} + w^i$

4. $u_1 \sim U[0, N^{-1}], i = 1$ — *Initialize threshold*
5. **For** $j = 1 \ldots N$ — *Draw samples …*
6. **While** ($u_j > c_i$) — *Skip until next threshold reached*
7. $i = i + 1$
8. $S' = S' \cup \{< x^i, N^{-1} >\}$ — *Insert*
9. $u_{j+1} = u_j + N^{-1}$ — *Increment threshold*

10. **Return** S'

Also called **stochastic universal sampling**

Idea behind low variance sampling

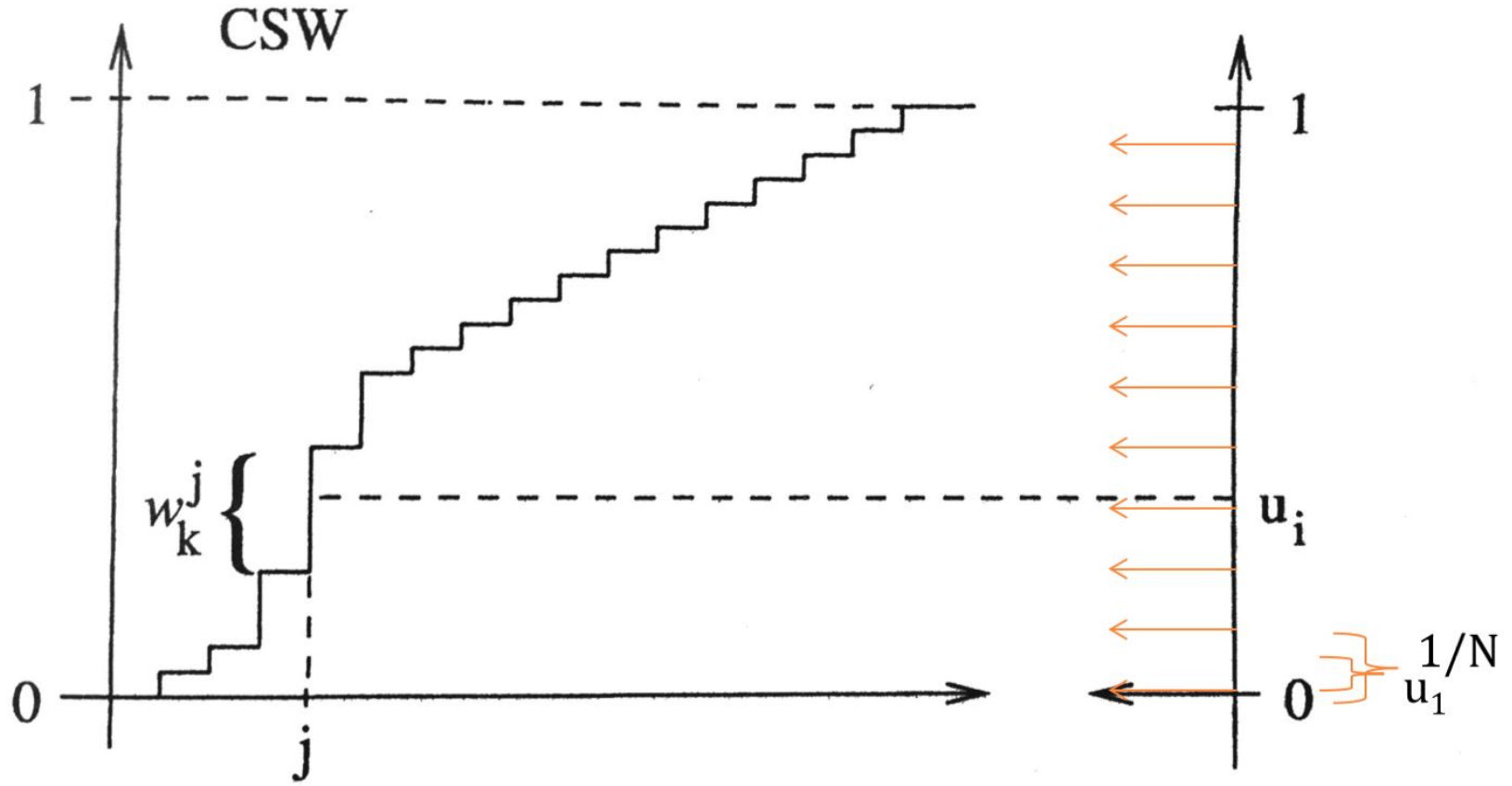


Figure 3.1 The process of resampling: $u_i \sim \mathcal{U}[0, 1]$ maps into index j; the corresponding particle $\mathbf{x}_k^j$ has a good chance of being selected and multiplied because of its high value of w_k^j.

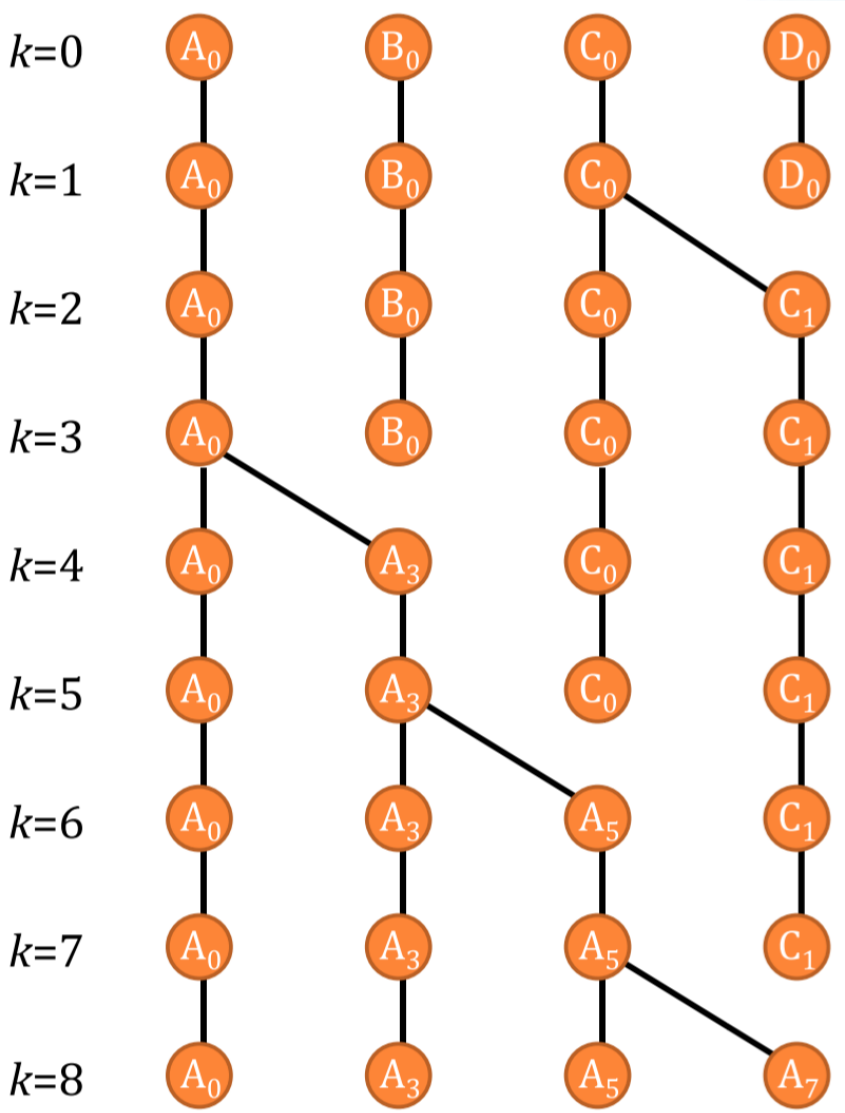
Comments on Resampling

- Pros
  - Reduces effects of degeneracy

- Cons
  - Limits opportunity to parallelize implementation since all particles must be combined
  - Particles with high importance weights are replicated. This leads to loss of diversity among particles, a.k.a. sample impoverishment
  - Diversity of particle paths is reduced, any smoothed estimate based on particles' paths degenerates

Trajectory Degeneracy Due to Resampling

- Implicitly, each particle represents a "guess" at the realization of the state sequence $\mathbf{x}_{0:k}$
- Resampling step causes some particle lineages to die
- Trajectories can eventually collapse to a single source node



Selection of Importance Density

- Selection of importance density $q(\cdot)$ is the most critical issue in the design of a PF.

