

Foundations of AI

15. Robotics

State Estimation and Filtering

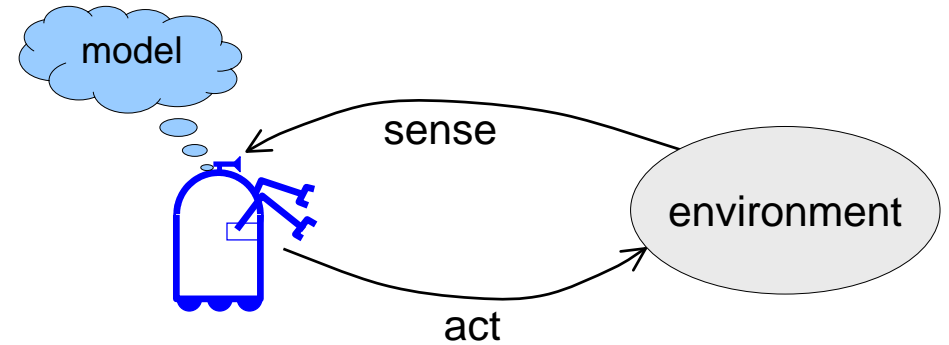
Wolfram Burgard, Bernhard Nebel, and Luc De Raedt

15/1

Robots

Physical agents that

- perceive their environment and
- generate actions to achieve their goals



15/2

Physical Agents are Inherently Uncertain

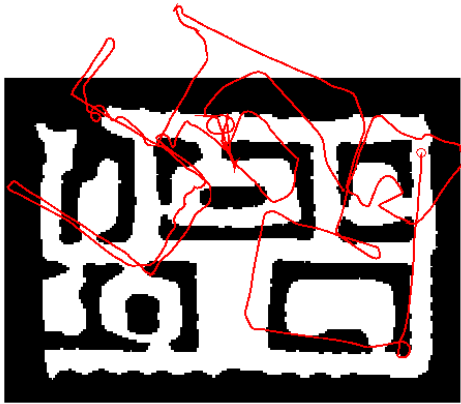
Uncertainty arises from four major factors:

- Environment stochastic, unpredictable
- Robot stochastic
- Sensor limited, noisy
- Models inaccurate

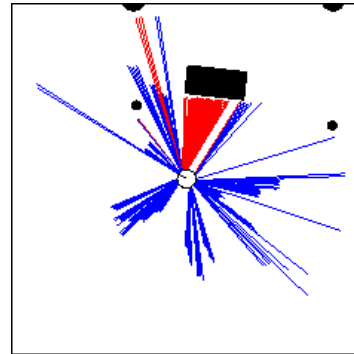
Example: Rhino



Nature of Sensor Data



Odometry Data



Range Data

Probabilistic Techniques for Physical Agents

Key idea:

Explicit representation of uncertainty using the calculus of probability theory

Perception = state estimation
Action = utility optimization

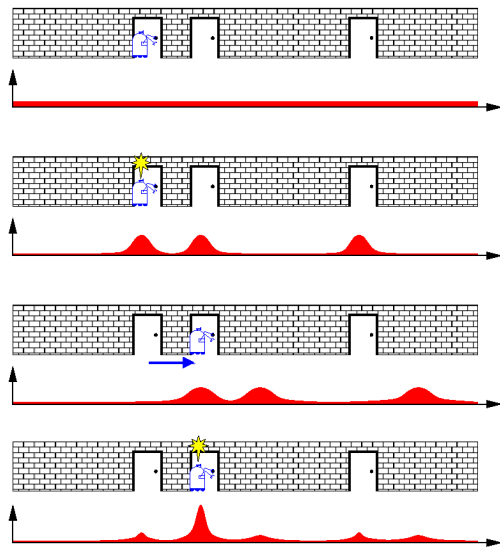
Advantages of Probabilistic Paradigm

- Can accommodate inaccurate models
- Can accommodate imperfect sensors
- Robust in real-world applications
- Best known approach to many hard robotics problems

Pitfalls

- Computationally demanding
- False assumptions
- Approximate

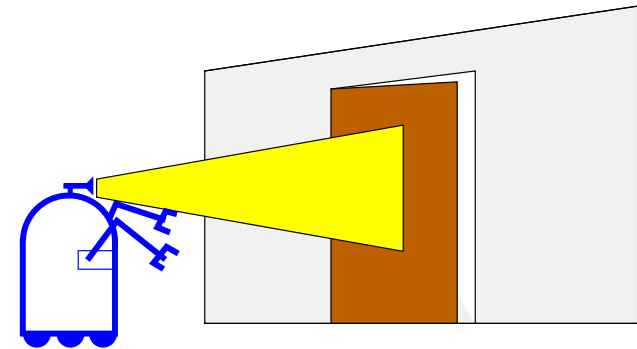
Probabilistic Localization



15/9

Simple Example of State Estimation

- Suppose a robot obtains measurement z
- What is $P(open|z)$?



