Bayes filtering : A deeper dive

Instructor: Chris Mavrogiannis

TAs: Kay Ke, Gilwoo Lee, Matt Schmittle

*Slides based on or adapted from Sanjiban Choudhury

Logistics

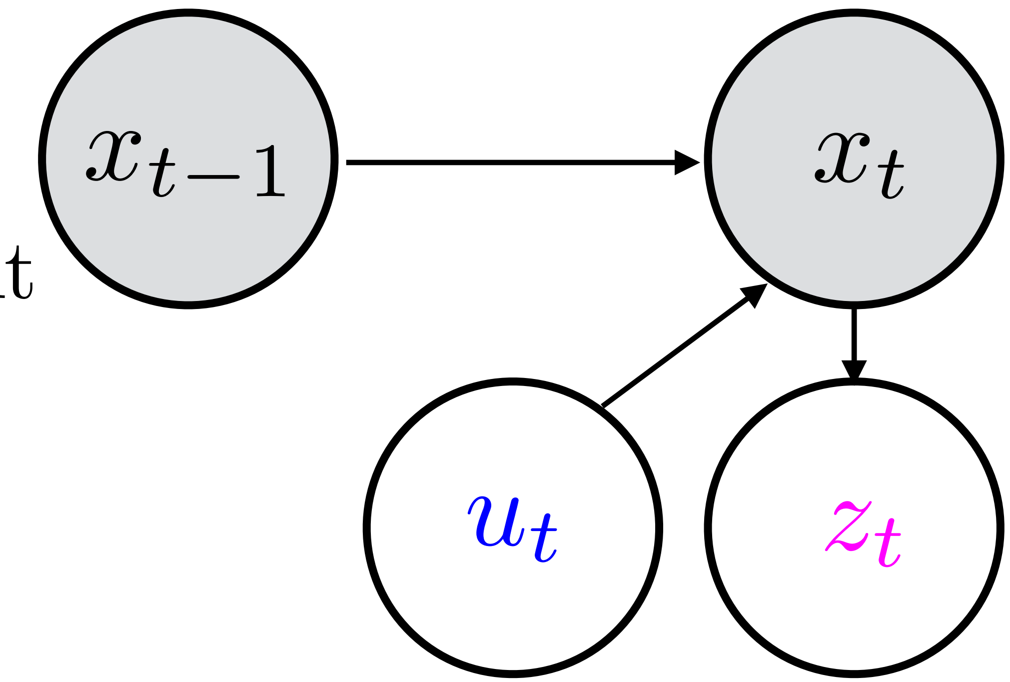
- Lab/CSE1 access
- Registering for class
- Team formation
- Probability background
- Lab0 deadline coming up!

Recap: Key players in a Bayes filter

State

"Hidden stuff we want to know"

(everything needed to predict measurement / effect of action)



New state

Measurement

"Some information relevant to state"

Action

"Affects how state evolves"

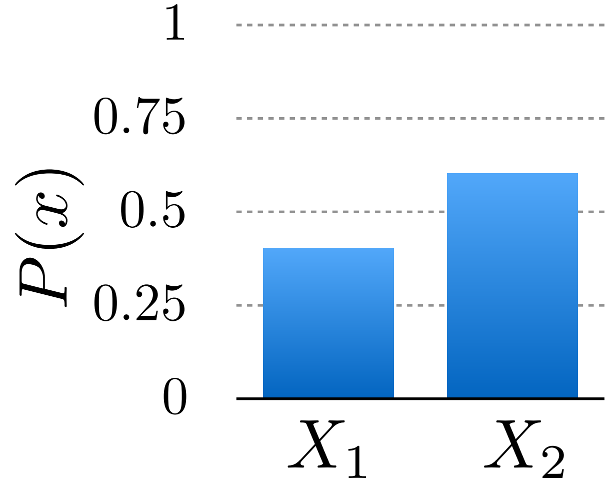
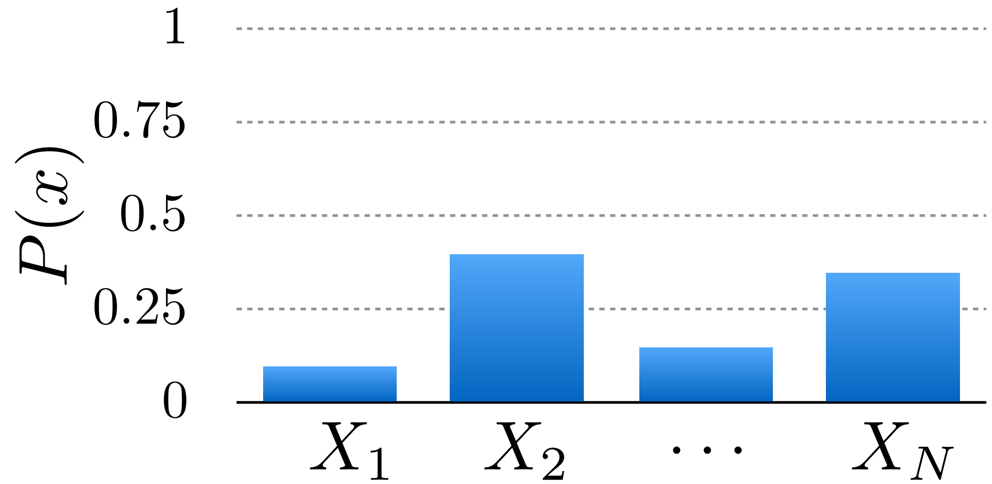
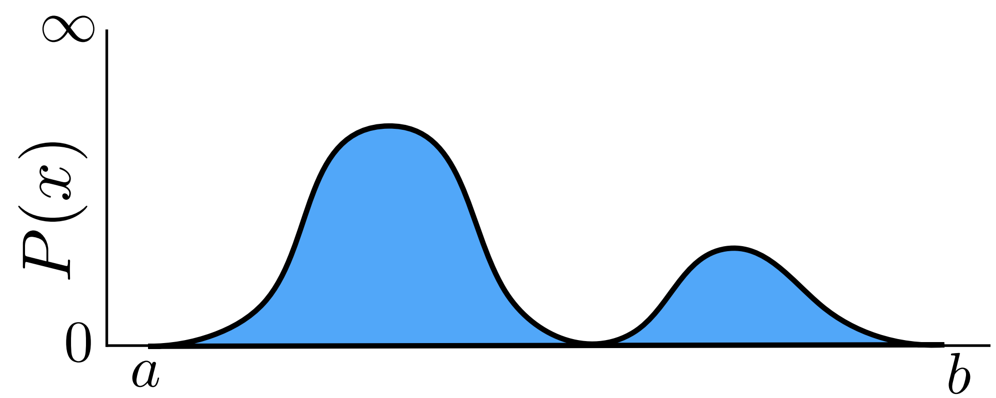
Today's objective

1. Examples of **nonparametric** Bayes filtering

2. Work through derivation

3. Question assumptions along the way

States and beliefs

State	Belief
Discrete (Binary) $X = \{X_1, X_2\}$	
Discrete (More than 2) $X = \{X_1, X_2, \ldots, X_N\}$	
Continuous $X = [a, b]$	

API of a general Bayes filter

Parameters of the Bayes filter:

Transition model: $P(x_t | x_{t-1}, u_t)$

Measurement model: $P(z_t | x_t)$

Input to the filter:

Old belief: $bel(x_{t-1})$

Action: u_t

Measurement: z_t

Output of the filter:

