SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

# Bayes' Filter


CMPT 419/983

Mo Chen

SFU Computing Science


25/11/2019

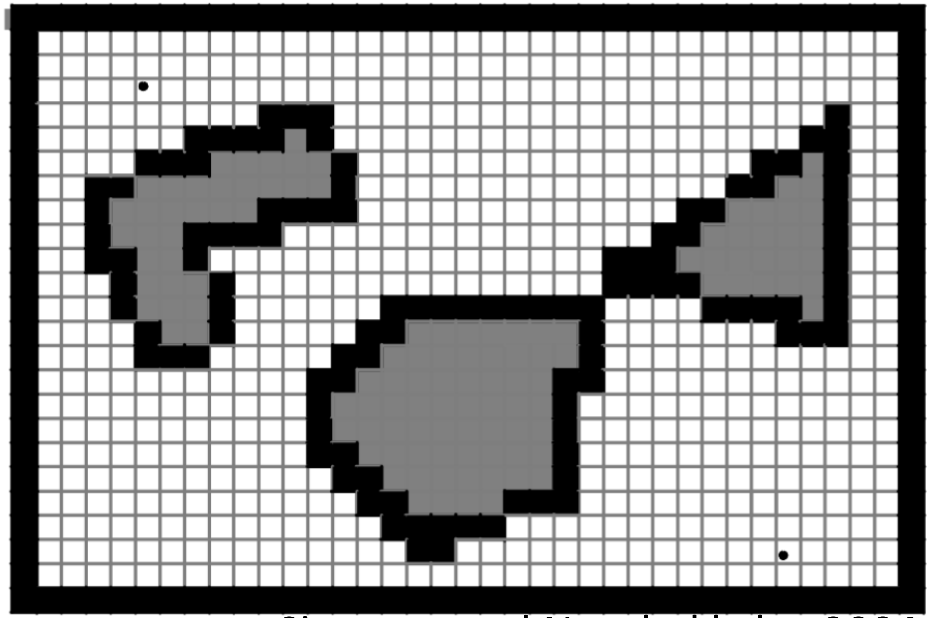
# Outline

- Localization Problem Setup

- Bayes’ Filter

# Localization: Problem Setup

- Assume a map is given: $m = \{m_1, m_2, \dots, m_N\}$
  - Location based: each $m_i$ represents a specific location and whether it's occupied (eg. Occupancy grid)
  - Feature based: each $m_i$ contains the location and properties of a feature (eg. Topological map)



Siegwart and Nourbakhshs, 2004

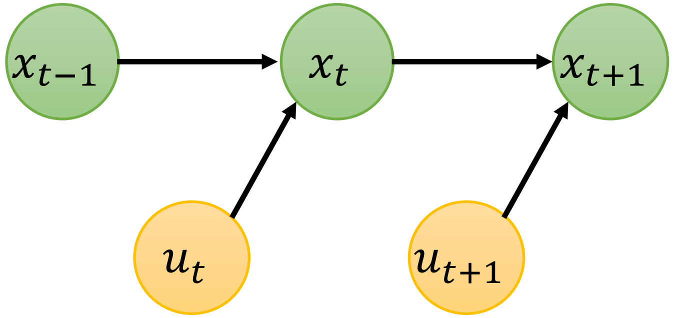
# Localization: Problem Setup

- Assume a map is given: $m = \{m_1, m_2, \dots, m_N\}$
    - Location based: each $m_i$ represents a specific location and whether it's occupied (eg. Occupancy grid)
    - Feature based: each $m_i$ contains the location and properties of a feature (eg. Topological map)

- Robot maintains and updates its belief about where it is with respect to the map
    - Position belief is updated based on sensor data
    - Position belief is a probability distribution