Causal vs. Diagnostic Reasoning

- $P(open|z)$ is diagnostic.
- $P(z|open)$ is causal.
- Often causal knowledge is easier to obtain.
- Bayes rule allows us to use causal knowledge:

count frequencies!

$$P(open|z) = \frac{P(z|open)P(open)}{P(z)}$$

Example

- $P(z|open) = 0.6$ $P(z|\neg open) = 0.3$
- $P(open) = P(\neg open) = 0.5$

$$P(open|z) = \frac{P(z|open)P(open)}{P(z|open)p(open) + P(z|\neg open)p(\neg open)}$$

$$P(open|z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67$$

z raises the probability that the door is open.

Combining Evidence

- Suppose our robot obtains another observation z_2 .
- How can we integrate this new information?
- More generally, how can we estimate $P(x | z_1, \dots, z_n)$?

Recursive Bayesian Updating

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

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

$$\begin{aligned} P(x | z_1, \ominus, z_n) &= \frac{P(z_n | x) P(x | z_1, \ominus, z_{n-1})}{P(z_n | z_1, \ominus, z_{n-1})} \\ &= \eta P(z_n | x) P(x | z_1, \ominus, z_{n-1}) \\ &= \eta_{1\dots n} \prod_{i=1\dots n} P(z_i | x) P(x) \end{aligned}$$

Example: Second Measurement

- $P(z_2 | \text{open}) = 0.5$ $P(z_2 | \neg \text{open}) = 0.6$
- $P(\text{open} | z_1) = 2/3$

$$\begin{aligned} P(\text{open} | z_2, z_1) &= \frac{P(z_2 | \text{open}) P(\text{open} | z_1)}{P(z_2 | \text{open}) P(\text{open} | z_1) + P(z_2 | \neg \text{open}) P(\neg \text{open} | z_1)} \\ &= \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625 \end{aligned}$$

z_2 lowers the probability that the door is open.

Actions

- Often the world is dynamic because of
 - actions carried out by the robot,
 - actions carried out by other agents,
 - or just the time passing by change the world.
- How can we incorporate such actions?

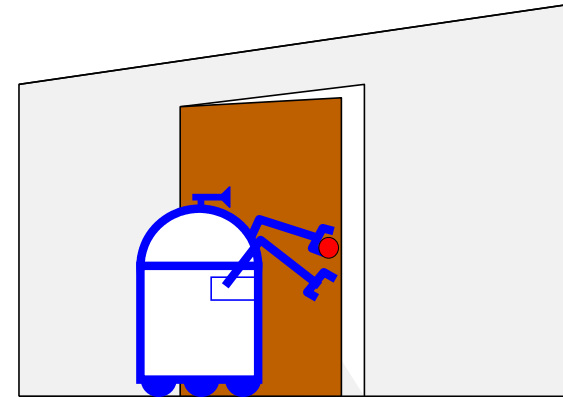
Modeling Actions

- To incorporate the outcome of an action u into the current "belief", we use the conditional pdf

$$P(x|u,x')$$

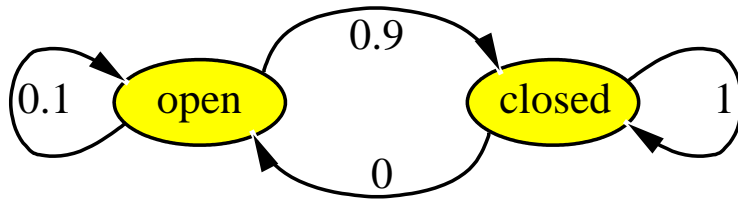
- This term specifies the pdf that executing u changes the state from x' to x .

Example: Closing the Door



State Transitions

$P(x|u,x')$ for $u =$ "close door":



If the door is open, the action "close door" succeeds in 90% of all cases.