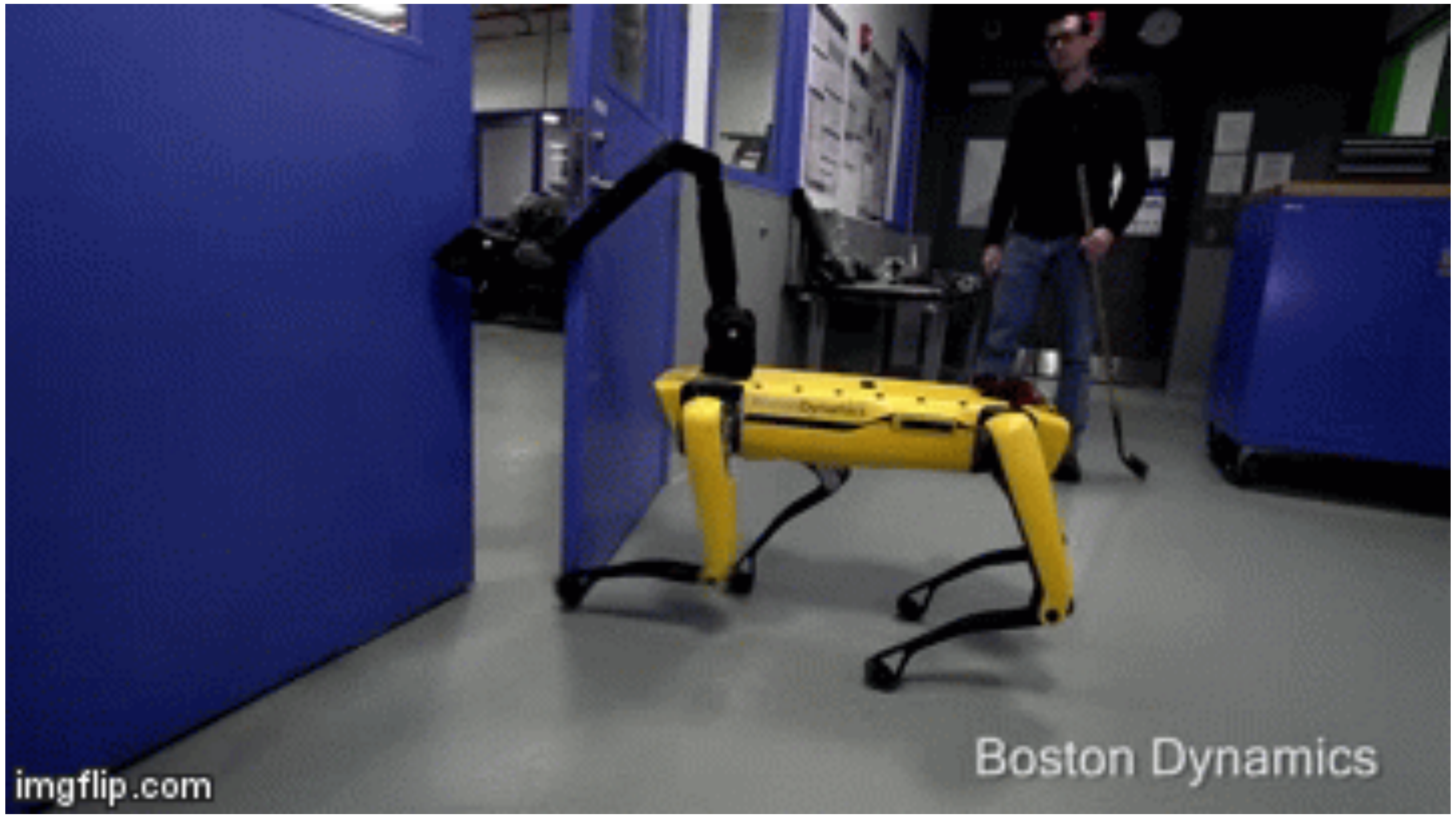
Updated belief: $bel(x_t)$

2 simple steps:

1. Predict belief after action
2. Correct belief after measurement

Discrete

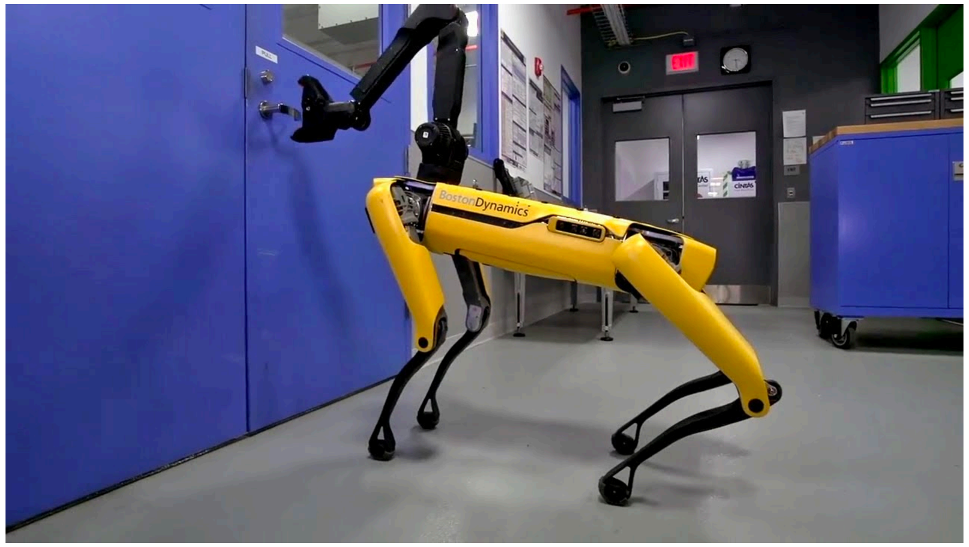
Example 1: Robot opening door



Example 1: Robot opening door

There are two states that we are tracking

X = { **O**pen, **C**losed}



Our robot can do two actions

A = { **P**ull, **L**eave}

We define a transition model (note: our robot is clumsy)

$$P(x_t | x_{t-1}, u_t)$$

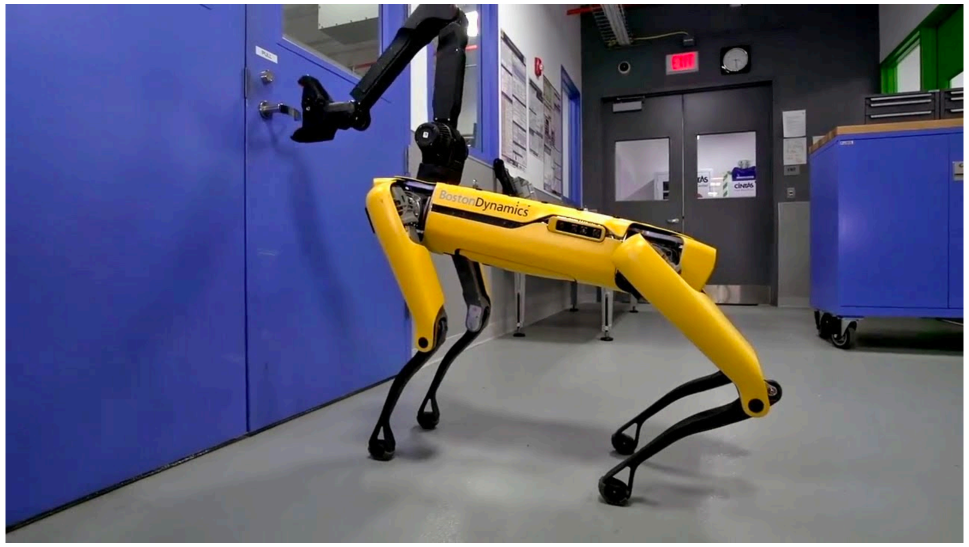
$$\mathrm{P}(\mathbf{O} \mid \mathbf{C}, \mathbf{P}) = 0.7 \qquad \mathrm{P}(\mathbf{C} \mid \mathbf{C}, \mathbf{P}) = 0.3$$

.... and so on

Example 1: Robot opening door

There are two states that we are tracking

$$X = \{ \mathbf{O}\text{pen}, \mathbf{C}\text{losed} \}$$



Our robot can do two actions

$$A = \{ \mathbf{P}\text{ull}, \mathbf{L}\text{eave} \}$$

Rewrite the transition model as a matrix

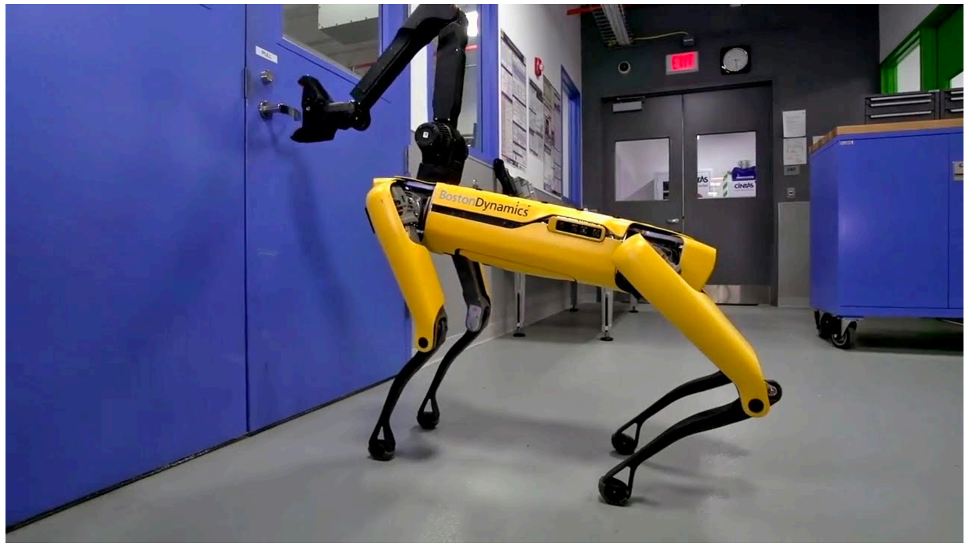
$$\begin{bmatrix} P(x_t = \mathbf{O} | x_{t-1} = \mathbf{O}, u_t) & P(x_t = \mathbf{O} | x_{t-1} = \mathbf{C}, u_t) \\ P(x_t = \mathbf{C} | x_{t-1} = \mathbf{O}, u_t) & P(x_t = \mathbf{C} | x_{t-1} = \mathbf{C}, u_t) \end{bmatrix}$$

$$P(.|., \mathbf{P}) = \begin{bmatrix} 0.8 & 0.7 \\ 0.2 & 0.3 \end{bmatrix} \qquad P(.|., \mathbf{L}) = \begin{bmatrix} 0.5 & 0 \\ 0.5 & 1 \end{bmatrix}$$

Example 1: Robot opening door

There are two states that we are tracking

X = { **O**pen, **C**losed}



Our robot can do two actions

A = { **P**ull, **L**eave}

We have a door detector sensor. The sensor is kinda buggy!

