ECE 484: Principles of Safe Autonomy (Fall 2025)
Lecture 12: Filtering and Localization

Professor: Huan Zhang

https://publish.illinois.edu/safe-autonomy/

https://huan-zhang.com

huanz@illinois.edu

Slides adapted from Prof. Sayan Mitra’s slides for Spring 2025;

Reference: Probabilistic Robotics by Sebastian Thrun, Wolfram Burgard, and Dieter Fox

Some slides are from the book’s website

Announcements

- GEM, F1-Tenth signup (check CampusWire)
- Safety Training required (Check CampusWire)

Outline of state estimation module

Problem. Estimate the current state x_t of the system from knowledge about past observations $z_{0:t}$, control inputs $u_{0:t}$, and map m

- Introduction: Localization problem, taxonomy
- Probabilistic models: motion and measurements
- Discrete Bayes Filter
- Histogram filter and grid localization
- Particle filter

Review of conditional probabilities

Random variable X takes values $\mathrm{x}_1, \mathrm{x}_2 \in \mathbb{R}^{\mathrm{n}}$

$\mathrm{P}(\mathrm{X} = \mathrm{x})$ is written as $\mathrm{P}(\mathrm{x})$

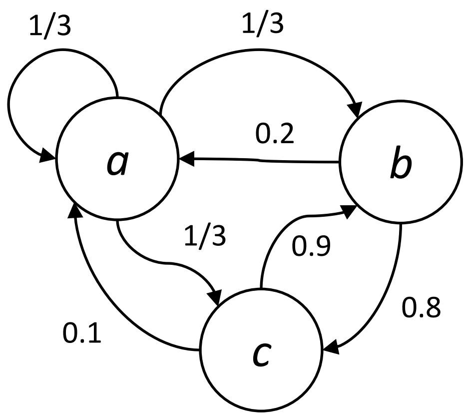
$\mathrm{P}(\mathrm{X} = \mathrm{x}, \mathrm{Z} = \mathrm{z})$ is written as $\mathrm{P}(\mathrm{x}, z)$

Conditional probability: $\mathrm{P}(\mathrm{X} = \mathrm{x} \,|\mathrm{Z} = \mathrm{z}) = \mathrm{P}(\mathrm{x}|\mathrm{z}) = \frac{\mathrm{P}(\mathrm{x},z)}{\mathrm{P}(\mathrm{z})}$ provided $\mathrm{P}(\mathrm{z}) > 0$

Bayes Rule $\mathrm{P}(\mathrm{x}|\mathrm{z}) = \frac{\mathrm{P}(\mathrm{Z}|\mathrm{X})\mathrm{P}(\mathrm{x})}{\mathrm{P}(\mathrm{z})}$, provided $\mathrm{P}(\mathrm{z}) > 0$

Evolution: probabilistic Markov Chain models

A probability distribution $\pi \in P(Q)$ over a finite set of states Q can be represented by a vector $\boldsymbol{\pi} \in \mathbb{R}^{|Q|}$ where $\sum \pi_i = 1$



Recall deterministic discrete transitions for automata $D: Q \rightarrow Q$

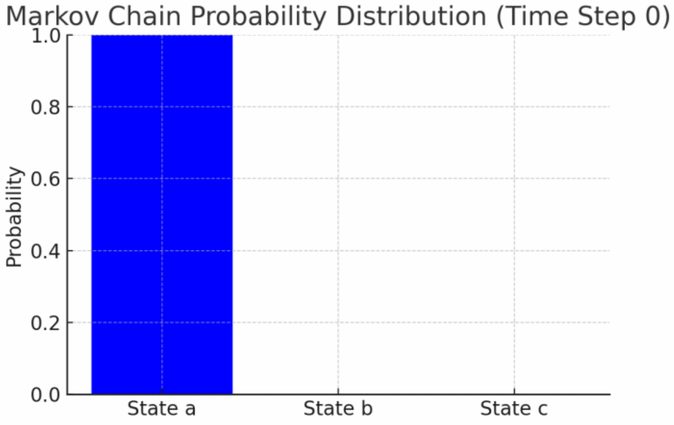
Probabilistic discrete transitions give a probability distribution $D: Q \rightarrow \boldsymbol{P}(Q)$ according to which the next state is chosen, i.e., $D(q)$ is a particular probability distribution over Q

For the example on the right $p_D(X_{t+1} = b \,|X_t = a) = \frac{1}{3}$, i.e., $D(a) = [a{:}\frac{1}{3} \quad b{:}\frac{1}{3} \quad c{:}\frac{1}{3}]$ $D(b) = [a{:}\frac{1}{5} \quad b{:}0 \quad c{:}\frac{4}{5}]$

$$\boldsymbol{D} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{5} & 0 & \frac{4}{5} \\ \frac{1}{10} & \frac{9}{10} & 0 \end{bmatrix}$$

Such a state machine model is called a **Markov chain**

A probabilistic transition $\boldsymbol{D}$ can be represented by a matrix $\boldsymbol{D} \in \mathbb{R}^{|Q|\times|Q|}$ where D_{ij} gives the probability of state i to transition to j

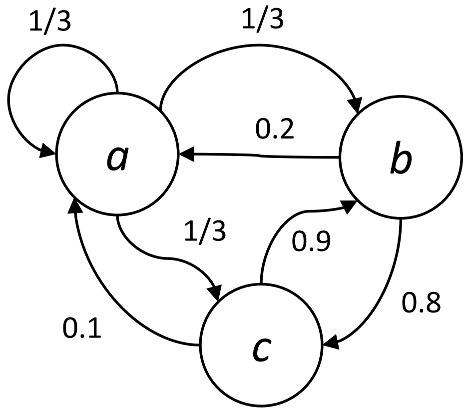


The evolution of the probability π over states can be represented as

$\pi_{t+1} = \pi_t \boldsymbol{D}$ starting with an initial distribution $\pi_0 \in \boldsymbol{P}(Q)$

