Lecture 11: Filtering I

Professor Katie Driggs-Campbell

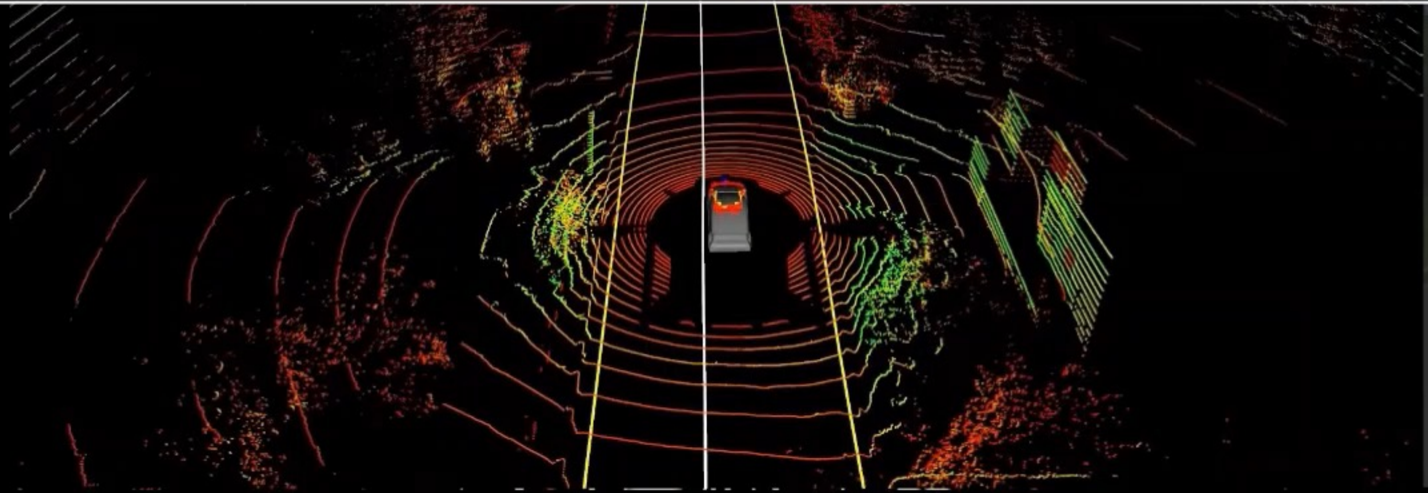
February 27, 2024

ECE484: Principles of Safe Autonomy

Administrivia

- Any volunteers to swap from GEM to another project?
- Upcoming due dates:
  - HW2 and MP2 due Friday 3/01
  - HW3 and MP3 due
  - Project Pitches in class 3/05 and 3/07
    - Presentation template on website
    - Sign-up for ordering will be posted later today (1 extra point for going on Tuesday)
- Exam has been moved to 4/18 at 7pm in 1013 and 1015 ECEB
  - If you have DRES accommodations, please make an appointment at the testing center at this same time