Z = { **O**pen, **C**losed}

$$P(z_t|x_t)$$

.... let's use our matrix format

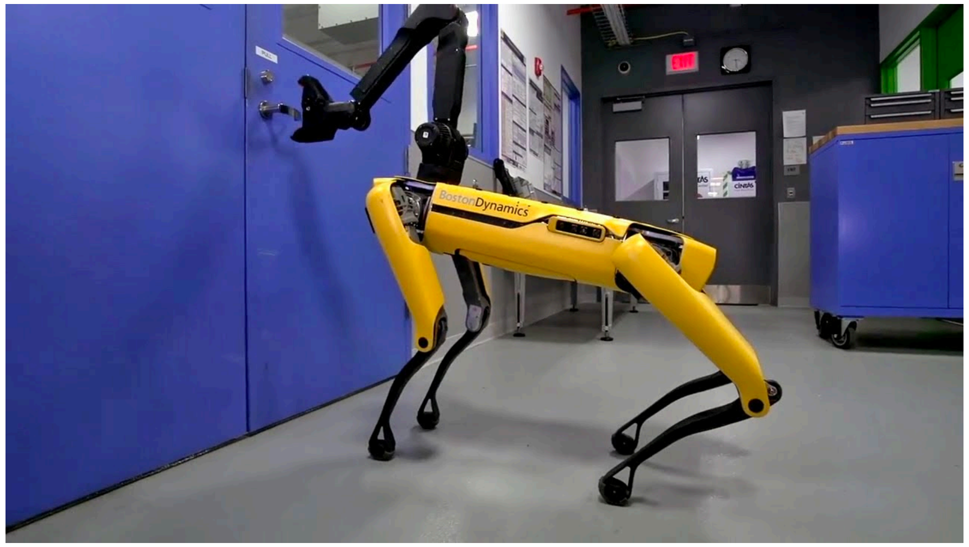
Example 1: Robot opening door

There are two states that we are tracking

$$X = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed} \}$$

$$A = \{ \textbf{P}\text{ull}, \textbf{L}\text{eave} \}$$

$$Z = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed} \}$$



Rewrite the measurement model as a vector

$$\begin{bmatrix} P(z_t|\mathbf{O}) \\ P(z_t|\mathbf{C}) \end{bmatrix}$$

$$P(\mathbf{O}|.) = \begin{bmatrix} 0.6 \\ 0.2 \end{bmatrix} \qquad P(\mathbf{C}|.) = \begin{bmatrix} 0.4 \\ 0.8 \end{bmatrix}$$

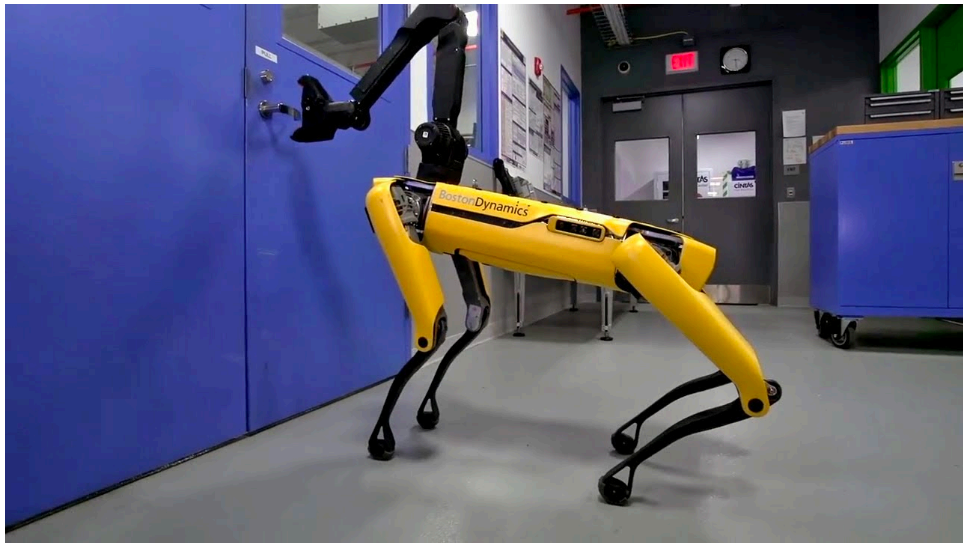
Example 1: Robot opening door

There are two states that we are tracking

X = { **O**pen, **C**losed}

A = { **P**ull, **L**eave}

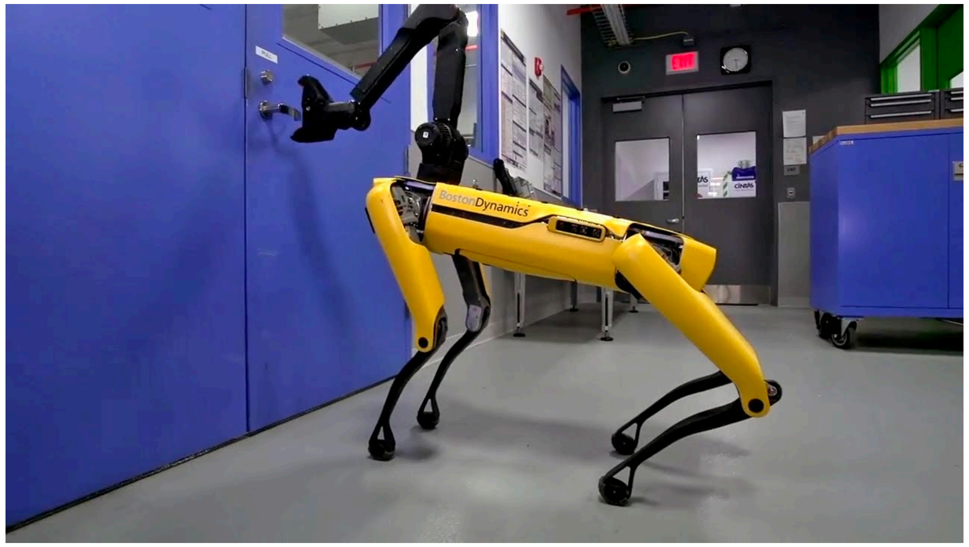
Z = { **O**pen, **C**losed}



Let's get ready to Bayes filter!

Example 1: Robot opening door

There are two states that we are tracking

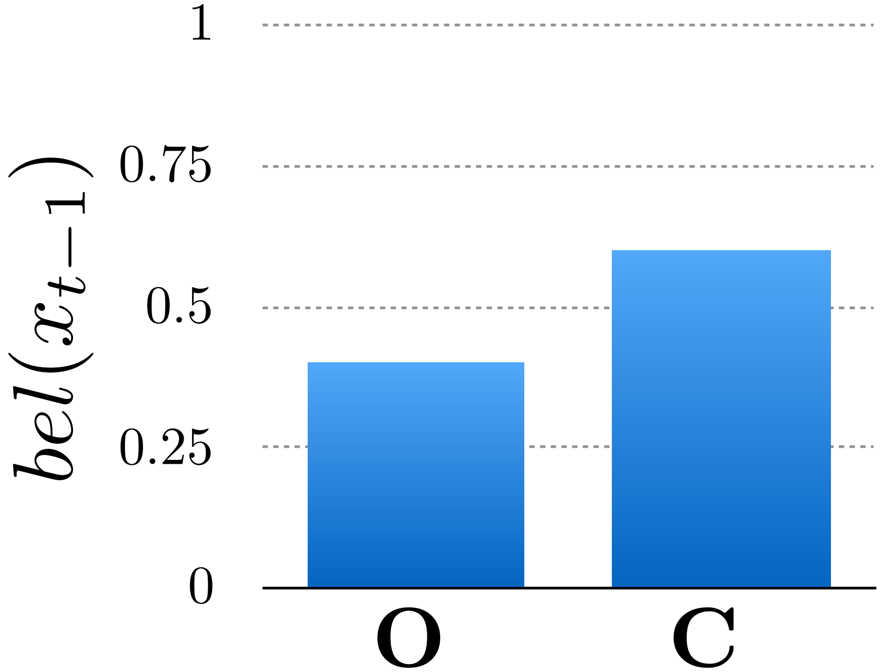


X = { **O**pen, **C**losed}

A = { **P**ull, **L**eave}

Z = { **O**pen, **C**losed}

Step 0. Start with the belief at time step t-1



$$bel(x_{t-1}) = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}$$

Robot thinks the door is open with 0.4 probability

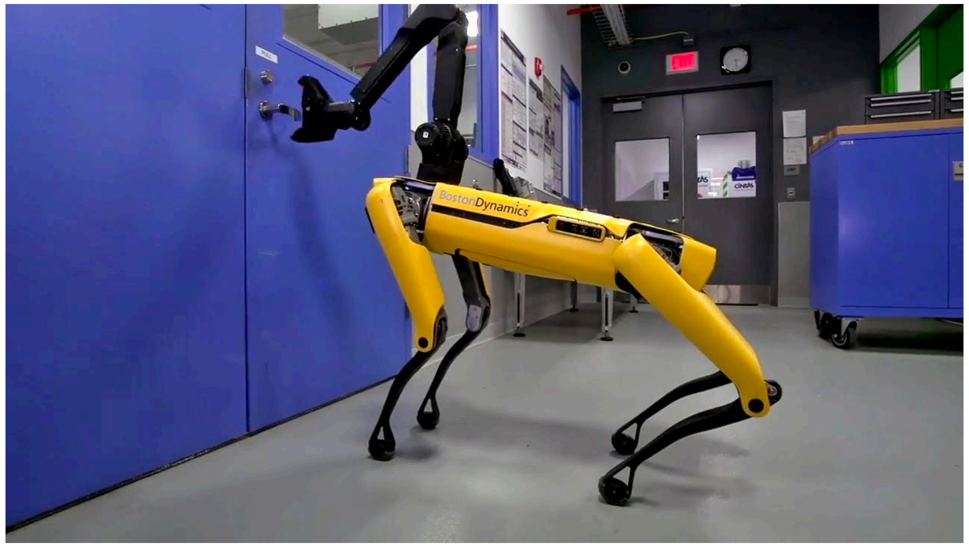
Example 1: Robot opening door

There are two states that we are tracking

$$X = \{ \mathbf{O}\text{pen}, \mathbf{C}\text{losed} \}$$

$$A = \{ \mathbf{P}\text{ull}, \mathbf{L}\text{eave} \}$$

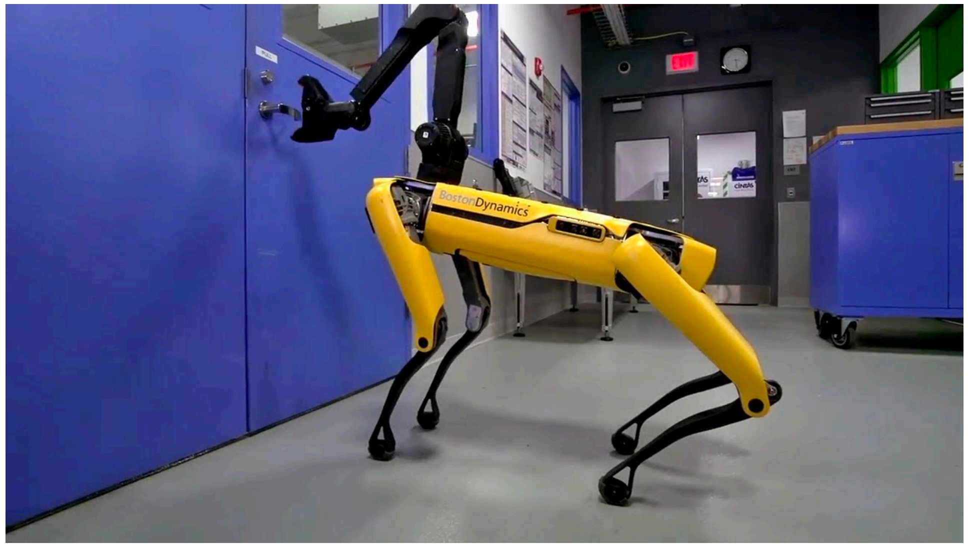
$$Z = \{ \mathbf{O}\text{pen}, \mathbf{C}\text{losed} \}$$