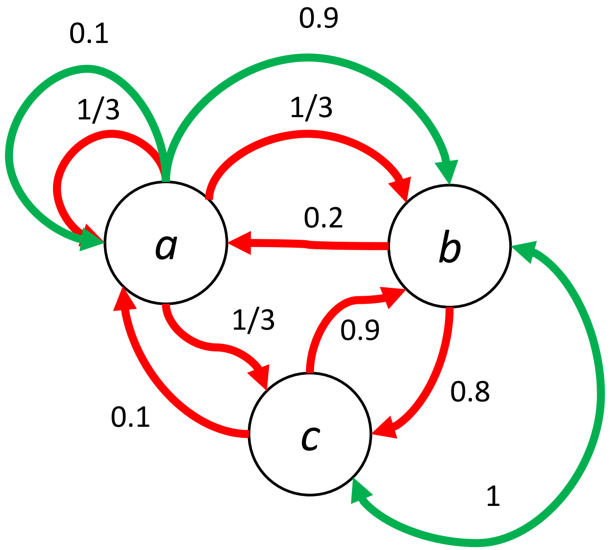
Evolution: probabilistic MDP models

More generally, transitions depend on input in which case the transition function $D: Q \times U \to P(Q)$ also depends on the control action U



For the example below $p_D(X_{t+1} = b \,|X_t = a, U_t = red) = \frac{1}{3}$

Such a state machine model with inputs is called a **Markov Decision Process (MDP)**



The probabilistic transitions $\boldsymbol{D}$ can be represented by a collection of matrices $\boldsymbol{D}: U \to \mathbb{R}^{|Q|\times|Q|}$ where $D_{ij}(u)$ gives the probability of state i to transition to j under action u

$p_D(x'|x, u)$ if transition probabilities are time invariant

Evolution and measurement: probabilistic models

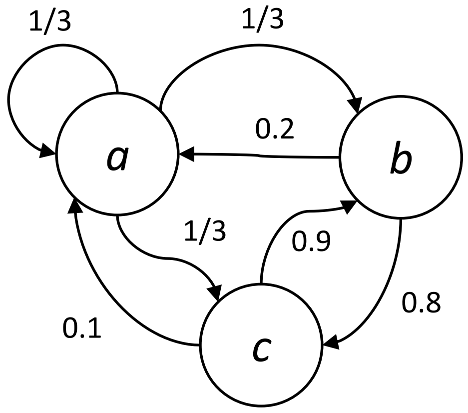
Even more generally, transitions depend on outputs and history

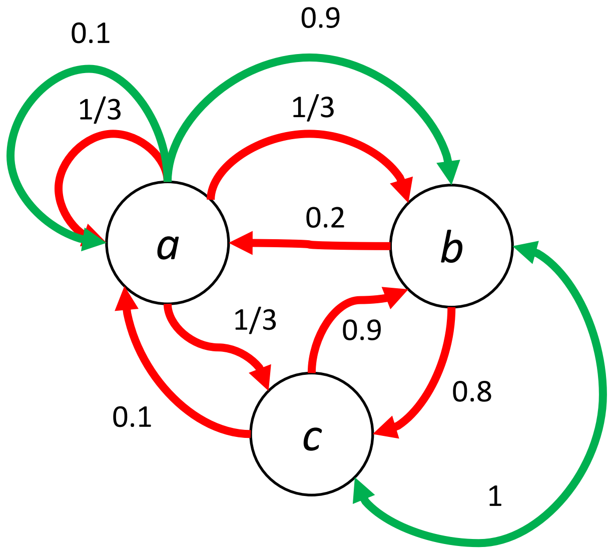
$p_D(X_t = x_t \,|X_0 = x_0, \ldots X_{t-1} = x_{t-1,} Z_1 = z_1, \ldots Z_{t-1} = z_{t-1,}, U_1 = u_1, \ldots U_t = u_{t,})$ describes state evolution model

$p_D(x_t \,|x_{0:t-1}, z_{1:t-1}, u_{1:t})$ describes motion/state evolution model

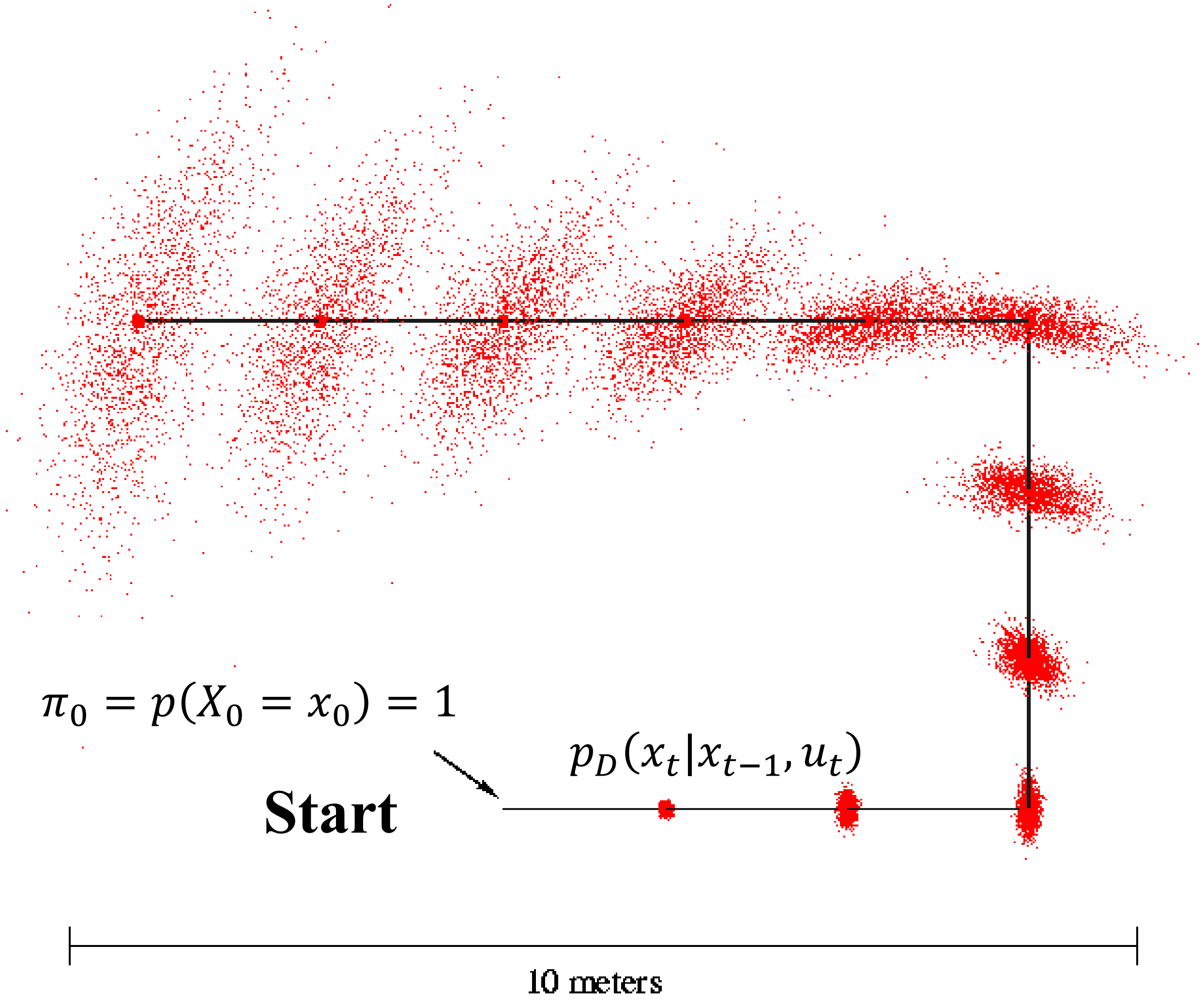
If state is complete, sufficient summary of the history then