Navigating Intersections

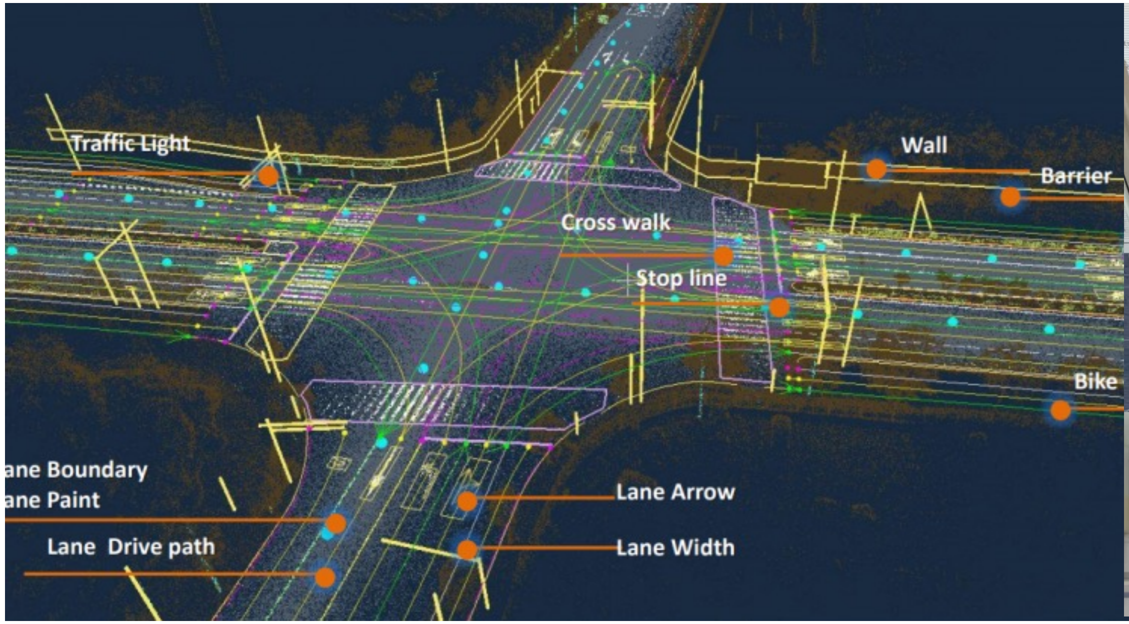


SAIC

SISL
Stanford Intelligent
Systems Laboratory

Filtering and Localization Use Cases

HD Maps



Object Tracking

Today's Plan

- What is filtering, mapping, and localization?
  - Probability review!
- Bayes Filters (discrete)

Today's Plan

- What is filtering, mapping, and localization?
  - Probability review!
- Bayes Filters (discrete)

Robot States and the Environment

- **State** represents the environment as well as the robot, for example:
  - location of walls or objects (environment or static)
  - pose of the robot (physical or dynamical)

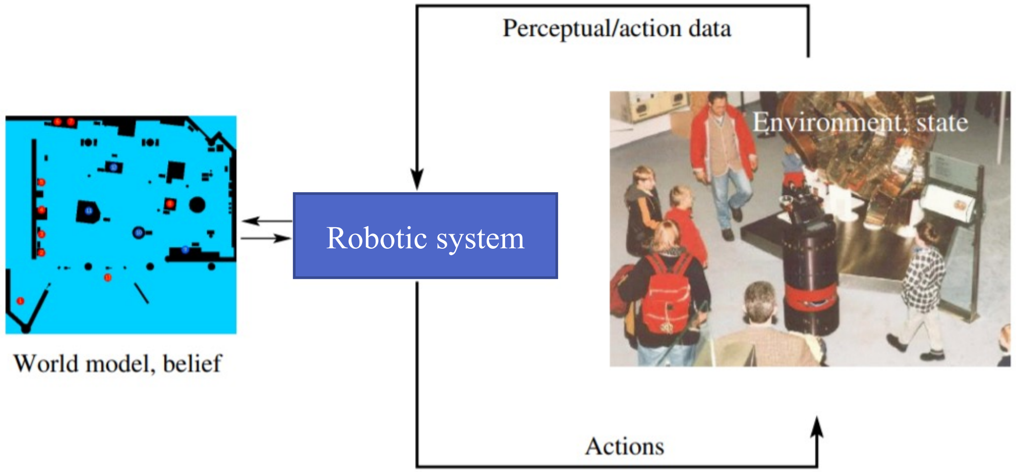
Robot States and the Environment

- **State** represents the environment as well as the robot, for example:
  - location of walls or objects (environment or static)
  - pose of the robot (physical or dynamical)
- **Environment interaction** comes in the form of
  - Sensor measurements
  - Control actions