Robot executes action **Pull**

Example 1: Robot opening door

There are two states that we are tracking



X = { **O**pen, **C**losed}

A = { **P**ull, **L**eave}

Z = { **O**pen, **C**losed}

Step 1: Prediction - push belief through dynamics given action

$$\overline{bel}(x_t) = \sum_{x_{t-1}} P(x_t | x_{t-1}, u_t) bel(x_{t-1})$$

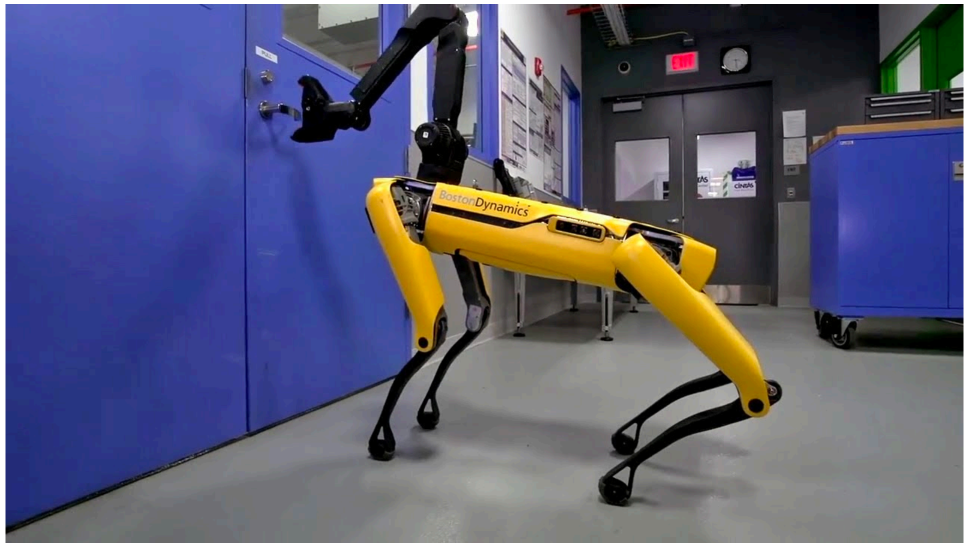
Example 1: Robot opening door

There are two states that we are tracking

$$X = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed} \}$$

$$A = \{ \textbf{P}\text{ull}, \textbf{L}\text{eave} \}$$

$$Z = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed} \}$$

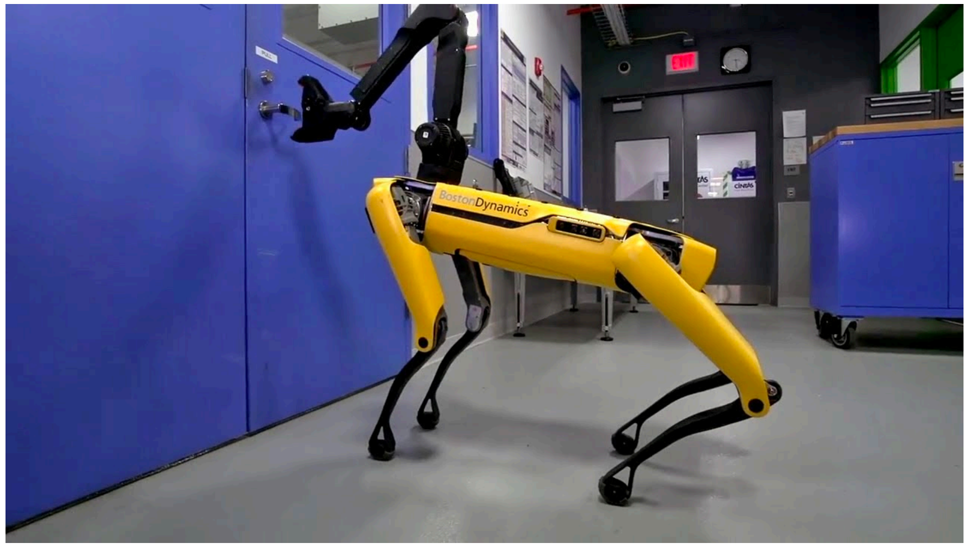


Step 1: Prediction - push belief through dynamics given action

$$\underbrace{\begin{bmatrix} P(x_t = \mathbf{O}) \\ P(x_t = \mathbf{C}) \end{bmatrix}}_{\overline{bel}(x_t)} = \begin{bmatrix} P(x_t = \mathbf{O} | x_{t-1} = \mathbf{O}, u_t) & P(x_t = \mathbf{O} | x_{t-1} = \mathbf{C}, u_t) \\ P(x_t = \mathbf{C} | x_{t-1} = \mathbf{O}, u_t) & P(x_t = \mathbf{C} | x_{t-1} = \mathbf{C}, u_t) \end{bmatrix} \underbrace{\begin{bmatrix} P(x_{t-1} = \mathbf{O}) \\ P(x_{t-1} = \mathbf{C}) \end{bmatrix}}_{bel(x_{t-1})}$$

Example 1: Robot opening door

There are two states that we are tracking



$$X = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed} \}$$

$$A = \{ \textbf{P}\text{ull}, \textbf{L}\text{eave} \}$$

$$Z = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed} \}$$

Step 1: Prediction - push belief through dynamics given action

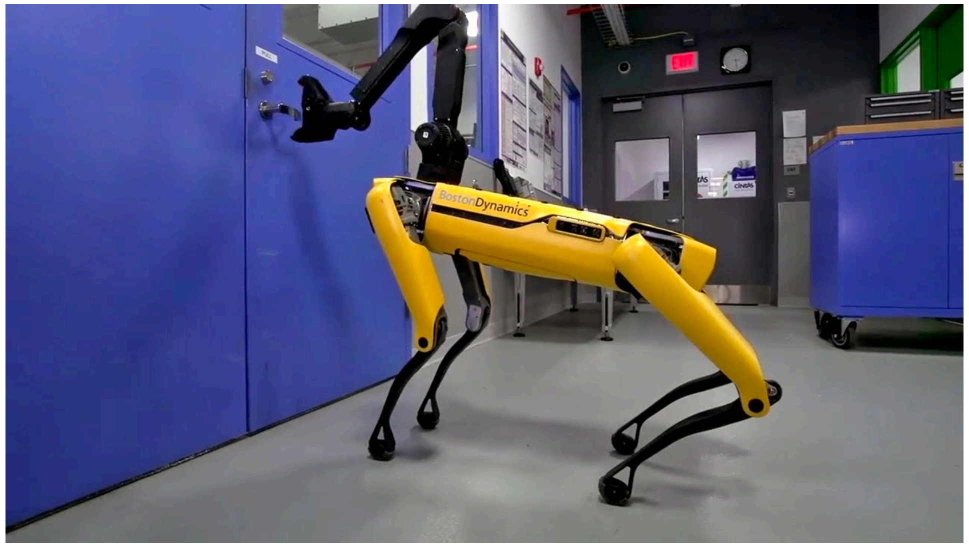
$$\begin{bmatrix} 0.74 \\ 0.26 \end{bmatrix} = \begin{bmatrix} 0.8 & 0.7 \\ 0.2 & 0.3 \end{bmatrix} \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}$$