- $p_D(x_t \,|x_{0:t-1}, z_{0:t-1}, u_{0:t-1}) = p_D(x_t \,|x_{t-1}, u_t)$ transition prob.
- $p_D(x'|x, u)$ if transition probabilities are time invariant





Example Motion Model without measurements



The state transition probabilities are defined by
$x_{t+1} = f(x_t, u_t) + \omega_t$

where $\omega_t \sim N(0,1)$

Probabilistic measurements

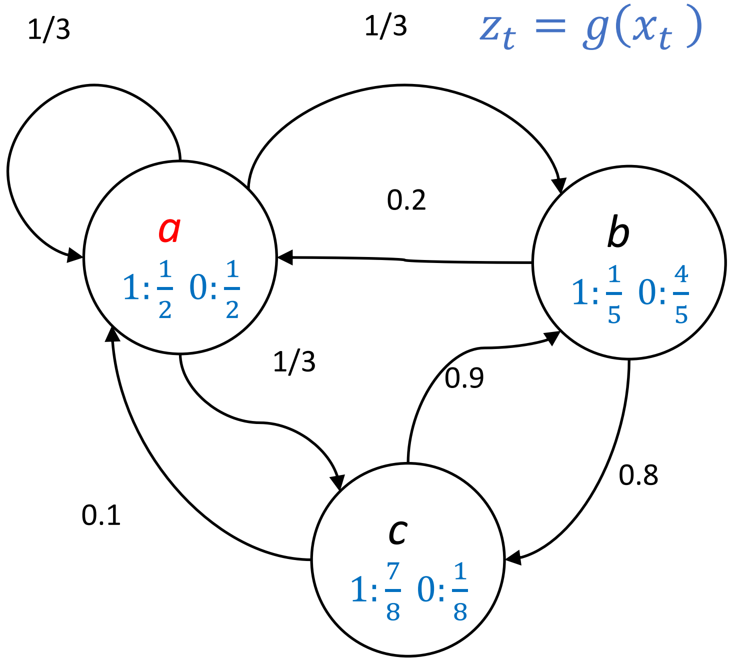
A **measurement model** gives the *output* or observation probability for a given state, e.g.:

$$p_M(z_t = 1 | x_t = a) = \frac{1}{2}$$

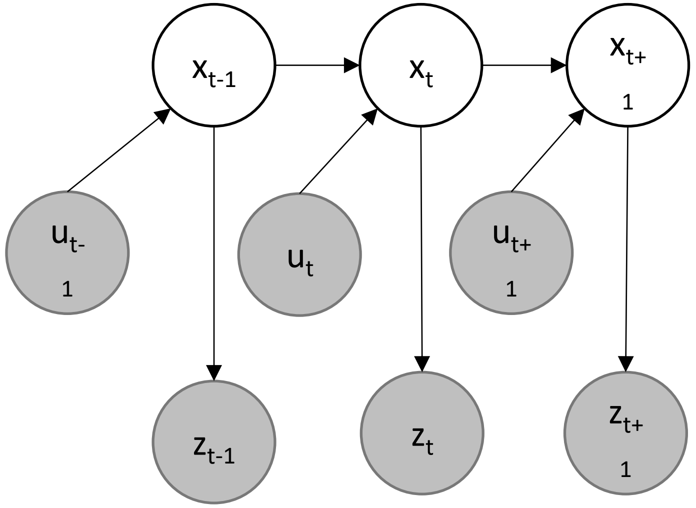
Generally, measurements can depend on history

$$p_M(z_t | x_{0:t}, z_{1:t-1}, u_{0:t-1})$$

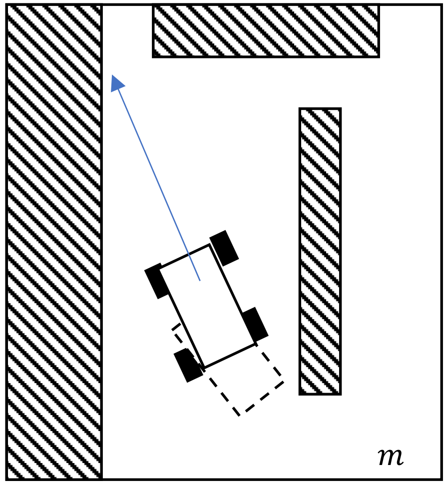
- If state is complete $p_M(z_t | x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t | x_t)$
- $p_M(z_t | x_t)$: measurement probability
- $p_M(z | x)$: time invariant measurement probability