Robot States and the Environment

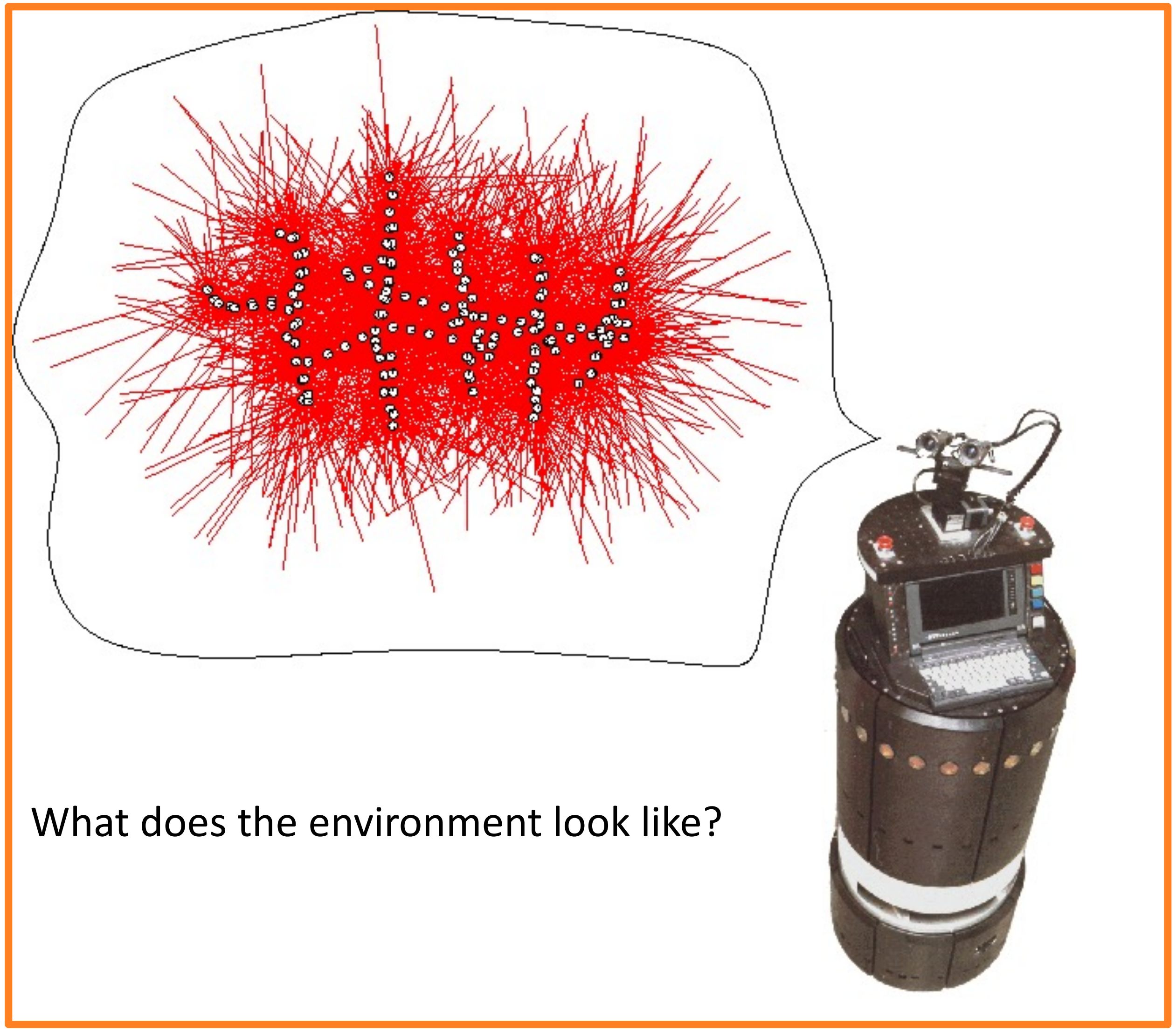
- **State** represents the environment as well as the robot, for example:
  - location of walls or objects (environment or static)
  - pose of the robot (physical or dynamical)
- **Environment interaction** comes in the form of
  - Sensor measurements
  - Control actions
- **Internal representation (or belief)** of the state of the world
  - In general, the state (or the world) cannot be measured directly
  - Perception is the process by which the robot uses its sensors to obtain information about the state of the environment