$\overline{bel}(x_t)$ $\quad P(.|., \mathbf{P})$ $\quad bel(x_{t-1})$

Robot thinks the door is open with 0.74 probability

Example 1: Robot opening door

There are two states that we are tracking



$$X = \{\, \textbf{O}\text{pen},\ \textbf{C}\text{losed}\}$$

$$A = \{\, \textbf{P}\text{ull},\ \textbf{L}\text{eave}\}$$

$$Z = \{\, \textbf{O}\text{pen},\ \textbf{C}\text{losed}\}$$

Robot receives measurement

Closed

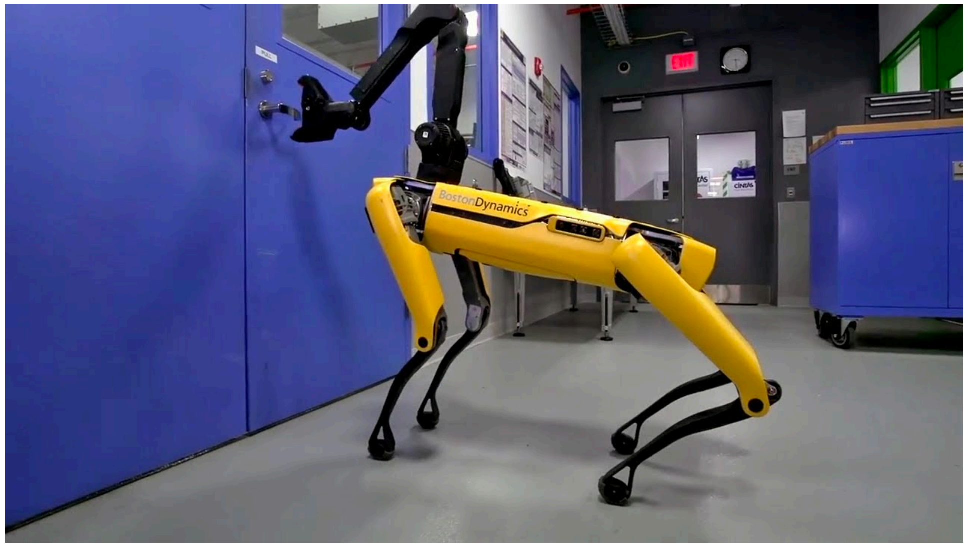
Example 1: Robot opening door

There are two states that we are tracking

X = { **O**pen, **C**losed}

A = { **P**ull, **L**eave}

Z = { **O**pen, **C**losed}



Step 2: Correction - apply Bayes rule given measurement

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

(normalize)

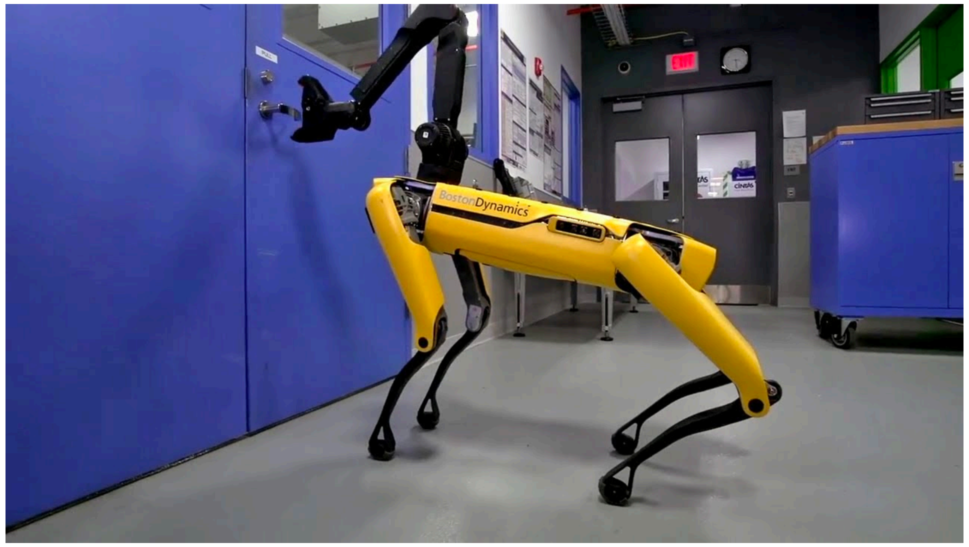
Example 1: Robot opening door

There are two states that we are tracking

$$X = \{ \mathbf{O}\text{pen}, \mathbf{C}\text{losed}\}$$

$$A = \{ \mathbf{P}\text{ull}, \mathbf{L}\text{eave}\}$$

$$Z = \{ \mathbf{O}\text{pen}, \mathbf{C}\text{losed}\}$$



Step 2: Correction - apply Bayes rule given measurement

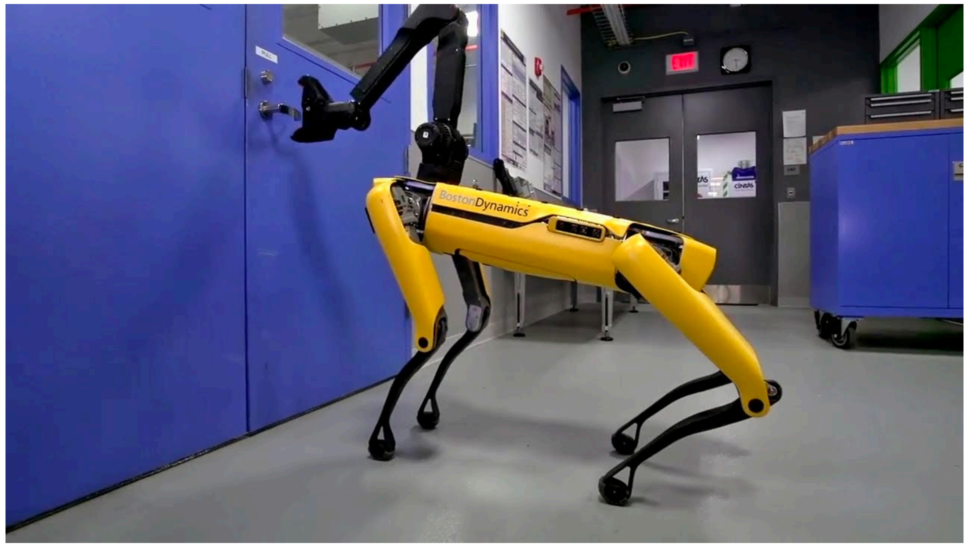
$$\begin{bmatrix} P(x_t = \mathbf{O}) \\ P(x_t = \mathbf{C}) \end{bmatrix} = \eta \begin{bmatrix} P(z_t | \mathbf{O}) \\ P(z_t | \mathbf{C}) \end{bmatrix} * \begin{bmatrix} P(x_t = \mathbf{O}) \\ P(x_t = \mathbf{C}) \end{bmatrix}$$

$bel(x_t)$ $P(\mathbf{C}|.)$ $\overline{bel}(x_t)$

element wise

Example 1: Robot opening door

There are two states that we are tracking



$$X = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed}\}$$

$$A = \{ \textbf{P}\text{ull}, \textbf{L}\text{eave}\}$$

$$Z = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed}\}$$

Step 2: Correction - apply Bayes rule given measurement

$$\begin{bmatrix} P(x_t = \mathbf{O}) \\ P(x_t = \mathbf{C}) \end{bmatrix} = \eta \quad \begin{bmatrix} 0.4 \\ 0.8 \end{bmatrix} \quad \begin{bmatrix} 0.74 \\ 0.26 \end{bmatrix}$$

$bel(x_t)$ $\qquad$ $\overline{bel}(x_t)$

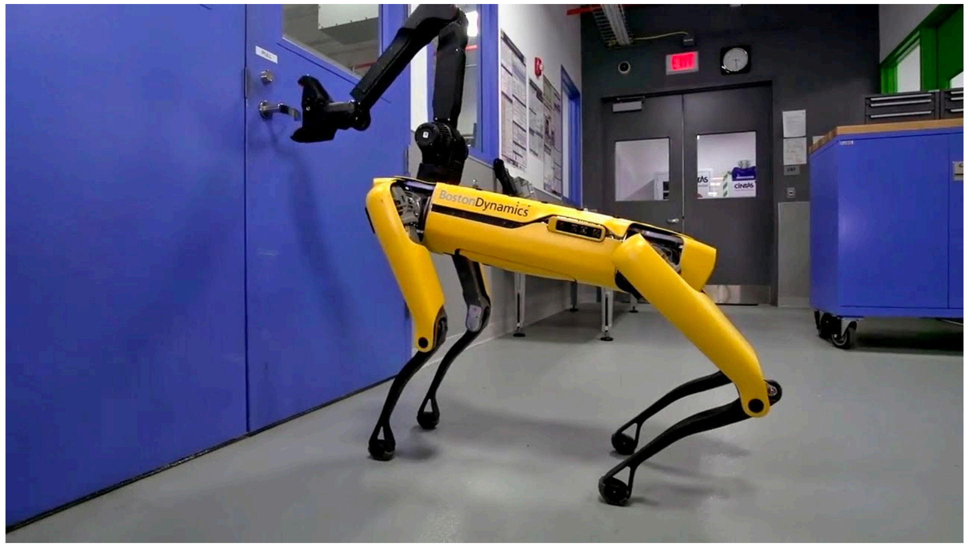
Example 1: Robot opening door

There are two states that we are tracking

$$X = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed} \}$$

$$A = \{ \textbf{P}\text{ull}, \textbf{L}\text{eave} \}$$

$$Z = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed} \}$$



Step 2: Correction - apply Bayes rule given measurement

$$\begin{bmatrix} P(x_t = \mathbf{O}) \\ P(x_t = \mathbf{C}) \end{bmatrix} = \eta \begin{bmatrix} 0.4 \\ 0.8 \end{bmatrix} \begin{bmatrix} 0.74 \\ 0.26 \end{bmatrix} = \eta \begin{bmatrix} 0.296 \\ 0.208 \end{bmatrix} = \begin{bmatrix} 0.58 \\ 0.42 \end{bmatrix}$$