# Robot-Environment Interaction: Definitions

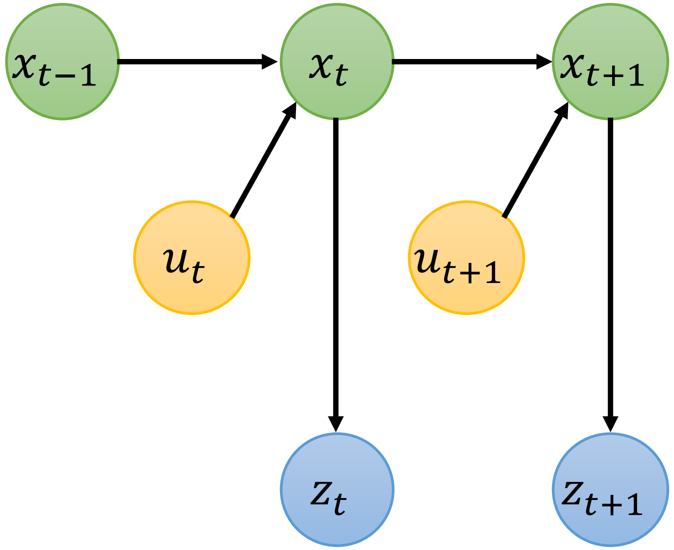
- State $x_t$: includes the environment (eg. objects, features)
  - Assume the state $x_t$ is complete / the Markov property
- Control data $u_t$
  - Usually decreases robot's knowledge
- Probabilistic model of state evolution
  - $p(x_t|x_{t-1}, u_t)$



# Robot-Environment Interaction: Definitions

- Measurement data $z_t$
  - Increases robot's knowledge

- Measurement equation:
  - $p(z_t|x_t)$



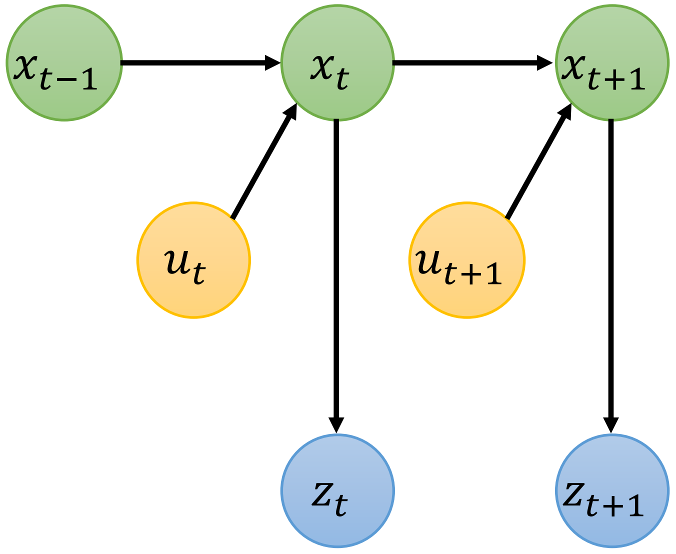
# Prediction and Belief Distributions

- Prediction distribution:
  - Robot's prediction of the state before making an observation