Integrating the Outcome of Actions

Continuous case:

$$P(x|u) = \int P(x|u,x')P(x')dx'$$

Discrete case:

$$P(x|u) = \sum P(x|u,x')P(x')$$

Example: The Resulting Belief

$$\begin{aligned}
 P(\text{closed} | u) &= \sum P(\text{closed} | u, x')P(x') \\
 &= P(\text{closed} | u, \text{open})P(\text{open}) \\
 &\quad + P(\text{closed} | u, \text{closed})P(\text{closed}) \\
 &= \frac{9}{10} * \frac{5}{8} + \frac{1}{1} * \frac{3}{8} = \frac{15}{16} \\
 P(\text{open} | u) &= \sum P(\text{open} | u, x')P(x') \\
 &= P(\text{open} | u, \text{open})P(\text{open}) \\
 &\quad + P(\text{open} | u, \text{closed})P(\text{closed}) \\
 &= \frac{1}{10} * \frac{5}{8} + \frac{0}{1} * \frac{3}{8} = \frac{1}{16} \\
 &= 1 - P(\text{closed} | u)
 \end{aligned}$$

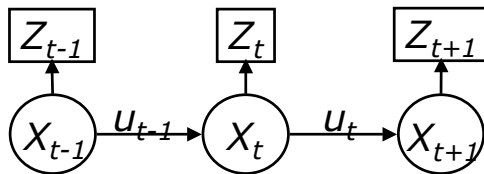
Bayes Filters: Framework

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

$$d_t = \{u_1, z_2 \ominus, u_{t-1}, z_t\}$$
 - 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**:

$$Bel(x_t) = P(x_t | u_1, z_2 \ominus, u_{t-1}, z_t)$$

Markov Assumption



$$\begin{aligned}
 p(d_t, d_{t-1}, \dots, d_0 | x_t, d_{t+1}, d_{t+2}, \dots) &= p(d_t, d_{t-1}, \dots, d_0 | x_t) \\
 p(d_t, d_{t+1}, \dots | x_t, d_1, d_2, \dots, d_{t-1}) &= p(d_t, d_{t+1}, \dots | x_t) \\
 p(x_t | u_{t-1}, x_{t-1}, d_{t-2}, \dots, d_0) &= p(x_t | u_{t-1}, x_{t-1})
 \end{aligned}$$

Underlying Assumptions

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

z = observation
 u = action
 x = state

Bayes Filters

$$Bel(x_t) = P(x_t | u_1, z_2 \ominus, u_{t-1}, z_t)$$

$$\text{Bayes} = \eta P(z_t | x_t, u_1, z_2, \ominus, u_{t-1}) P(x_t | u_1, z_2, \ominus, u_{t-1})$$

$$\text{Markov} = \eta P(z_t | x_t) P(x_t | u_1, z_2, \ominus, u_{t-1})$$

$$\text{Total prob.} = \eta P(z_t | x_t) \int P(x_t | u_1, z_2, \ominus, u_{t-1}, x_{t-1}) P(x_{t-1} | u_1, z_2, \ominus, u_{t-1}) dx_{t-1}$$

$$\text{Markov} = \eta P(z_t | x_t) \int P(x_t | u_{t-1}, x_{t-1}) P(x_{t-1} | u_1, z_2, \ominus, u_{t-1}) dx_{t-1}$$

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

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

1. Algorithm **Bayes_filter**($Bel(x), d$):
2. $\eta = 0$
3. if d is a **perceptual** data item z then
4. For all x do
5. $Bel'(x) = P(z | x) Bel(x)$
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. $Bel'(x) = \int P(x | u, x') Bel(x') dx'$
12. return $Bel'(x)$

Bayes Filters are Familiar!

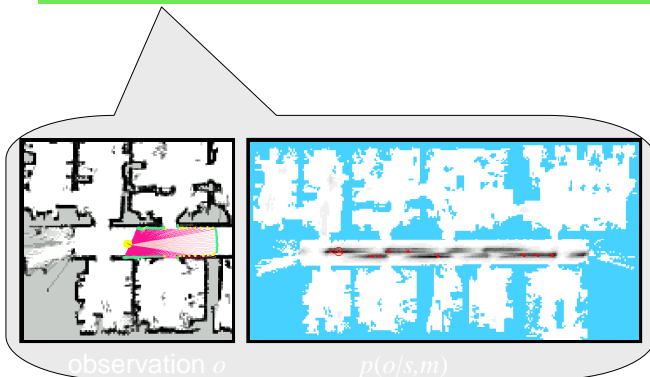
$$Bel(x_t) = \eta P(z_t | x_t) \int P(x_t | u_{t-1}, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