$bel(x_t)$ $\qquad$ $\overline{bel}(x_t)$

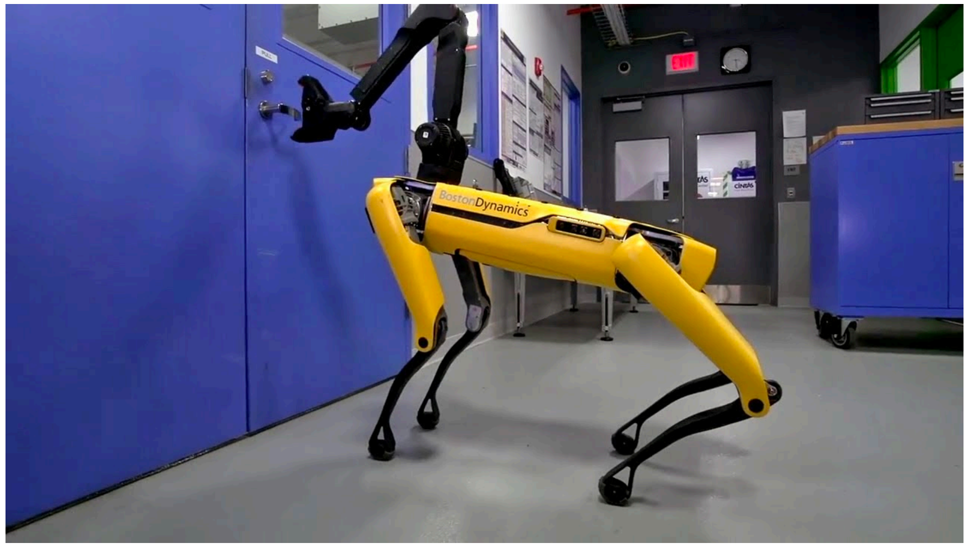
Example 1: Robot opening door

There are two states that we are tracking

X = { **O**pen, **C**losed}

A = { **P**ull, **L**eave}

Z = { **O**pen, **C**losed}



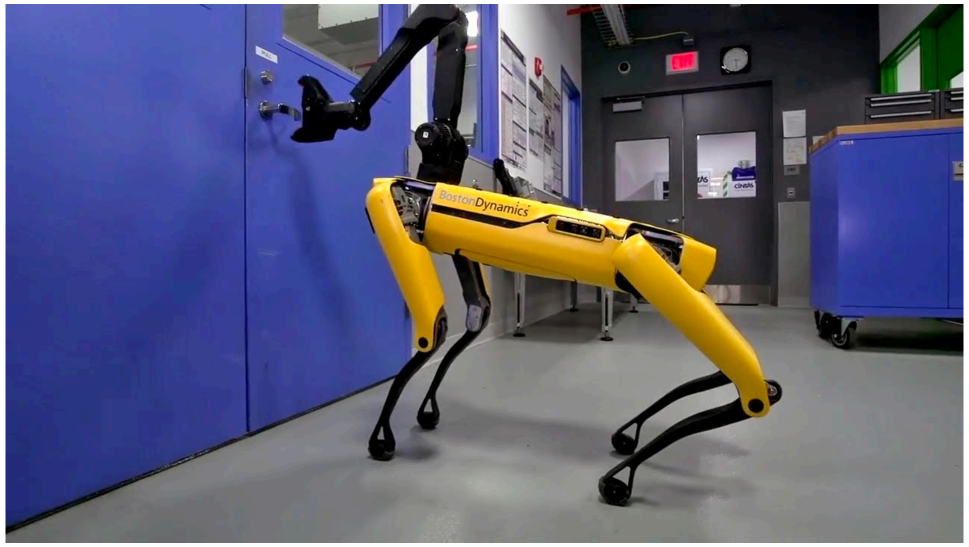
Step 2: Correction - apply Bayes rule given measurement

$$bel(x_t) = \begin{bmatrix} 0.58 \\ 0.42 \end{bmatrix}$$

Robot thinks the door is open with 0.58 probability

Example 1: Robot opening door

There are two states that we are tracking



$$X = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed} \}$$

$$A = \{ \textbf{P}\text{ull}, \textbf{L}\text{eave} \}$$

$$Z = \{ \textbf{O}\text{pen}, \textbf{C}\text{losed} \}$$

Let's summarize

Robot thought the door is open with 0.4 probability

Robot executed Pull action.
Robot thinks the door is open with 0.74 probability

Robot got Closed measurement.
Robot thinks the door is open with 0.58 probability

Continuous

Bayes filter in a nutshell

Step 0. Start with the belief at time step t-1

$$bel(x_{t-1})$$

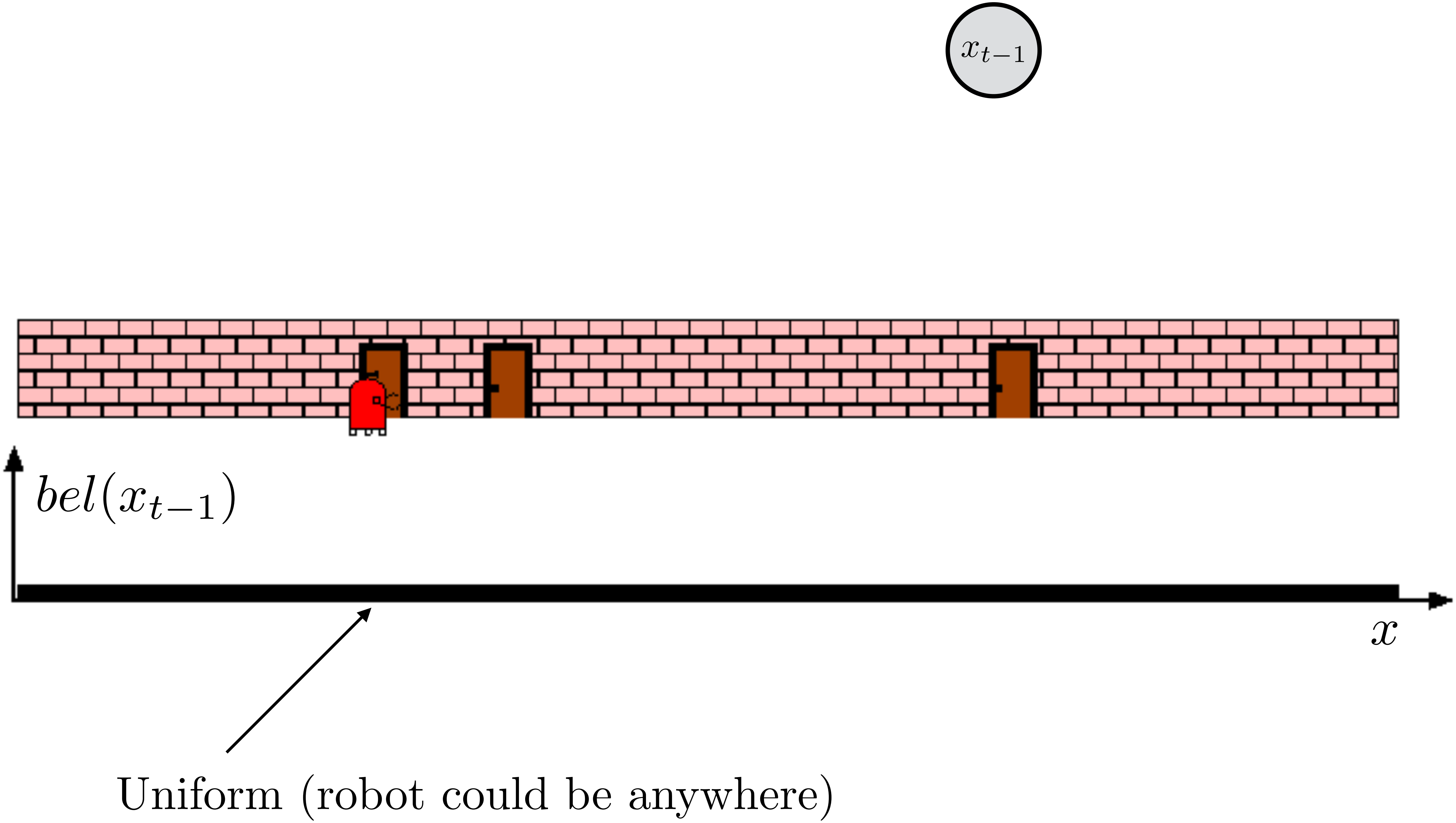
Step 1: Prediction - push belief through dynamics given action

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

Step 2: Correction - apply Bayes rule given measurement

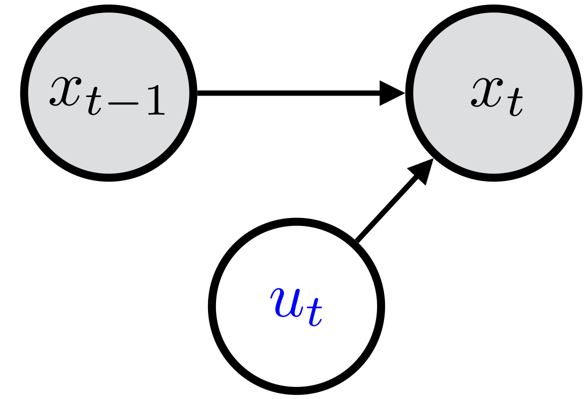
$$bel(x_t) = \eta P(z_t|x_t) \overline{bel}(x_t)$$

Robot lost in a 1-D hallway

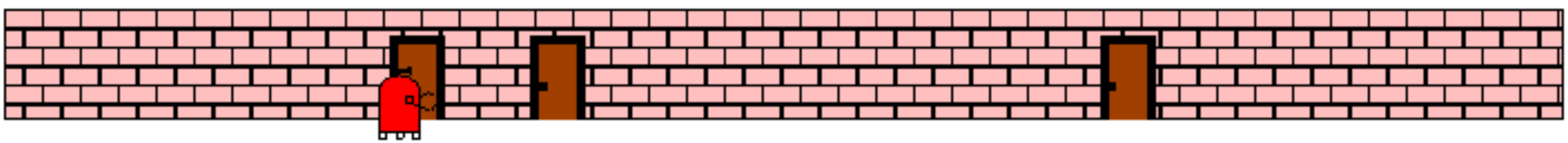


Action at time t: NOP

$u_t = \text{NOP}$



$$P(x_t | u_t, x_{t-1}) = \delta(x_t = x_{t-1})$$



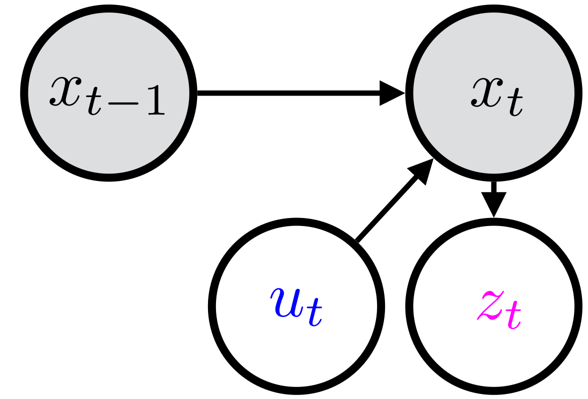
$$\overline{bel}(x_t) = \int P(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} = bel(x_t)$$