$$\overline{\mathrm{bel}}(x_t) \coloneqq p(x_t | z_{1:t-1}, u_{1:t})$$

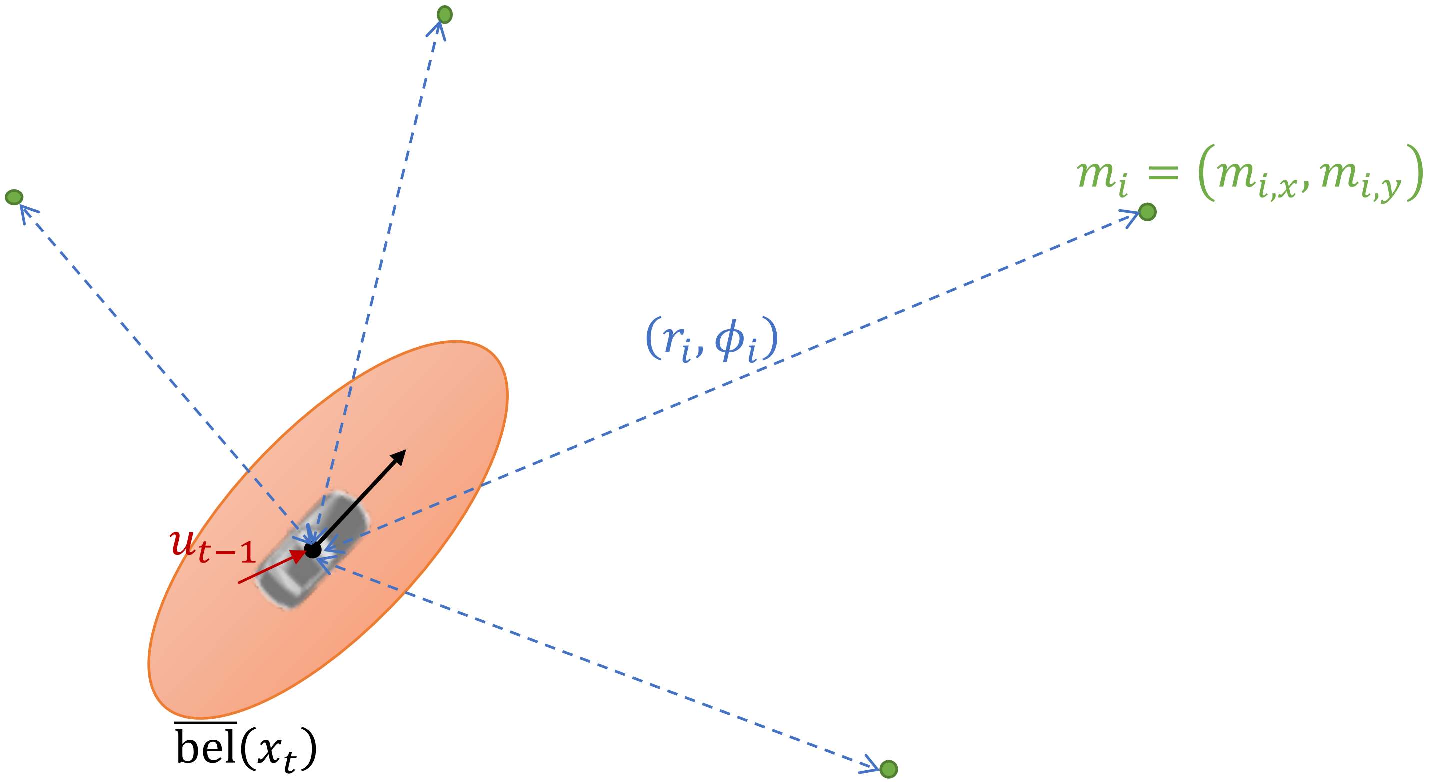
- Belief distribution:
  - Robot's internal knowledge about the state