- The optimal importance density that minimizes the variance of the importance weights conditioned upon $\mathbf{x}^i_{k-1}$ and $\mathbf{z}_k$ is†:

$$q_{\text{opt}}(\mathbf{x}_k \mid \mathbf{x}^i_{k-1}, \mathbf{z}_k) \equiv p(\mathbf{x}_k \mid \mathbf{x}^i_{k-1}, \mathbf{z}_k) = \frac{p(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{x}^i_{k-1})\, p(\mathbf{x}_k \mid \mathbf{x}^i_{k-1})}{p(\mathbf{z}_k \mid \mathbf{x}^i_{k-1})}$$

- This yields

$$w^i_k \propto \frac{p(\mathbf{z}_k \mid \mathbf{x}^i_k)\, p(\mathbf{x}^i_k \mid \mathbf{x}^i_{k-1})}{q(\mathbf{x}^i_k \mid \mathbf{x}^i_{k-1}, \mathbf{z}_k)} w^i_{k-1} = \frac{\cancel{p(\mathbf{z}_k \mid \mathbf{x}^i_k)}\, \cancel{p(\mathbf{x}^i_k \mid \mathbf{x}^i_{k-1})}}{\frac{\cancel{p(\mathbf{z}_k \mid \mathbf{x}^i_k, \mathbf{x}^i_{k-1})}\, \cancel{p(\mathbf{x}^i_k \mid \mathbf{x}^i_{k-1})}}{p(\mathbf{z}_k \mid \mathbf{x}^i_{k-1})}} w^i_{k-1}$$

optimal

$$w^i_k = p(\mathbf{z}_k \mid \mathbf{x}^i_{k-1})\, w^i_{k-1}$$

† A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197-208, 2000.

Comments on Optimal Importance Density

$$w_k^i = p(\mathbf{z}_k \mid \mathbf{x}_{k-1}^i)\, w_{k-1}^i$$

- In order to use optimal importance function, one has to be able to:
  - i) Sample from $p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k)$
  - ii) Evaluate $p(\mathbf{z}_k \mid \mathbf{x}_{k-1}^i) = \int p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i) d\mathbf{x}_k$

    (up to a normalizing constant)

- In general, neither i) nor ii) is straightforward

Suboptimal Importance Density

- Most popular suboptimal choice is the transitional density

$$q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i)$$

- This yields

$$w_k^i \propto \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^i)\, p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k)} w_{k-1}^i = \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^i)\, p(\cancel{\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i})}{p(\cancel{\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i})} w_{k-1}^i$$

suboptimal

$$w_k^i = p(\mathbf{z}_k \mid \mathbf{x}_k^i)\, w_{k-1}^i$$

Comments on Suboptimal Importance Density

$$w_k^i = p(\mathbf{z}_k \mid \mathbf{x}_k^i)\, w_{k-1}^i$$

- Pros
  - Importance weights can be easily evaluated $p(\mathbf{z}_k \mid \mathbf{x}_k^i)$
  - Importance density can be easily sampled $\mathbf{x}_k^i \sim p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)$
- Cons
  - With q_{opt}, weights are computed before the particles are propagated to time k. This is not possible with the suboptimal transitional prior.
  - State space is explored without any knowledge of the observation, hence filter can be inefficient

Particle Filter Algorithm (Sub-optimal BUT EFFICIENT version)

Algorithm **particle_filter**(S_{t-1}, $\mathbf{u}_{t-1}$, $\mathbf{z}_t$):