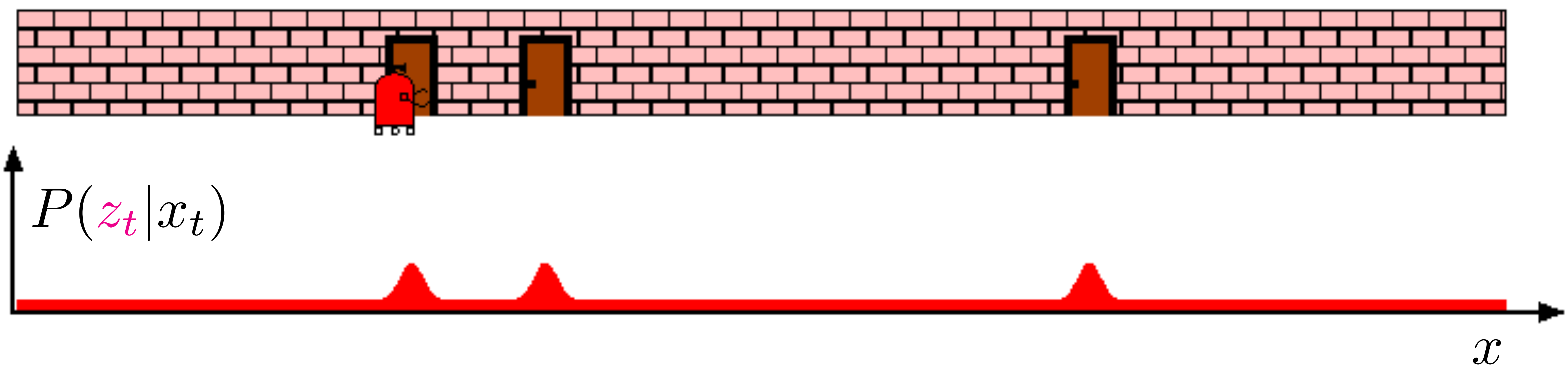
NOP action implies belief remains the same!
(still uniform — no idea where I am)

Measurement at time t: “Door”

$z_t = \text{Door}$

$P(z_t|x_t) = \mathcal{N}(\text{door centre}, 0.75m)$

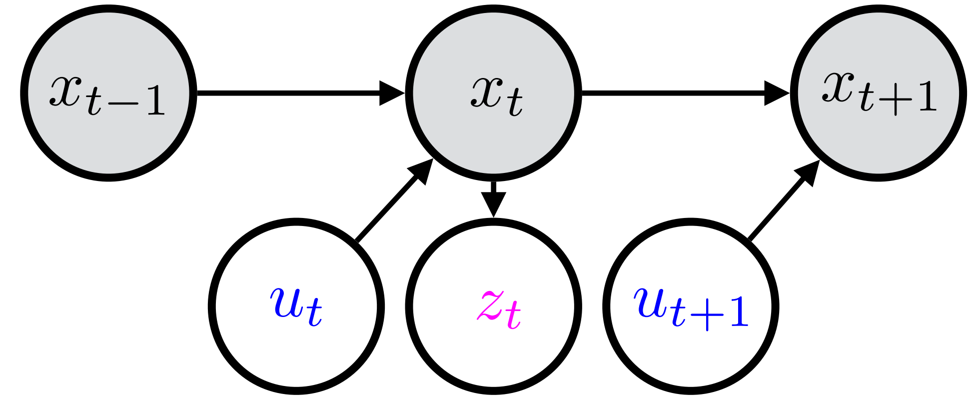


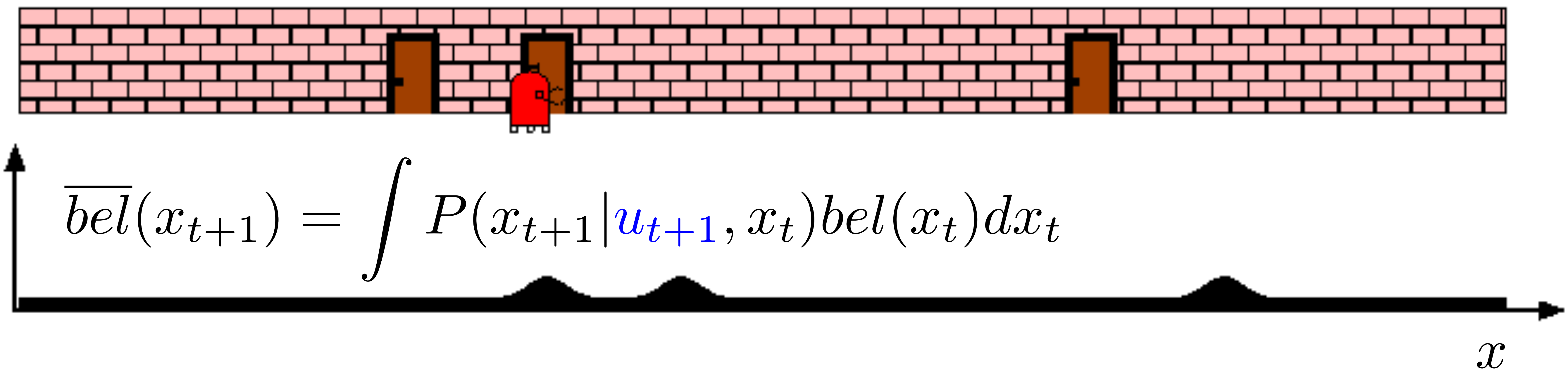


Action at time t+1: Move 3m right

u_{t+1} = 3m right

$$P(x_{t+1}|u_{t+1}, x_t) = \mathcal{N}(x_t + u_{t+1}, 0.25m)$$



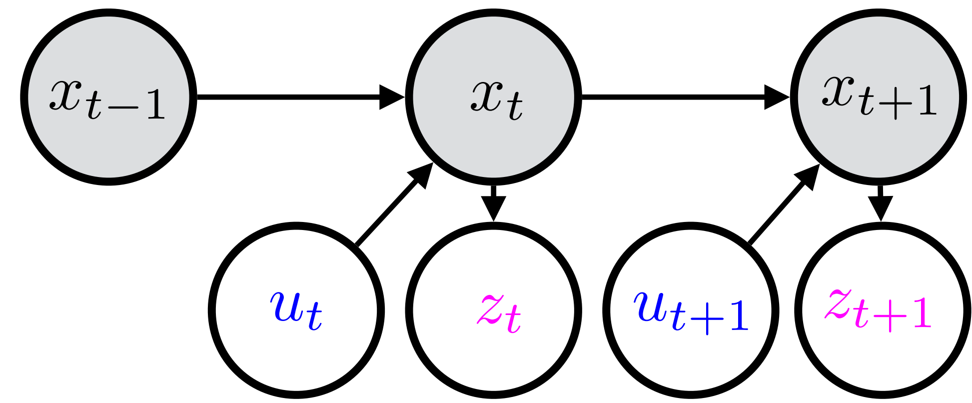


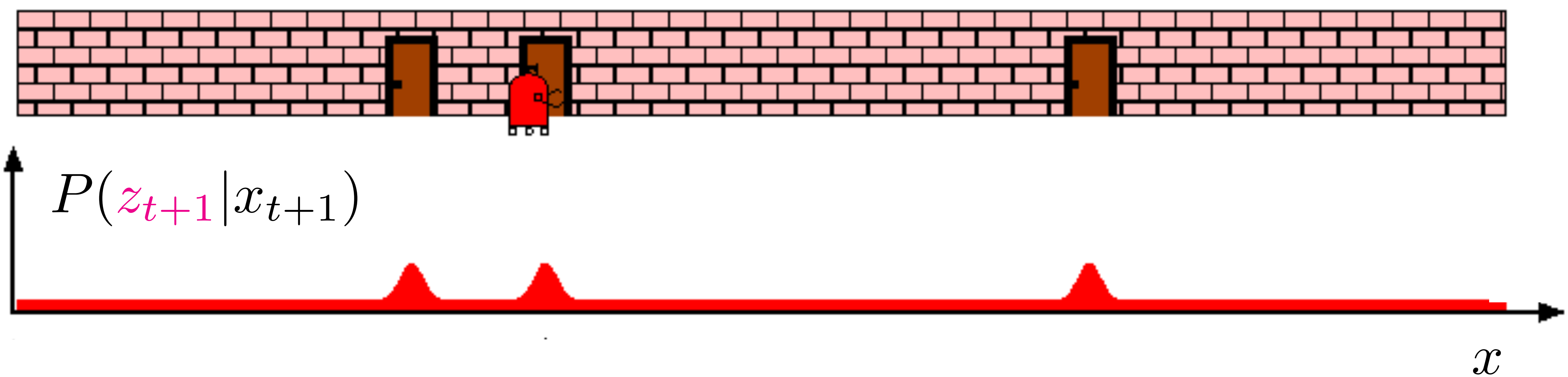
$$\overline{bel}(x_{t+1}) = \int P(x_{t+1}|u_{t+1}, x_t) bel(x_t) dx_t$$