$$\mathrm{bel}(x_t) \coloneqq p(x_t | z_{1:t}, u_{1:t})$$

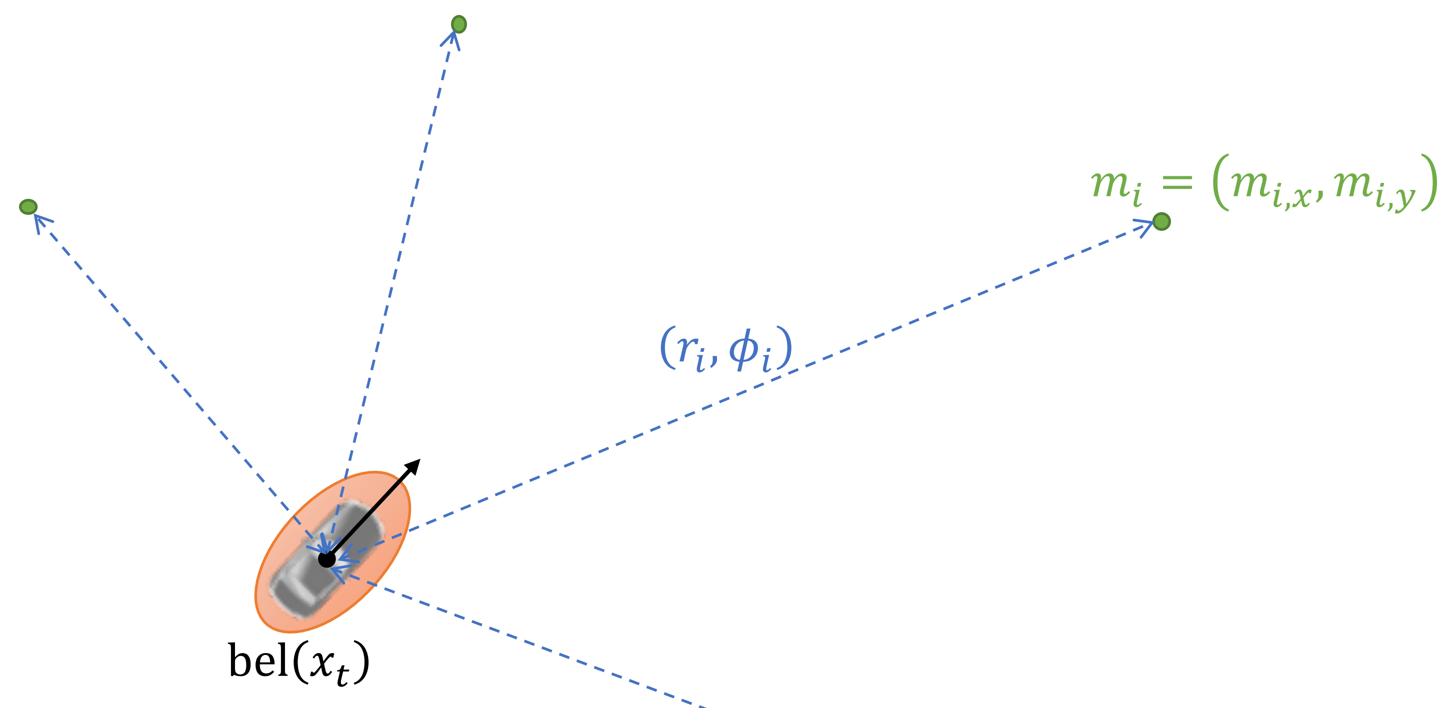


# Localization

# Localization



# Localization

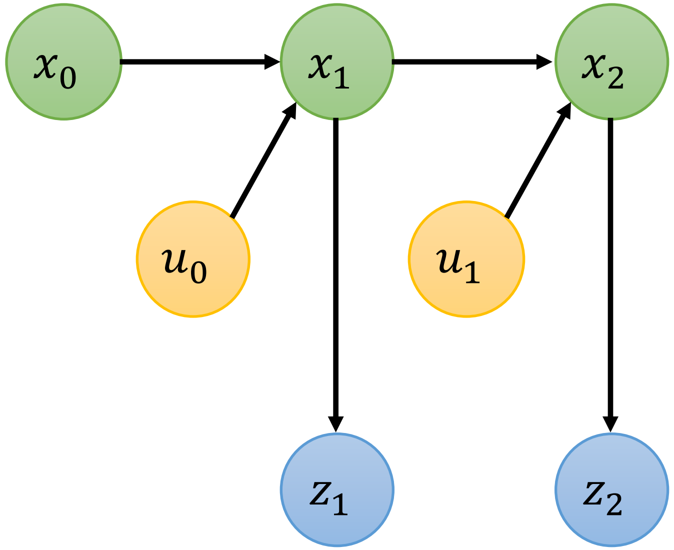


# Bayes Filter (Continuous)

- Robot and environment have state $x_0$
  - Initialize $\mathrm{bel}(x_0)$ (eg. to be uniform or dirac distribution)

- From $x_0$, choose a control $u_0$ → robot moves to $x_1$
  1. Predict the next state by computing $\overline{\mathrm{bel}}(x_1)$ using dynamics
  2. Make an observation $z_1$, and use it to compute $\mathrm{bel}(x_1)$

- Repeat for $x_2, x_3, \ldots$



# Bayes Filter (Continuous)

- Robot and environment have state $x_0$
  - Initialize $\mathrm{bel}(x_0)$ (eg. to be uniform or dirac distribution)
- From $x_0$, choose a control $u_0$ → robot moves to $x_1$
  1. Predict the next state by computing $\overline{\mathrm{bel}}(x_1)$ using dynamics
  2. Make an observation $z_1$, and use it to compute $\mathrm{bel}(x_1)$
- Repeat for $x_2, x_3, \ldots$

- Bayes' filter algorithm:

Input: $\mathrm{bel}(x_{t-1}), u_t, z_t$

Output: $\mathrm{bel}(x_t)$

For every $x_t$,

Perform prediction:

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

Perform measurement update:

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

Return $\mathrm{bel}(x_t)$

# Bayes Filter (Continuous)

$$\overline{\mathrm{bel}}(x_t) = p(x_t|z_{1:t-1}, u_{1:t})$$
$$= \int p(x_t|x_{t-1}, z_{1:t-1}, u_{1:t})p(x_{t-1}|z_{1:t-1}, u_{1:t})dx_{t-1}$$

Theorem of total probability

$$p(y) = \int p(x,y)dx = \int p(y|x)p(x)dx$$

- Bayes' filter algorithm:

Input: $\mathrm{bel}(x_{t-1}), u_t, z_t$

Output: $\mathrm{bel}(x_t)$

For every $x_t$,

Perform prediction:

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

Perform measurement update:

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

Return $\mathrm{bel}(x_t)$

# Bayes Filter (Continuous)

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

Markov assumption

- Bayes' filter algorithm:

Input: $\text{bel}(x_{t-1}), u_t, z_t$
Output: $\text{bel}(x_t)$
For every $x_t$,

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

Perform measurement update:
$\text{bel}(x_t) = \eta p(z_t|x_t)\overline{\text{bel}}(x_t)$

Return $\text{bel}(x_t)$

# Bayes Filter (Continuous)

$$\begin{aligned}\overline{\mathrm{bel}}(x_t) &= p(x_t|z_{1:t-1}, u_{1:t}) \\ &= \int p(x_t|x_{t-1}, z_{1:t-1}, u_{1:t})p(x_{t-1}|z_{1:t-1}, u_{1:t})dx_{t-1} \\ &= \int p(x_t|x_{t-1}, u_t)p(x_{t-1}|z_{1:t-1}, u_{1:t})dx_{t-1} \\ &= \int p(x_t|x_{t-1}, u_t)p(x_{t-1}|z_{1:t-1}, u_{1:t-1})dx_{t-1}\end{aligned}$$

$u_t$ does not affect probability of $x_{t-1}$

- Bayes' filter algorithm:

Input: $\mathrm{bel}(x_{t-1}), u_t, z_t$

Output: $\mathrm{bel}(x_t)$

For every $x_t$,

Perform prediction:

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

Perform measurement update:

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

Return $\mathrm{bel}(x_t)$

# Bayes Filter (Continuous)

$$\begin{aligned}\overline{\mathrm{bel}}(x_t) &= p(x_t|z_{1:t-1}, u_{1:t}) \\ &= \int p(x_t|x_{t-1}, z_{1:t-1}, u_{1:t})p(x_{t-1}|z_{1:t-1}, u_{1:t})dx_{t-1} \\ &= \int p(x_t|x_{t-1}, u_t)p(x_{t-1}|z_{1:t-1}, u_{1:t})dx_{t-1} \\ &= \int p(x_t|x_{t-1}, u_t)p(x_{t-1}|z_{1:t-1}, u_{1:t-1})dx_{t-1} \\ &= \int p(x_t|x_{t-1}, u_t)\mathrm{bel}(x_{t-1})dx_{t-1}\end{aligned}$$

- Bayes' filter algorithm:

Input: $\mathrm{bel}(x_{t-1}), u_t, z_t$

Output: $\mathrm{bel}(x_t)$

For every $x_t$,

Perform prediction:

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

Perform measurement update:

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

Return $\mathrm{bel}(x_t)$

# Bayes Filter (Continuous)

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

$$\mathrm{bel}(x_t) = p(x_t|z_{1:t}, u_{1:t})$$
$$= \frac{p(z_t|x_t, z_{1:t-1}, u_{1:t})p(x_t|z_{1:t-1}, u_{1:t})}{p(z_t|z_{1:t-1}, u_{1:t})}$$

Bayes' rule

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

- Bayes' filter algorithm:

Input: $\mathrm{bel}(x_{t-1}), u_t, z_t$
Output: $\mathrm{bel}(x_t)$
For every $x_t$,

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

Perform measurement update:
$$\mathrm{bel}(x_t) = \eta p(z_t|x_t)\overline{\mathrm{bel}}(x_t)$$

Return $\mathrm{bel}(x_t)$

# Bayes Filter (Continuous)

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

$$\text{bel}(x_t) = p(x_t|z_{1:t}, u_{1:t})$$
$$= \frac{p(z_t|x_t, z_{1:t-1}, u_{1:t})p(x_t|z_{1:t-1}, u_{1:t})}{p(z_t|z_{1:t-1}, u_{1:t})}$$
$$= \eta p(z_t|x_t)p(x_t|z_{1:t-1}, u_{1:t})$$

Markov property

- Bayes' filter algorithm:

Input: $\text{bel}(x_{t-1}), u_t, z_t$
Output: $\text{bel}(x_t)$
For every $x_t$,

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

Perform measurement update:
$\text{bel}(x_t) = \eta p(z_t|x_t)\overline{\text{bel}}(x_t)$