1. $S_t = \varnothing, \quad \eta = 0$
2. **For** $i = 1 \ldots n$ — ***Generate new samples***
3. Sample $\mathbf{x}_t^i$ from $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ using $\mathbf{x}_{t-1}^i$ and $\mathbf{u}_{t-1}$
4. $w_t^i = p(\mathbf{z}_t \mid \mathbf{x}_t^i)$ — ***Compute importance weight***
5. $\eta = \eta + w_t^i$ — ***Update normalization factor***
6. **For** $i = 1 \ldots N$
7. $w_t^i = w_t^i / \eta$ — ***Normalize weights***
8. Resample $\mathbf{x}_t^{i^*}$ from the discrete distribution given by w_t
9. ***Return*** $S_t = \{\mathbf{x}_t^{i^*}\}$

Particle Filter Algorithm (Sub-optimal BUT EFFICIENT version)

$$Bel(\mathbf{x}_t) = \eta\, p(\mathbf{z}_t \mid \mathbf{x}_t) \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1})\, Bel(\mathbf{x}_{t-1})\, d\mathbf{x}_{t-1}$$

draw $\mathbf{x}^i_{t-1}$ from $Bel(\mathbf{x}_{t-1})$

draw $\mathbf{x}^i_t$ from $p(\mathbf{x}_t \mid \mathbf{x}^i_{t-1}, \mathbf{u}_{t-1})$

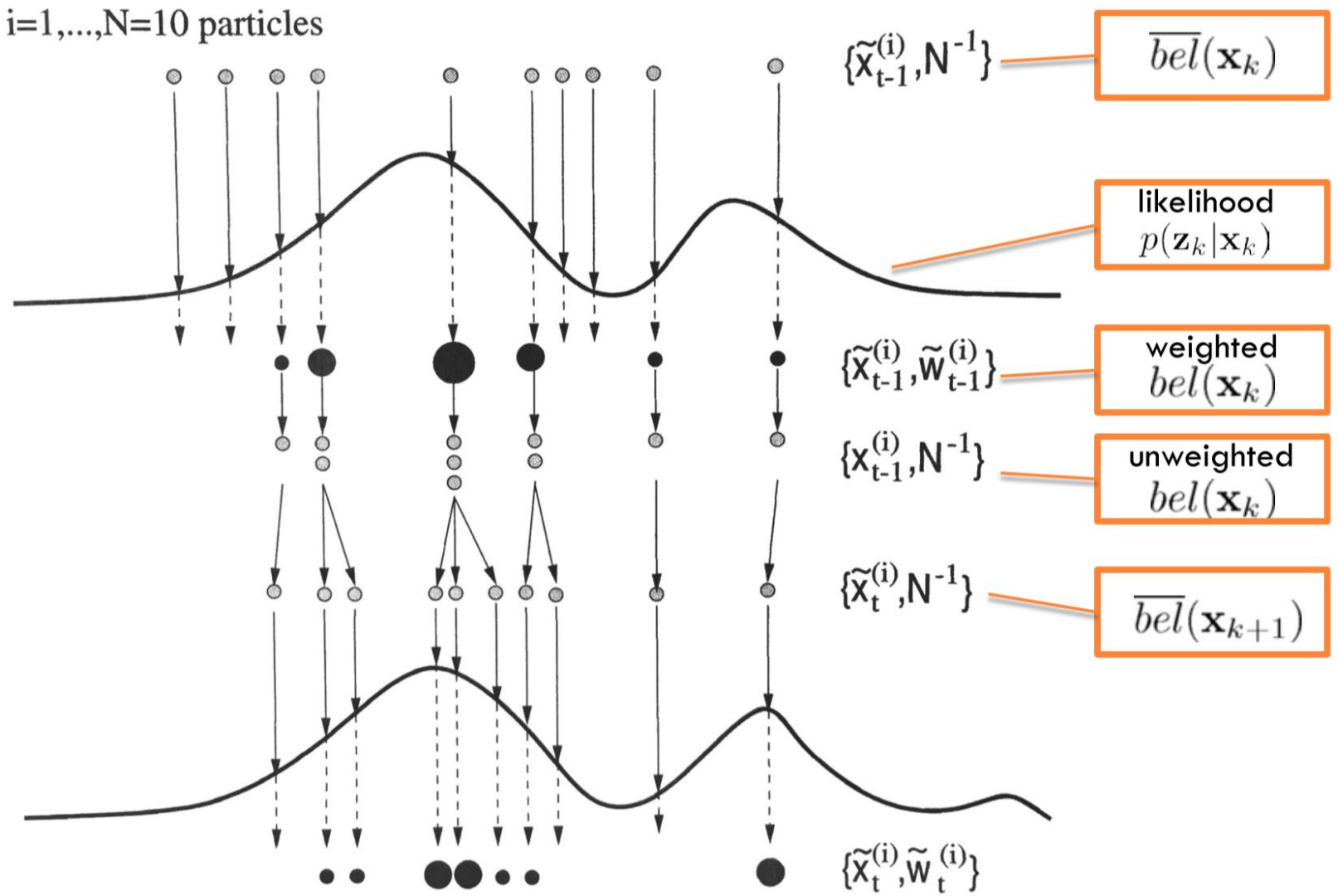
Importance factor for $\mathbf{x}^i_t$:

$$w^i_t = \frac{\text{target distribution}}{\text{proposal distribution}}$$

$$= \frac{\eta\, p(\mathbf{z}_t \mid \mathbf{x}_t)\, p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1})\, Bel(\mathbf{x}_{t-1})}{p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1})\, Bel(\mathbf{x}_{t-1})}$$

$$\propto p(\mathbf{z}_t \mid \mathbf{x}_t)$$

One Iteration of SIR PF



Monte Carlo Localization

- Each particle is a pose (trajectory) hypothesis
- Proposal is the motion model

$$\mathbf{x}_t^{[j]} \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t)$$

- Correction via the observation model

$$w_t^{[j]} = \frac{\text{target}}{\text{proposal}} \propto p(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{m})$$

Courtesy: Cyrill Stachniss

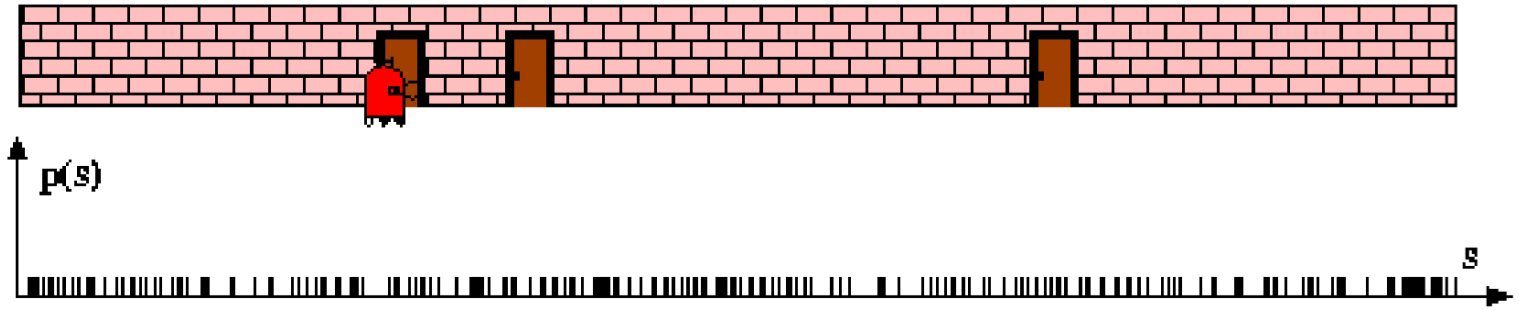
Particle Filter Localization

Particle_filter$(\mathcal{X}_{t-1}, \mathbf{u}_t, \mathbf{z}_t)$**:**

1: $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
2: *for* $j = 1$ *to* J *do*
3: *sample* $\mathbf{x}_t^{[j]} \sim p(\mathbf{x}_t \mid \mathbf{u}_t, \mathbf{x}_{t-1}^{[j]})$
4: $w_t^{[j]} = p(\mathbf{z}_t \mid \mathbf{x}_t^{[j]})$
5: $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle \mathbf{x}_t^{[j]}, w_t^{[j]} \rangle$
6: *endfor*
7: *for* $j = 1$ *to* J *do*
8: *draw* $i \in 1, \ldots, J$ *with probability* $\propto w_t^{[i]}$
9: *add* $\mathbf{x}_t^{[i]}$ *to* $\mathcal{X}_t$
10: *endfor*
11: *return* $\mathcal{X}_t$