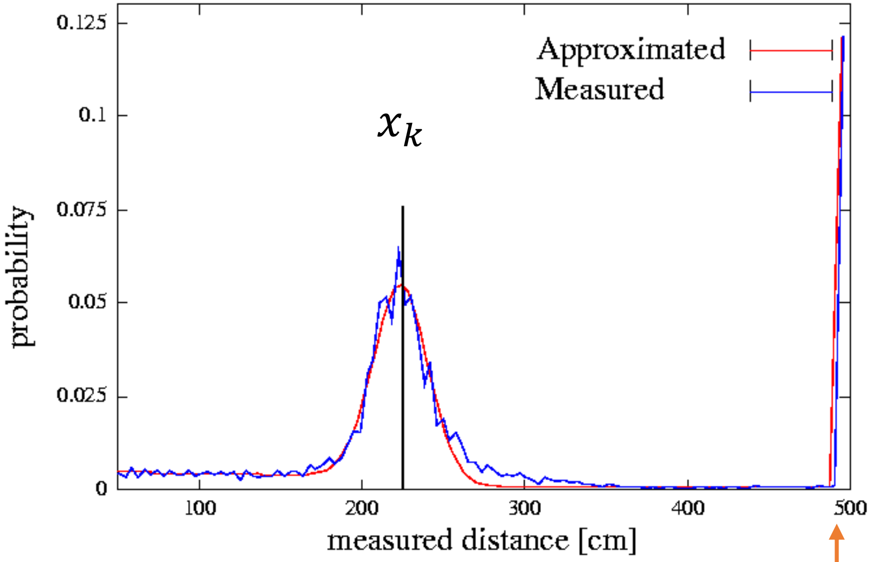
State a produces output 1 and 0 each with probability 0.5

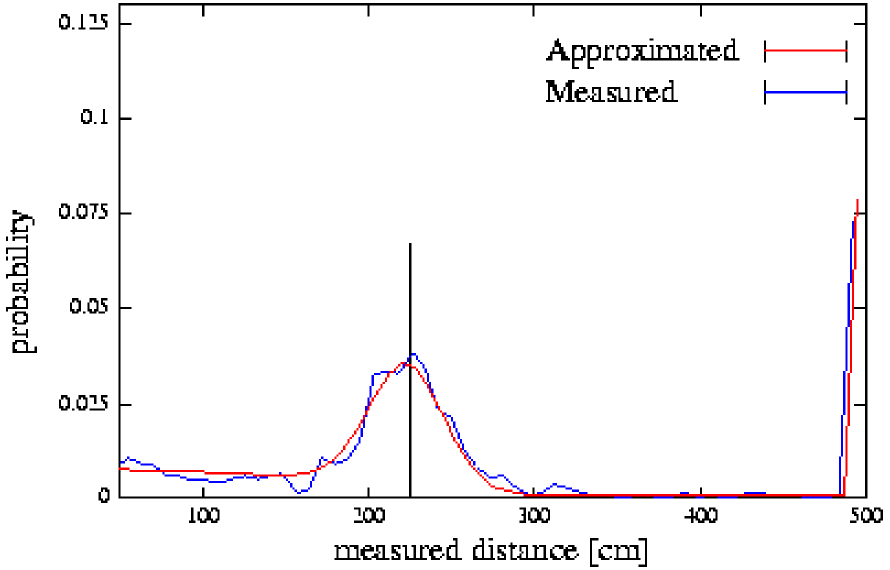


Example Proximity Sensor Measurement Models



$p_M(z_t \,|X_t = x_k)$





Laser sensor

Sonar sensor

max-range
spike

Motion and measurement models

$p_D(x_t \mid x_{0:t-1}, z_{0:t-1}, u_{1:t})$ describes motion/state evolution model

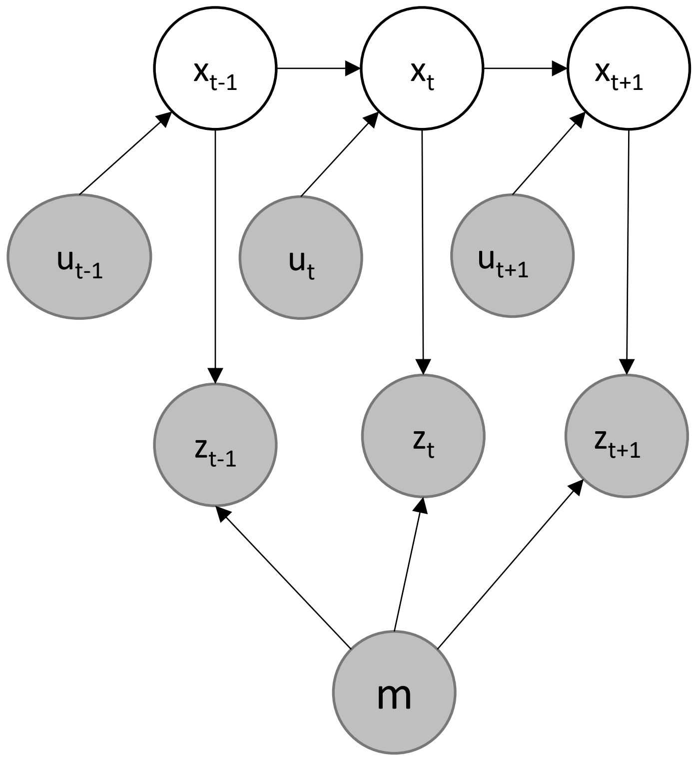
If state is complete (sufficient summary of the history) then

- $p_D(x_t \mid x_{0:t-1}, z_{0:t-1}, u_{0:t-1}) = p_D(x_t \mid x_{t-1}, u_t)$ **motion model**
- $p_D(x' \mid x, u)$ if transition probabilities are time invariant

$p_M(z_t \mid x_{0:t}, z_{0:t-1}, u_{0:t-1}, m)$ describes measurement

If state is complete

- $p_M(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}, m) = p_M(z_t \mid x_t, m)$ **measurement model**
- $p_M(z \mid x, m)$: time invariant measurement probability



Beliefs

Belief: Robot's knowledge about the state

True state x_t is not directly measurable or observable and the robot must infer or estimate state from measurements and this distribution of states is called the *belief*

$bel(x_t) = p(x_t|z_{1:t}, u_{1:t})$

Posterior distribution over state at time t given all past measurements and control. This will be calculated in two steps:

Initially: $bel(x_0) = \pi_0$

1. Prediction: $\overline{bel}(x_t) = p(x_t|z_{1:t-1}, u_{1:t})$ based on past measurements and control u
2. Correction: $bel(x_t)$ from $\overline{bel}(x_t)$ based on most recent measurement z_t