- Kalman filters
- Particle filters
- Hidden Markov models
- Dynamic Bayes networks
- Partially Observable Markov Decision Processes (POMDPs)

Localization with Bayes Filters (1)

$$bel(x_t) = \eta p(s_t | x_t) \int p(x_t | x_{t-1}, a_{t-1}) bel(x_{t-1}) dx_{t-1}$$

$$\Rightarrow bel(x_t | m) = \eta p(s_t | x_t, m) \int p(x_t | x_{t-1}, a_{t-1}, m) bel(x_{t-1} | m) dx_{t-1}$$



Localization with Bayes Filters (2)

$$bel(x_t) = \eta p(s_t | x_t) \int p(x_t | x_{t-1}, a_{t-1}) bel(x_{t-1}) dx_{t-1}$$

$$\Rightarrow bel(x_t | m) = \eta p(s_t | x_t, m) \int p(x_t | x_{t-1}, a_{t-1}, m) bel(x_{t-1} | m) dx_{t-1}$$

Motion: $bel^-(x_t) = \int p(x_t | x_{t-1}, a_{t-1}) bel(x_{t-1}) dx_{t-1}$

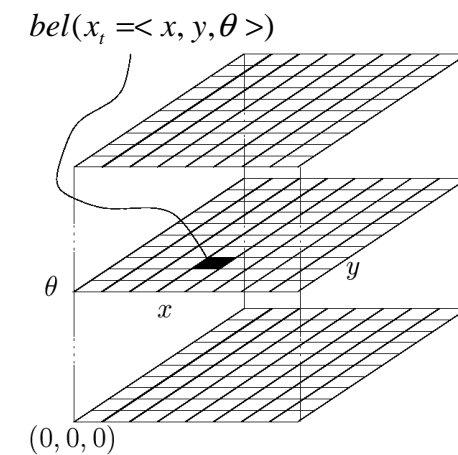
Perception: $bel(x_t) = \eta p(s_t | x_t) bel^-(x_t)$

... is optimal under the Markov assumption

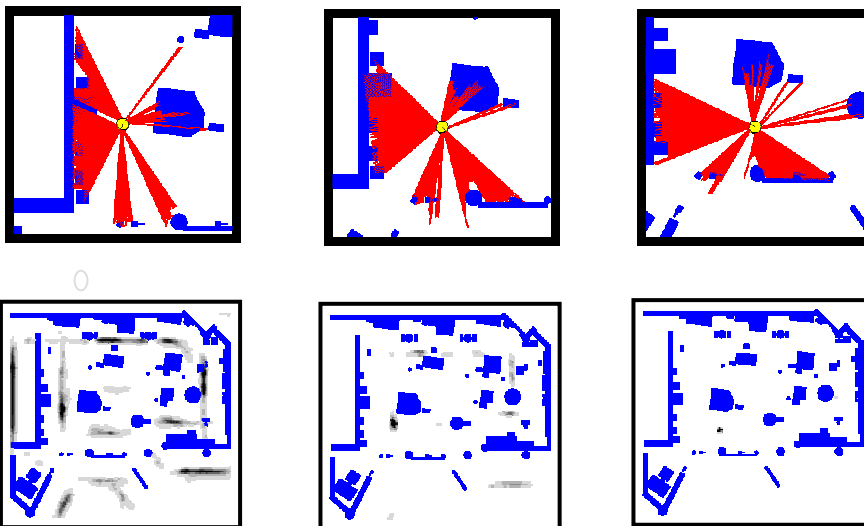
What is the Right Representation?

- Kalman filters
- Multi-hypothesis tracking
- Grid-based representations
- Topological approaches
- Particle filters

Piecewise Constant Representation (Histograms)



Grid-based Localization



Particle Filters

- Represent density 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]

Monte-Carlo Localization

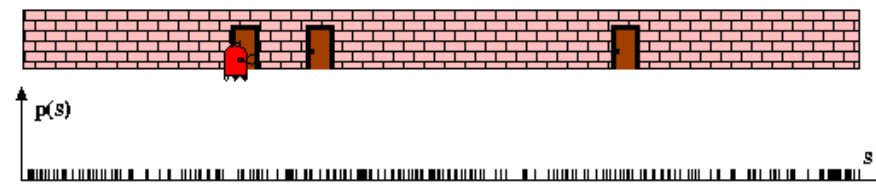
- Set of N samples $\{ \langle x_1, w_1 \rangle, \dots, \langle x_N, w_N \rangle \}$ containing a **state** x and an **importance weight** w
- Initialize sample set according to prior knowledge
- For each **motion** u do:
 - **Sampling**: Generate from each sample a new pose according to the motion model $p(x | u, x')$

$$x_n \leftarrow x_n + u'$$
- For each **observation** z do:
 - **Importance sampling**: weigh each sample with the likelihood

$$w_n \leftarrow \alpha p(z | x_n)$$
 - **Re-sampling**: Draw N new samples from the sample set according to the importance weights