Courtesy: Cyrill Stachniss

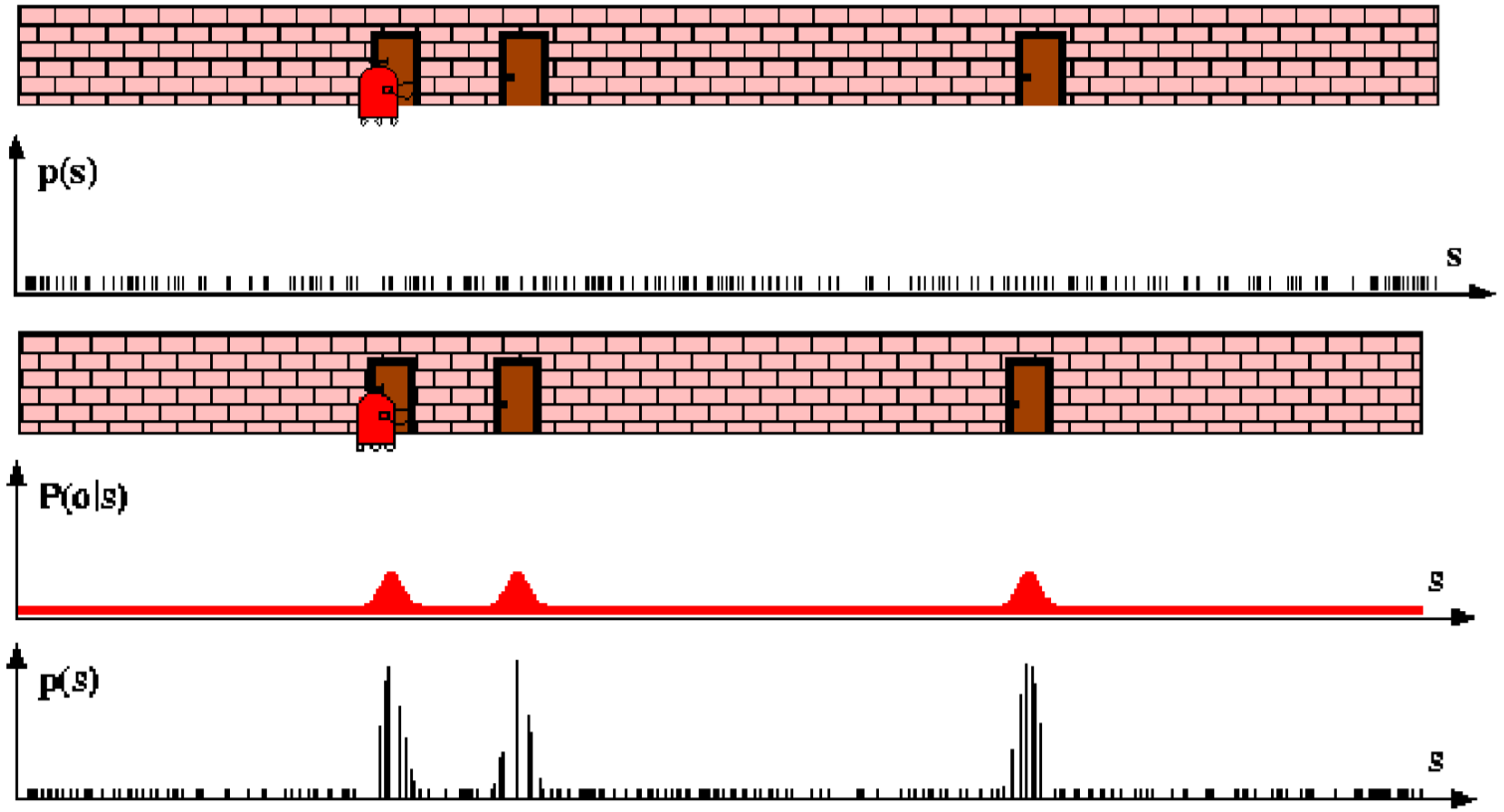
Particle Filters



Particle Filters: Sensor Information (Importance Sampling)

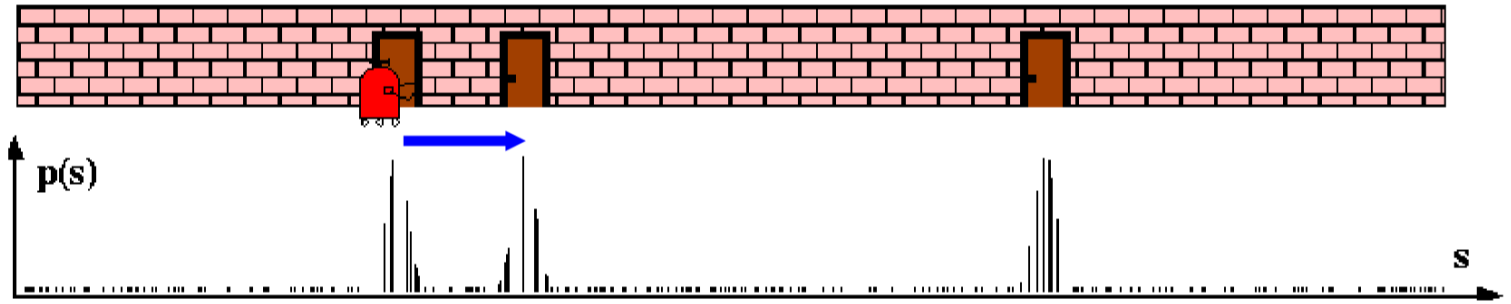
$$Bel(\mathbf{x}) \leftarrow \alpha\, p(\mathbf{z}\,|\,\mathbf{x})\, Bel^{-}(\mathbf{x})$$

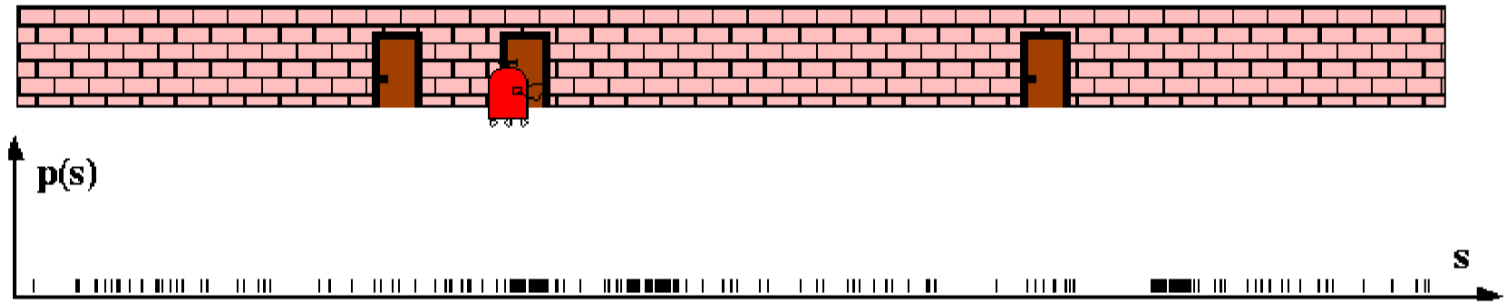
$$w \leftarrow \frac{\alpha\, p(\mathbf{z}\,|\,\mathbf{x})\, Bel^{-}(\mathbf{x})}{Bel^{-}(\mathbf{x})} = \alpha\, p(\mathbf{z}\,|\,\mathbf{x})$$