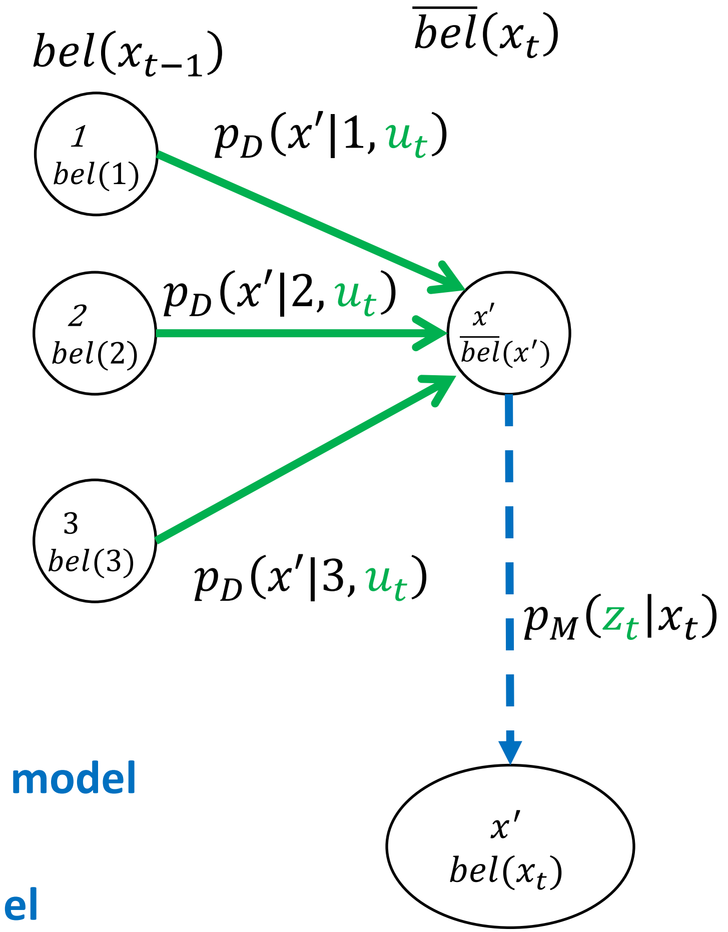
Bayes Filter: Prediction and Correction

Algorithm Bayes_filter($bel(x_{t-1}), u_t, z_t$) iteratively calculates $bel(x_t)$ given $bel(x_{t-1})$, the recent control u_t, and the measurement z_t

$bel(x_{t-1})$: $P(Q)$ is a probability distribution over Q

$\overline{bel}(x_t) = p(x_t | bel(x_{t-1}), z_{1:t-1}, u_{1:t}) = p(x_t | bel(x_{t-1}), u_t)$ is the intermediate belief which uses only prediction but not the most recent measurement

For discrete distributions for each $x' \in Q$ the beliefs can be calculated as



$$\overline{bel}(X_t = x') = \sum_{x \in Q} p_D(X_t = x' | X_{t-1} = x, U_t = u_t)\, bel(X_{t-1} = x)$$ **motion model**

$$bel(X_t = x') = \eta\, p_M(Z_t = z_t | X_t = x')\, \overline{bel}(X_t = x')$$ **measurement model**

where η is a normalizing constant to make $bel(x_t) \in \boldsymbol{P}(Q)$

Recall Bayes rule $P(x|z) = \frac{P(Z|X)P(x)}{P(z)}$, provided $P(z) > 0$

Histogram Filter or Discrete Bayes Filter

Notation: $bel(X_t = x_k) \coloneqq p_{k,t}$

Finitely many states $x_i, x_k, etc.$ Random state vector X_t

$p_{k,t}$: belief at time t for state x_k; discrete probability distribution

Algorithm Discrete_Bayes_filter($\{p_{k,t-1}\}, u_t, z_t$):

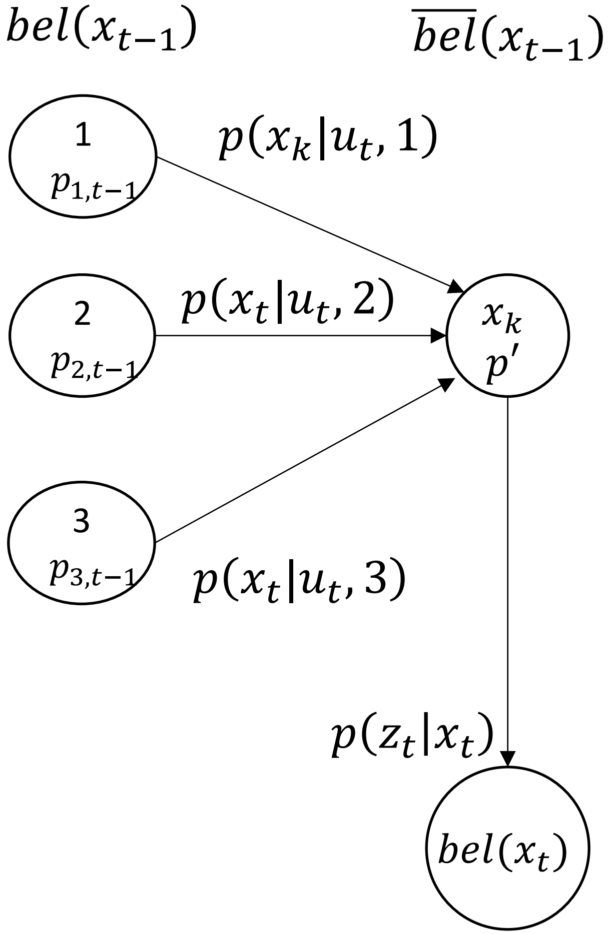
for all k do:

$$\bar{p}_{k,t} = \sum_i p(X_t = x_k | u_t, X_{t-1} = x_i) p_{i,t-1}$$

$$p_{k,t} = \eta\, p(z_t | X_t = x_k) \bar{p}_{k,t}$$

end for

return $\{p_{k,t}\}$



Bayes Filter: Continuous Distributions

Algorithm Bayes_filter($bel(x_{t-1}), u_t, z_t$)