15/33

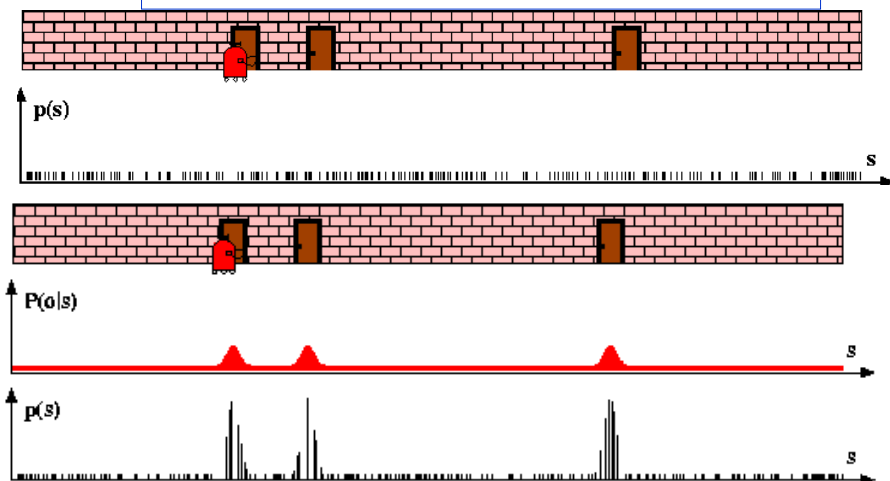
MCL: Global Localization



MCL: Sensor Update

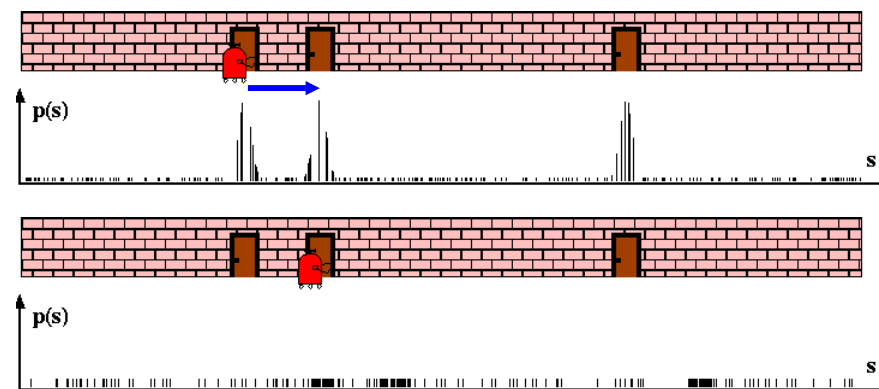
$$Bel(x) \leftarrow \alpha p(z | x) Bel^-(x)$$

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



MCL: Robot Motion

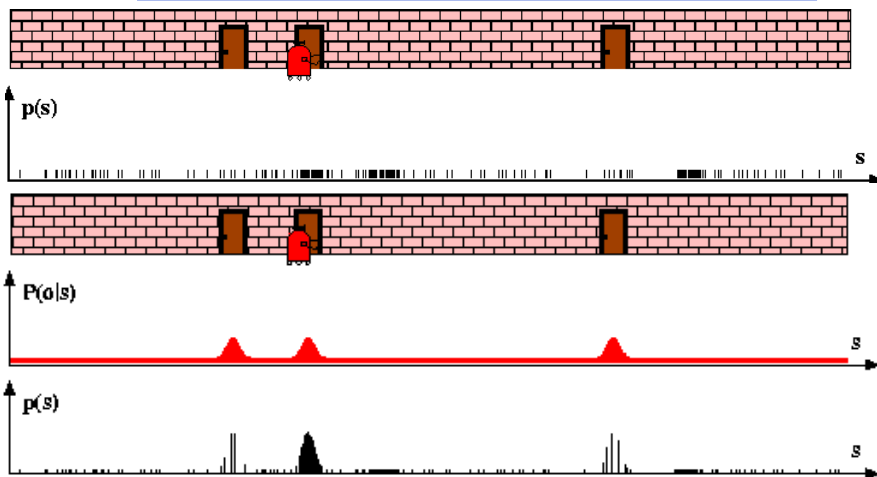
$$Bel^-(x) \leftarrow \int p(x | u, x') Bel(x') dx'$$