Maps and Representations



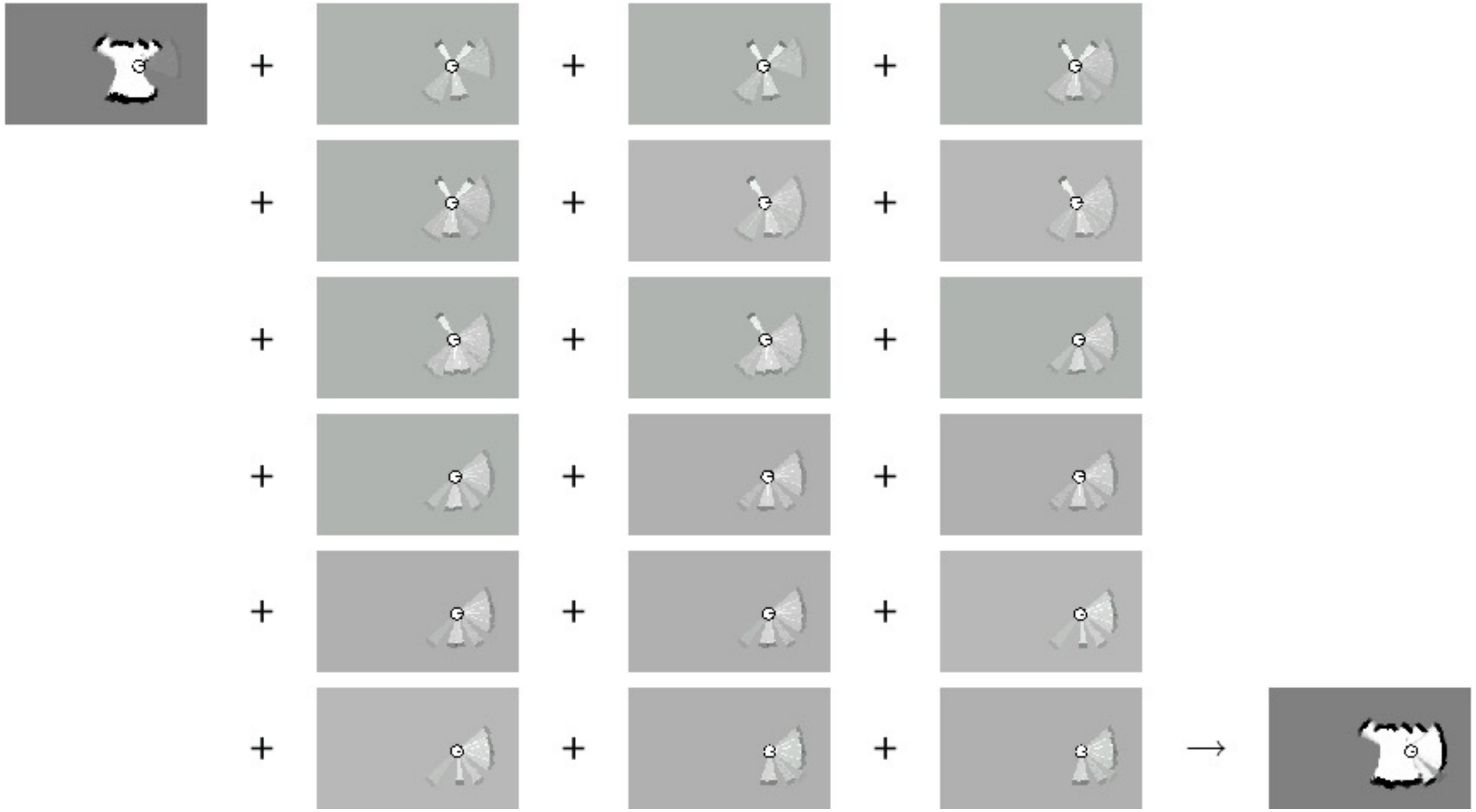
- **Mapping** is one of the fundamental problems in (mobile) robotics
- Maps allow robots to efficiently carry out their tasks and enable **localization**
- Successful robot systems rely on maps for localization, navigation, path planning, activity planning, control, etc.