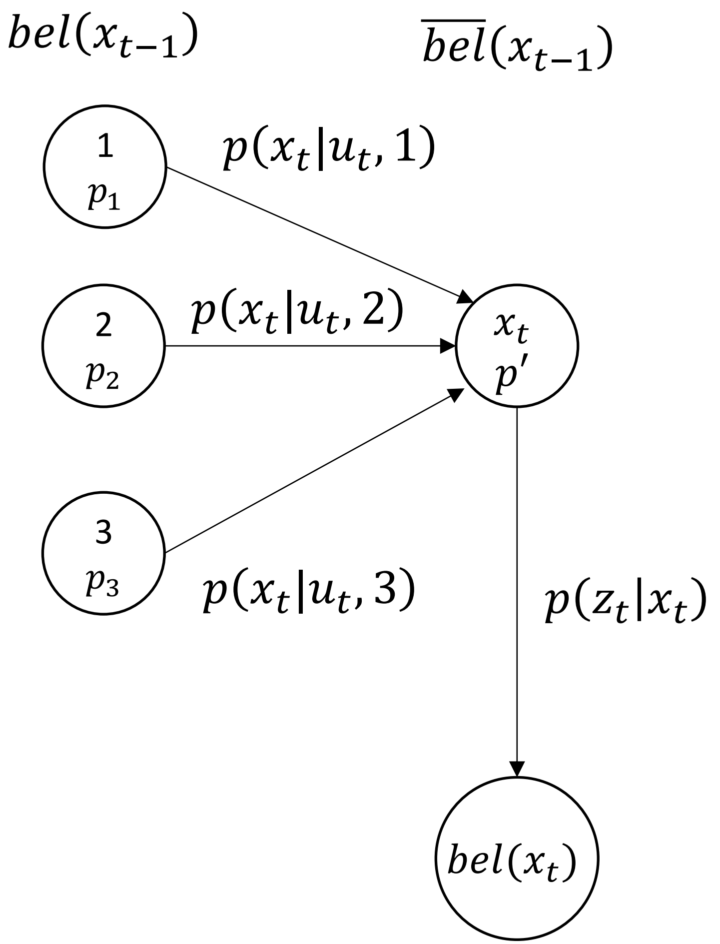
for all x_t do:

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

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

end for

return $bel(x_t)$



Grid Localization

Solves global localization in some cases kidnapped robot problem using Bayes filter

Can process raw sensor data

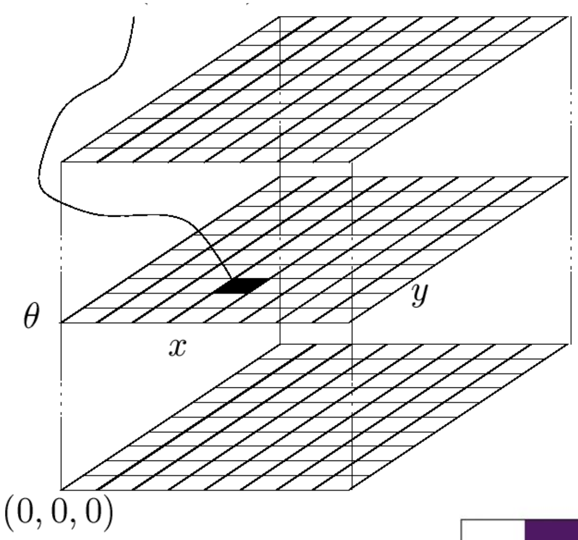
- No need for feature extraction

Non-parametric method, i.e., does not rely on specific form of probability distributions

- In particular, not bound to unimodal distributions (unlike Extended Kalman Filter)

Grid localization with bicycle model + landmarks

$bel(X_t = \langle x, y, \theta \rangle)$

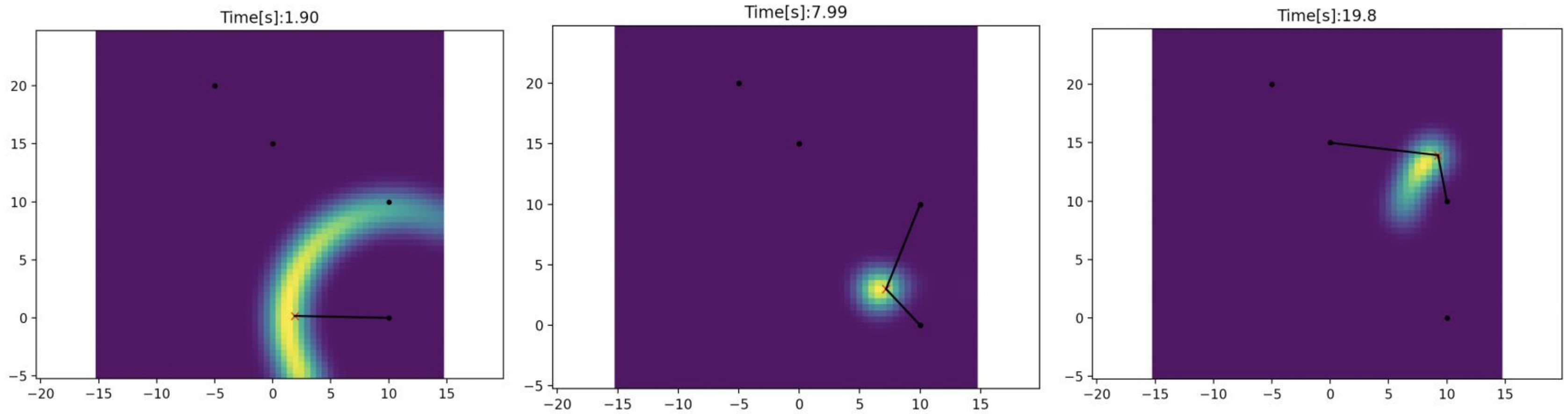


The state space Q is a quantization of position and orientation $q = \langle x, y, \theta \rangle$