Particle Filters: Robot Motion

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

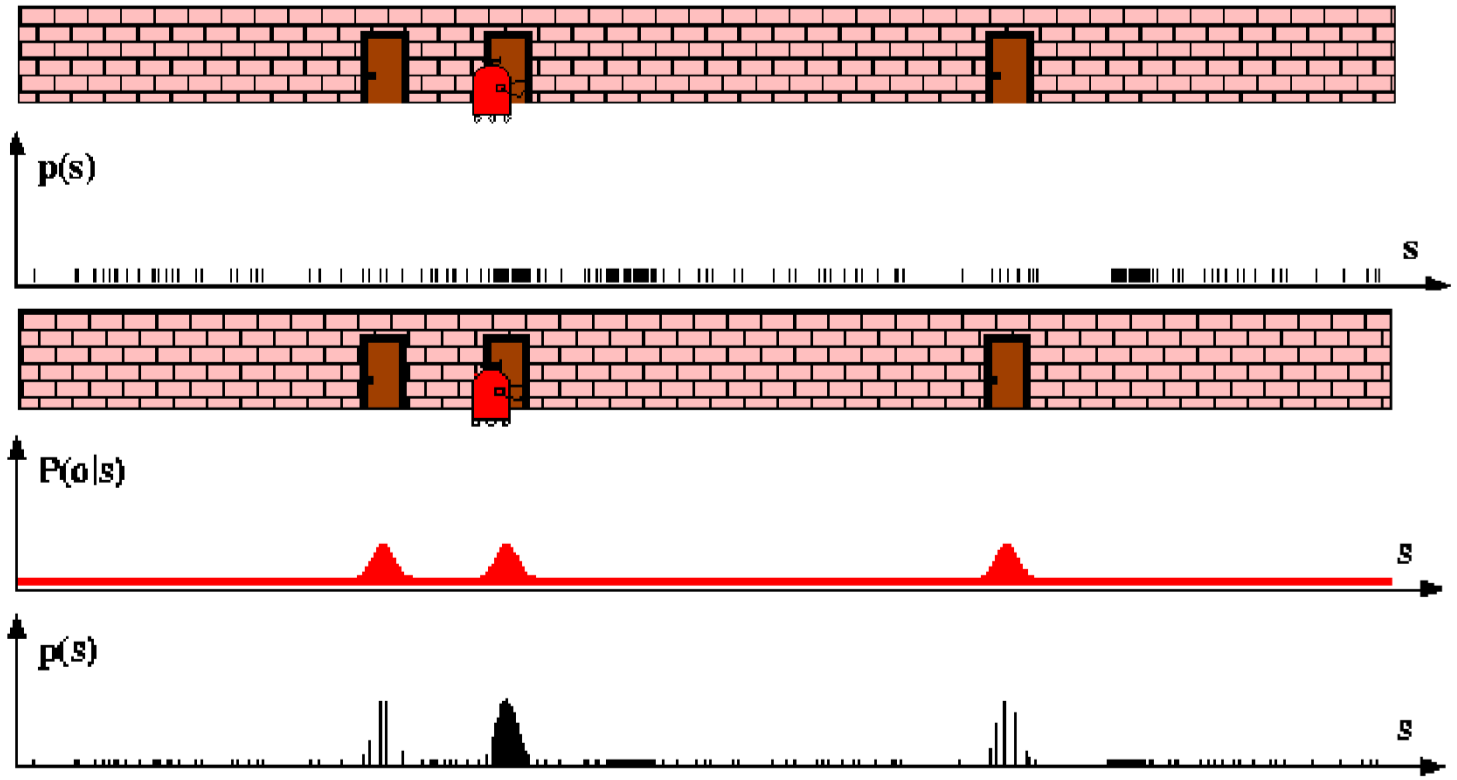




Particle Filters: Sensor Information (Importance Sampling)

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

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



Particle Filter: Robot Motion

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