MCL: Sensor Update

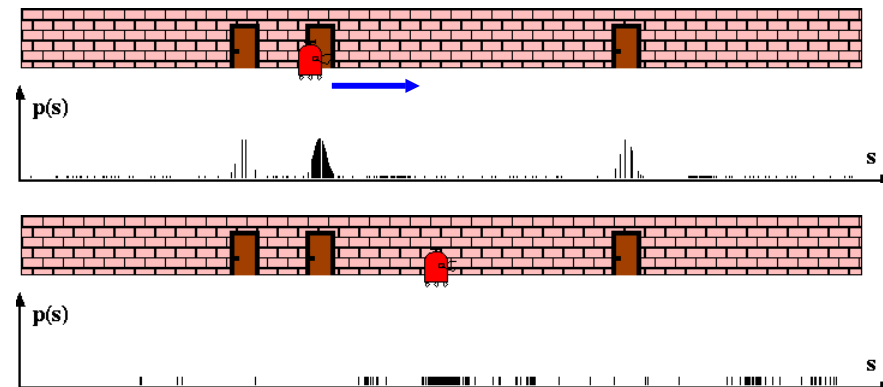
$$Bel(x) \leftarrow \alpha p(z|x) Bel^-(x)$$

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



MCL: Robot Motion

$$Bel^-(x) \leftarrow \int p(x|u, x') Bel(x') dx'$$



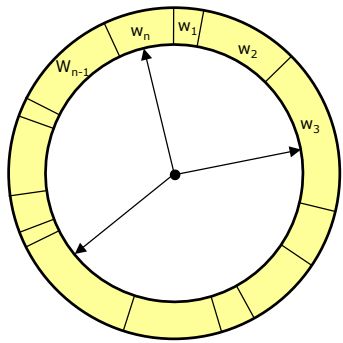
Particle Filter Algorithm

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

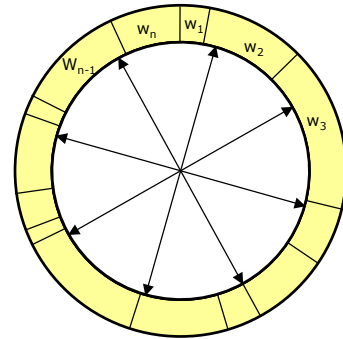
Resampling

- **Given:** Set S of weighted samples.
- **Wanted :** Random sample, where the probability of drawing x_i is given by w_i .
- Typically done n times with replacement to generate new sample set S' .

Resampling

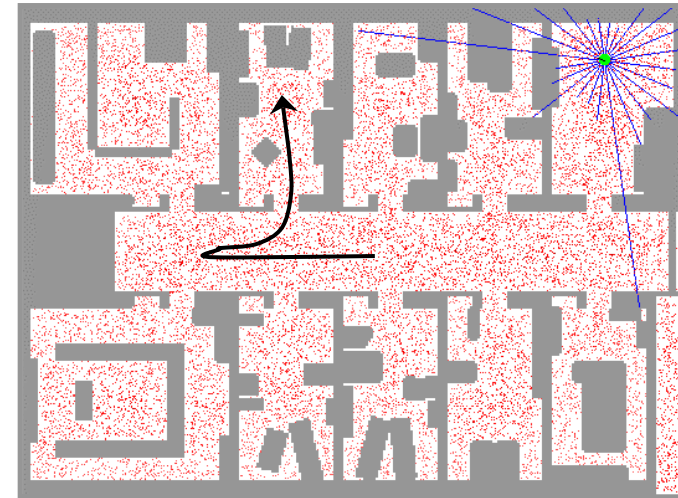


- Roulette wheel
- Binary search, $\log n$



- Stochastic universal sampling
- Systematic resampling
- Linear time complexity
- Easy to implement, low variance

MCL: Global Localization (Sonar)



Summary

- Bayes filters are a probabilistic tool for estimating the state of dynamic systems.
- It can be used in the context of mobile robotics to robustly estimate the pose of a vehicle.
- Particle filters are an efficient implementation of Recursive Bayesian filtering.