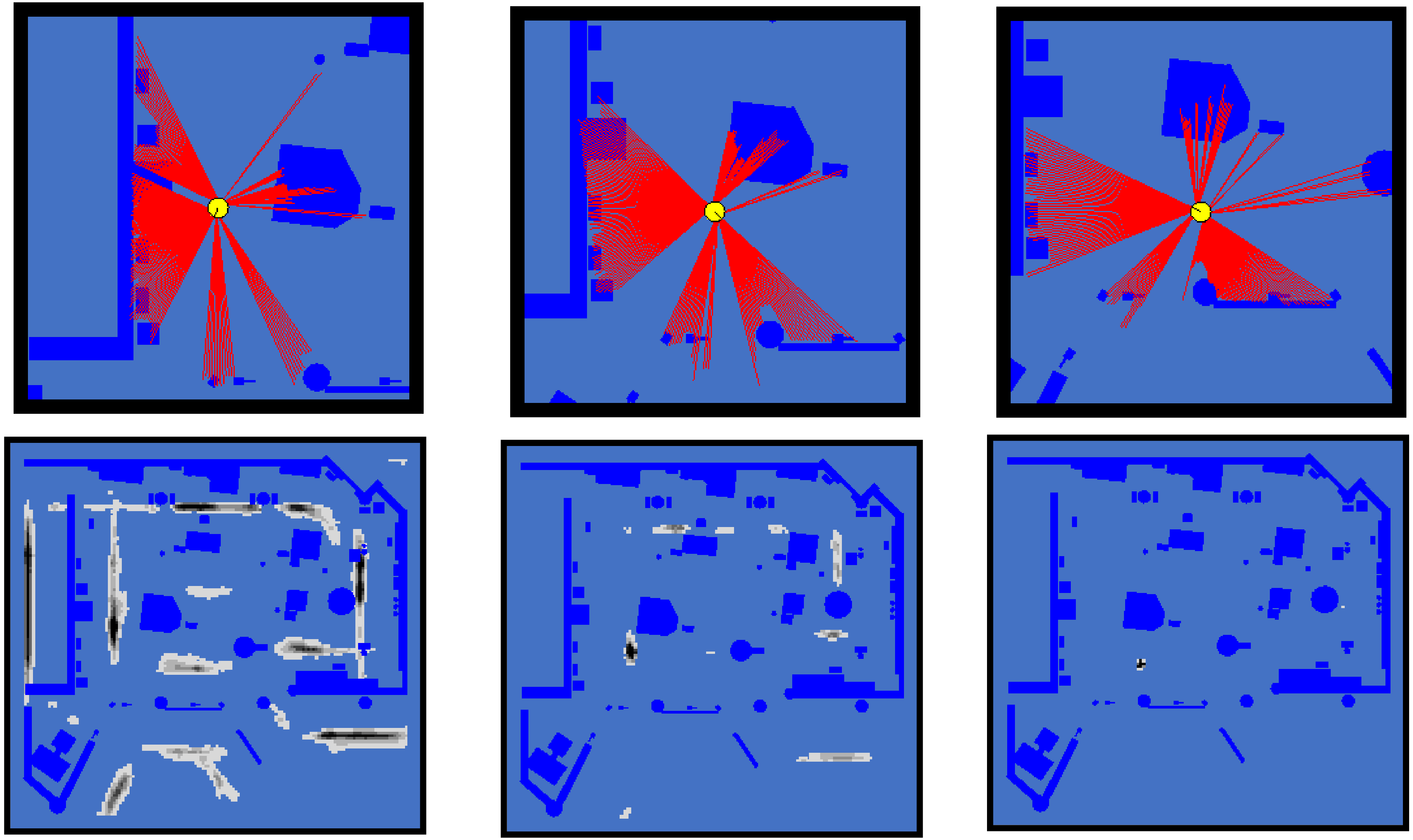
A belief is a probability distribution over states $\text{bel}(q_t) \in P(Q)$

Prediction: Fixing an (steering) input u_t compute the new intermediate belief over Q using motion model $p_D(q_{t+1}|q_t, u_{t+1})$

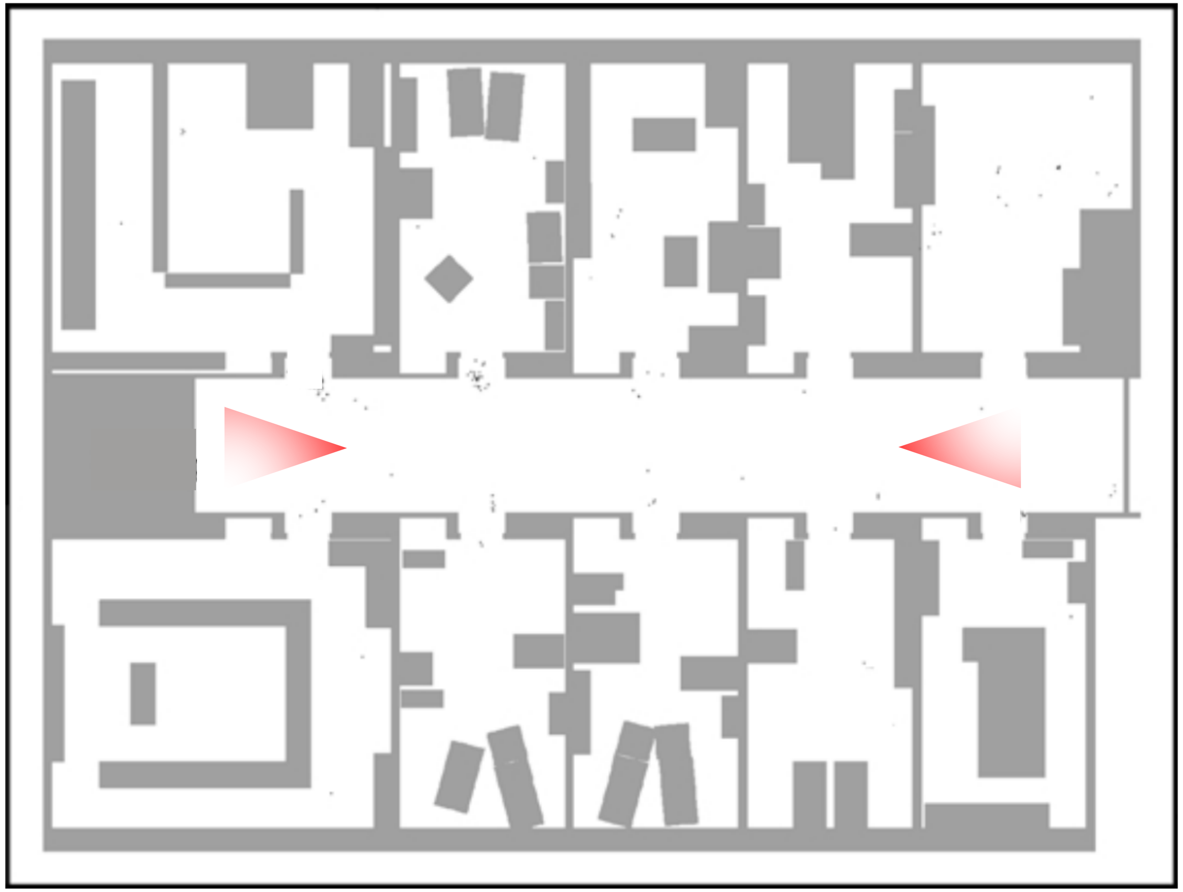
Correction: Update intermediate belief with received distance to landmark z_{t+1} based on measurement model p_M