The General Problem of Mapping

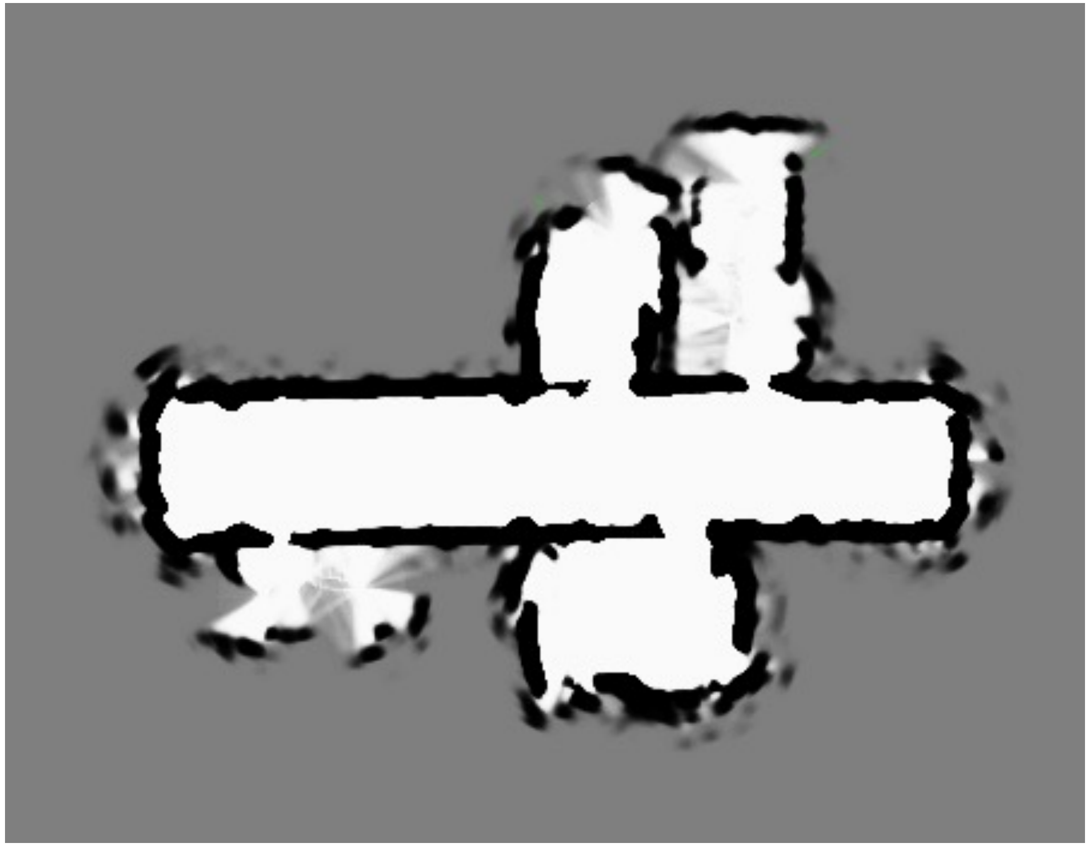


- Sensor interpretation
  - How do we extract relevant information from raw sensor data?
  - How do we represent and integrate this information over time?
- Robot locations must be estimated
  - How can we identify that we are at a previously visited place?
  - This problem is the so-called data association problem