Measurement at time t+1: “Door”

$z_{t+1} = \text{Door}$

$P(z_{t+1}|x_{t+1}) = \mathcal{N}(\text{door centre}, 0.75m)$





Exercise: Discrete bayes filter

Step 1: Prediction - push belief through dynamics given action

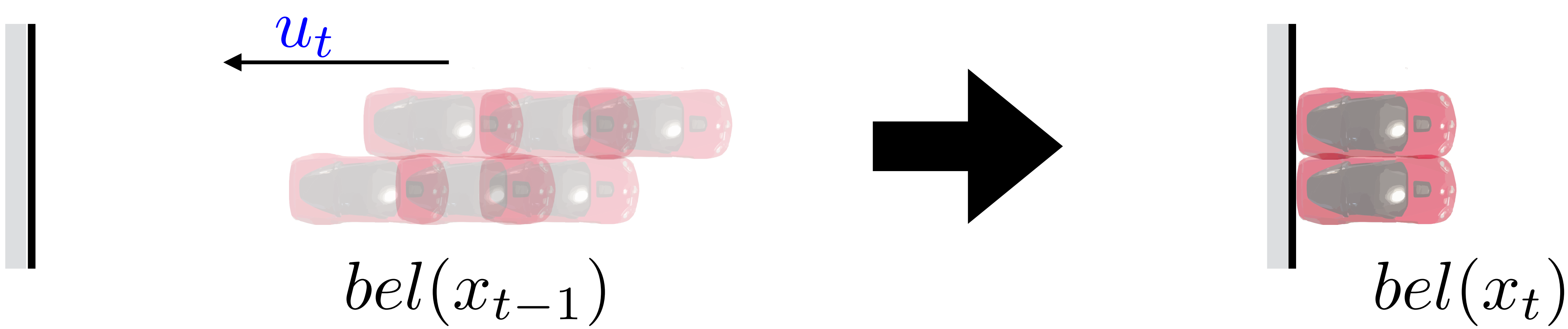
$$\begin{bmatrix} P(\overline{x}_t = 1) \\ \vdots \\ P(\overline{x}_t = n) \end{bmatrix} = \begin{bmatrix} P(x_t = 1 | u_t, x_{t-1} = 1) & \cdots & P(x_t = 1 | u_t, x_{t-1} = n) \\ \vdots & \ddots & \vdots \\ P(x_t = n | u_t, x_{t-1} = 1) & \cdots & P(x_t = n | u_t, x_{t-1} = n) \end{bmatrix} \begin{bmatrix} P(x_{t-1} = 1) \\ \vdots \\ P(x_{t-1} = n) \end{bmatrix}$$

Step 2: Correction - apply Bayes rule given measurement

$$\begin{bmatrix} P(x_t = 1) \\ P(x_t = 2) \\ \vdots \\ P(x_t = n) \end{bmatrix} = \begin{bmatrix} P(z_t | x_t = 1) & 0 & \cdots & 0 \\ 0 & P(z_t | x_t = 2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P(z_t | x_t = n) \end{bmatrix} \begin{bmatrix} P(x_{t-1} = 1) \\ P(x_{t-1} = 2) \\ \vdots \\ P(x_{t-1} = n) \end{bmatrix}$$

Questions

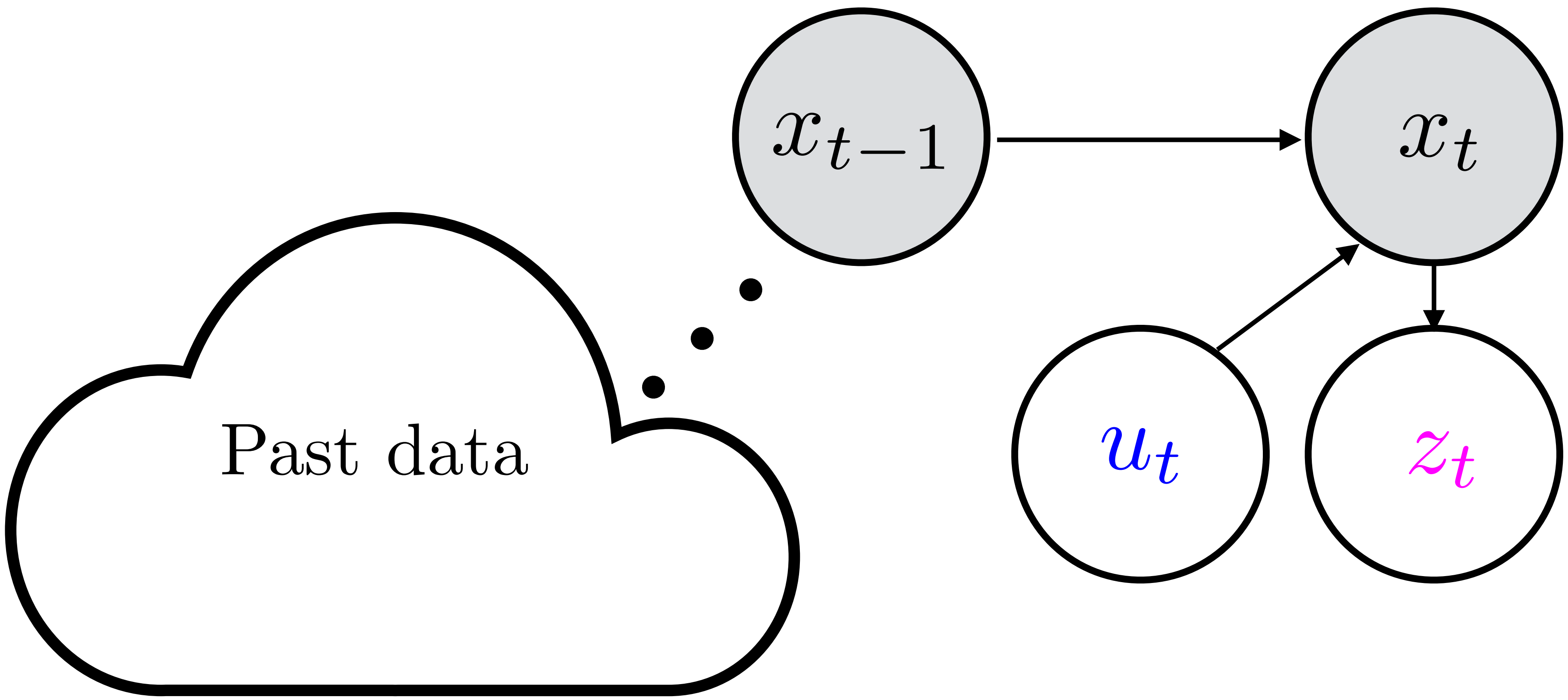
Do actions always increase uncertainty?



Do measurements always reduce uncertainty?

(What happens when you reach into your
bag and don't find your keys?
Example of a negative measurement)

Bayes derivation



Bayes derivation

$$bel(x_t) = P(x_t | z_{1:t-1}, u_{1:t-1}, z_t, u_t)$$

← Incorporate new action, measurement

(Bayes) $$= \eta P(z_t | x_t, z_{1:t-1}, u_{1:t-1}, u_t) \; P(x_t | z_{1:t-1}, u_{1:t-1}, u_t)$$

$P(A|B) = \eta P(B|A) P(A)$

Get rid of this ($z_{1:t-1}, u_{1:t-1}, u_t$)

(Markov) $$= \eta P(z_t | x_t) \; P(x_t | z_{1:t-1}, u_{1:t-1}, u_t)$$

$$= \eta P(z_t | x_t) \; \overline{bel}(x_t)$$

Bayes derivation

$$\overline{bel}(x_t) = P(x_t | z_{1:t-1}, u_{1:t-1}, u_t)$$

(Total prob.) $$= \int P(x_t | x_{t-1}, \underline{z_{1:t-1}, u_{1:t-1}}, u_t)\, P(x_{t-1} | z_{1:t-1}, u_{1:t-1}, u_t) dx_{t-1}$$

Get rid of this

(Markov) $$= \int P(x_t | x_{t-1}, u_t)\, P(x_{t-1} | z_{1:t-1}, u_{1:t-1}, \cancel{u_t}) dx_{t-1}$$

(Cond. indep) $$= \int P(x_t | x_{t-1}, u_t)\, \boxed{P(x_{t-1} | z_{1:t-1}, u_{1:t-1})} dx_{t-1}$$

Previous Belief!

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

After thoughts ...

Question: When is cond. independence not true?

$$= \int P(x_t | x_{t-1}, u_t) \; P(x_{t-1} | z_{1:t-1}, u_{1:t-1}, u_t) dx_{t-1}$$

(Cond. indep) $$= \int P(x_t | x_{t-1}, u_t) \; P(x_{t-1} | z_{1:t-1}, u_{1:t-1}) dx_{t-1}$$

i.e. when can you tell something about the past
based on future data?

E.g. Motion capture data of a human.
Human knows the true state and generates control actions accordingly.

Bayes filter in a single line

$$P(x_t|x_{t-1}, u_t)$$

Motion model

$$P(z_t|x_t)$$

Measurement model

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

Note that order does not really matter -
we can flip measurement and control.

Bayes Filter Pseudocode (Asynchronous)

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

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

h=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)*

Slide by Prof. Dieter Fox

Practical Issues

1. Bayes filter can be overconfident

Once belief collapses to 0/1 only motion model can shake it loose

2. Too many measurements will collapse belief

3. Correlated incorrect measurements are dangerous

Bayes filter is a powerful tool

Localization Mapping SLAM POMDP

This Week

1. Motion Models (Wed)
2. Measurement Models (Fri)