Grid-based Localization



Ambiguity in global localization arising from locally symmetric environment



Grid localization

Algorithm Grid_localization ($\{p_{k,t-1}\}, u_t, z_t, m$)

for all k do:

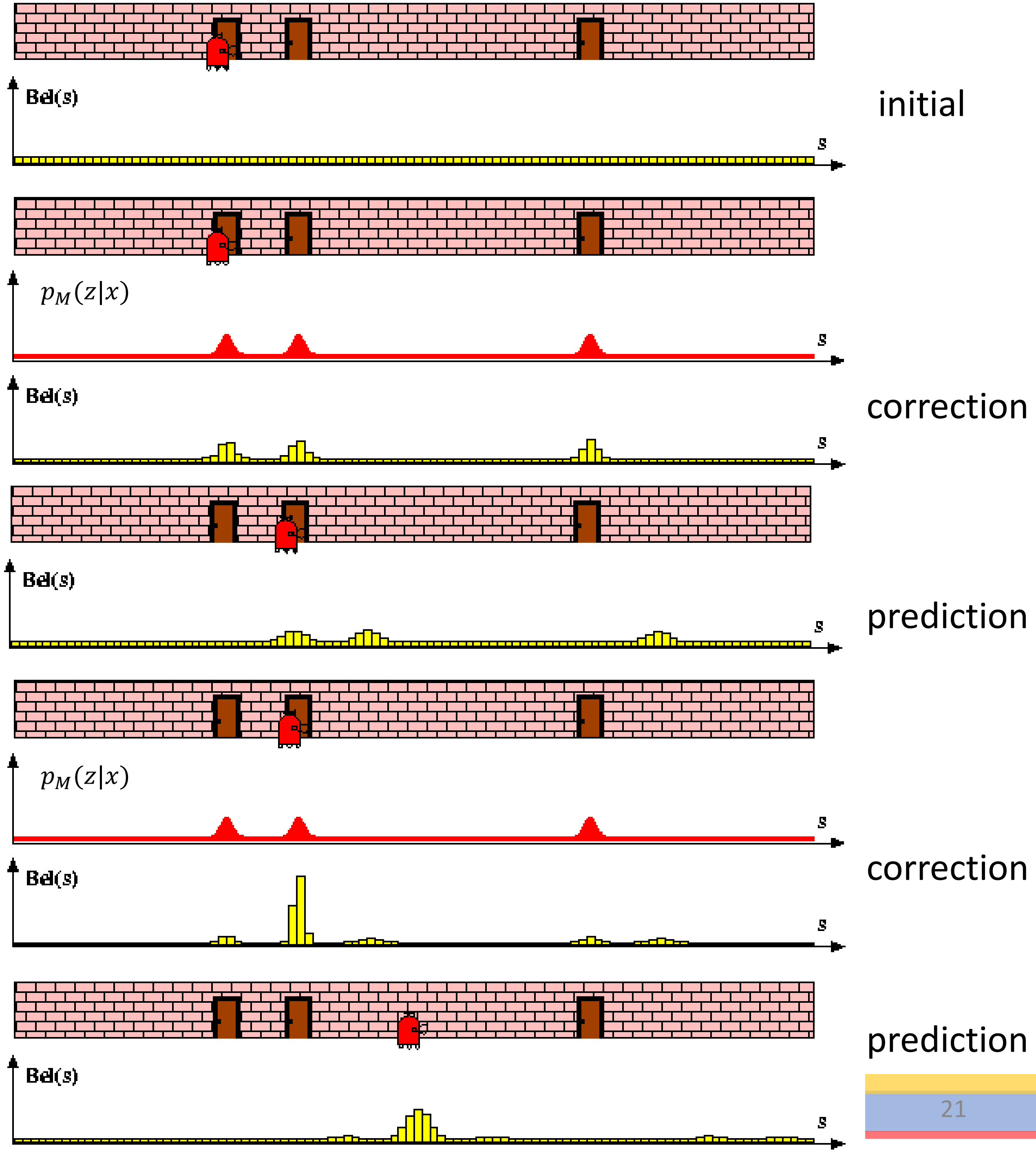
$$\bar{p}_{k,t} = \sum_i p_{i,t-1}\, \boldsymbol{motion\_model}(mean(x_k), u_t, mean(x_i))$$

$$p_{k,t} = \eta\, \bar{p}_{k,t}\, \boldsymbol{measurement\_model}(z_t, mean(x_k), m)$$

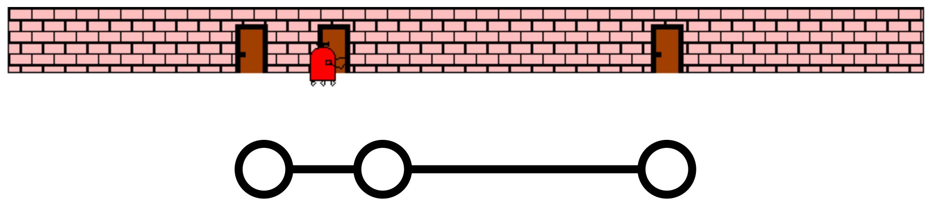
end for

return $bel(x_t)$

Grid localization, $bel(x_t)$ represented by a histogram over grid



Summary



- Key variable: Grid resolution
- Two approaches
  - Topological: break-up pose space into regions of significance (landmarks)
  - Metric: fine-grained uniform partitioning; more accurate at the expense of higher computation costs
- Important to compensate for coarseness of resolution
  - Evaluating measurement/motion based on the center of the region may not be enough. *If motion is updated every 1s, robot moves at 10 cm/s, and the grid resolution is 1m, then naïve implementation will not have any state transition!*
- Computation
  - Motion model update for a 3D grid required a 6D operation, measurement update 3D
  - With fine-grained models, the algorithm cannot be run in real-time
  - Some calculations can be cached (ray-casting results)