Return $\text{bel}(x_t)$

# Bayes Filter (Continuous)

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

$$\text{bel}(x_t) = p(x_t|z_{1:t}, u_{1:t})$$
$$= \frac{p(z_t|x_t, z_{1:t-1}, u_{1:t})p(x_t|z_{1:t-1}, u_{1:t})}{p(z_t|z_{1:t-1}, u_{1:t})}$$
$$= \eta p(z_t|x_t)p(x_t|z_{1:t-1}, u_{1:t})$$
$$= \eta p(z_t|x_t)\overline{\text{bel}}(x_t)$$

- Bayes' filter algorithm:

Input: $\text{bel}(x_{t-1}), u_t, z_t$
Output: $\text{bel}(x_t)$
For every $x_t$,

Perform prediction:

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

Perform measurement update:

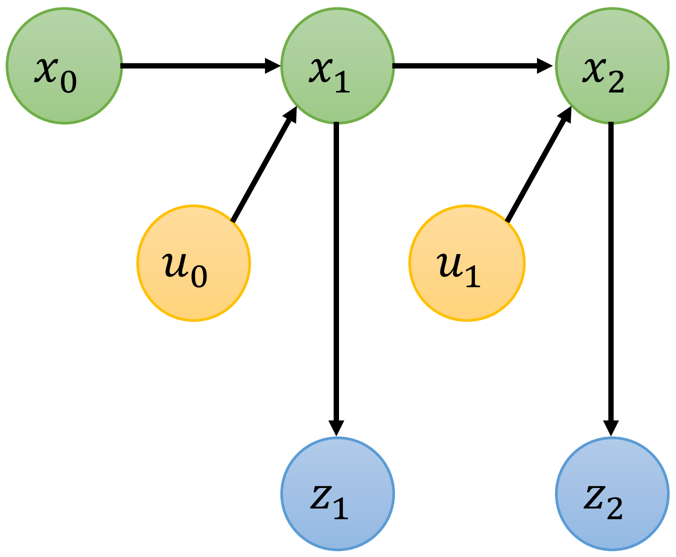
$$\text{bel}(x_t) = \eta p(z_t|x_t)\overline{\text{bel}}(x_t)$$

Return $\text{bel}(x_t)$

# Bayes Filter (Discrete)

- Robot and environment have state $x_0$
  - Initialize $\text{bel}(x_0)$ (eg. to be uniform or Dirac distribution)
- From $x_0$, choose a control $u_0$ → robot moves to $x_1$
  1. Predict the next state by computing $\overline{\text{bel}}(x_1)$
  2. Make an observation $z_1$, and use it to compute $\text{bel}(x_1)$
- Repeat for $x_2, x_3, \ldots$

- Discrete state space, $x_t$ takes on discrete values index by $k$
  - $\text{bel}(x_t)$ represented as pmf $\{p_{k,t}\}$



# Bayes Filter (Discrete)

- Robot and environment have state $x_0$
  - Initialize $\mathrm{bel}(x_0)$ (eg. to be uniform or Dirac distribution)
- From $x_0$, choose a control $u_0 \rightarrow$ robot moves to $x_1$
  1. Predict the next state by computing $\overline{\mathrm{bel}}(x_1)$
  2. Make an observation $z_1$, and use it to compute $\mathrm{bel}(x_1)$
- Repeat for $x_2, x_3, \ldots$

- Discrete state space, $x_t$ takes on discrete values index by $k$
  - $\mathrm{bel}(x_t)$ represented as pmf $\{p_{k,t}\}$

Input: $\mathrm{bel}(x_{t-1}), u_t, z_t$

Output: $\mathrm{bel}(x_t)$

For every $k$,

Perform prediction:

$\bar{p}_{k,t} = \sum p(x_t | u_t, x_{t-1}) p_{k,t-1}$

Perform measurement update:

$\mathrm{bel}(x_t) = \eta p(z_t | x_t) \bar{p}_{k,t}$

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

# Bayes Filter

- Continuous state space: Closed-form $\mathrm{bel}(x_t)$ is unlikely. Need discretization and interpolation
- Must iterate through every $x_t$ or every $k$
  - Number of states is exponential in state space dimension
- Solution: exploit structure or make assumptions
  - Parametric filters: assume a form for distributions
  - Non-parametric filters: represent distributions using samples