Building a Map with Ultrasound Sensors

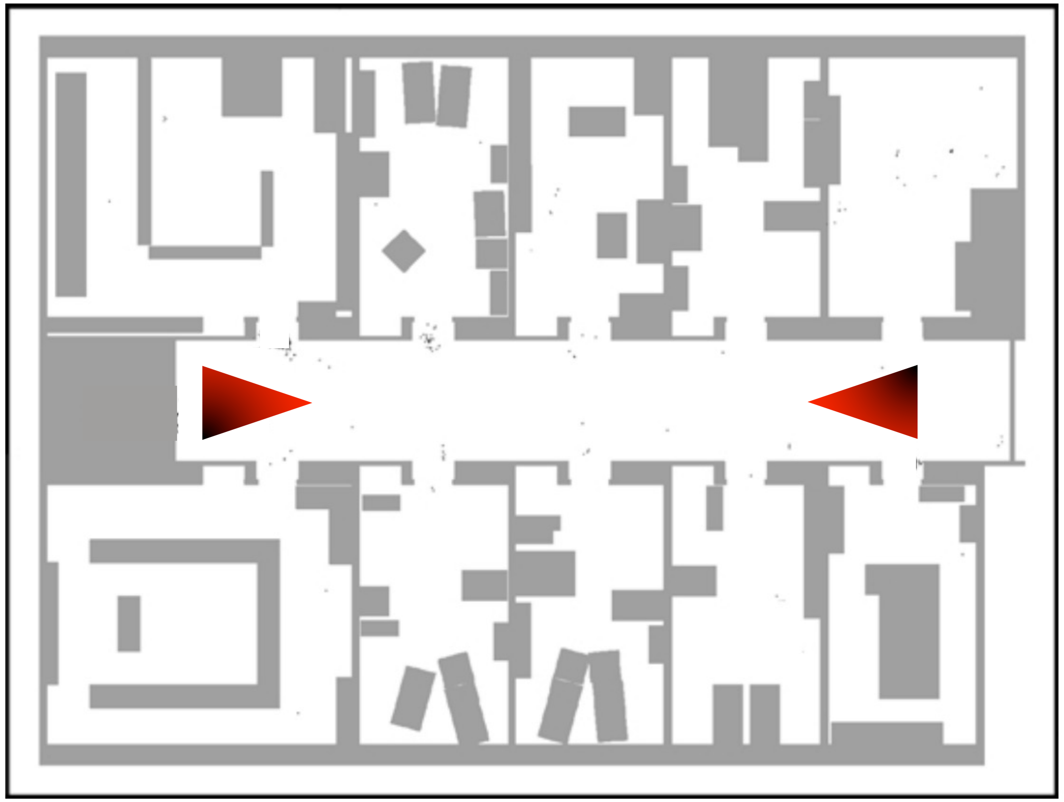


Building a Map with Ultrasound Sensors



The Localization Problem

- Determine the pose (state) of the robot relative to the **given map** of the environment
- This is also known as **position or state estimation problem**
- Given uncertainty in our measurements and ambiguity from locally symmetric environment, we need to recursively update our estimate or belief



Probability review

- $P(A)$ denotes the probability that event A is true
  - $0 \leq \mathrm{P}(A) \leq 1, P(\text{true}) = 1, P(\text{false}) = 0$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

Probability review

- $P(A)$ denotes the probability that event A is true
  - $0 \leq \mathrm{P}(A) \leq 1, P(\text{true}) = 1, P(\text{false}) = 0$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

Discrete Random Variables

- X can take on a countable number of values in $\{x_1, x_2, \dots, x_n\}$
- $P(X = x_i)$, or $P(x_i)$, is the probability that the random variable X takes on value x_i
- $P(\cdot)$ is called probability mass function

Continuous Random Variables

- X takes on values in the continuum
- $p(X = x)$, or $p(x)$, is a probability density function
- $P\big(x \in (a, b)\big) = \int_a^b p(x)dx$

Joint, Conditional, and Total Probability

- Joint Probability: $P(X = x \text{ and } Y = y) = P(x, y)$
  - If x and y are independent, then $P(x, y) = P(x) \cdot P(y)$

Joint, Conditional, and Total Probability

- Joint Probability: $P(X = x \text{ and } Y = y) = P(x, y)$
  - If x and y are independent, then $P(x, y) = P(x) \cdot P(y)$
- Conditional Probability: $P(x|y)$
  - $P(x, y) = P(x|y)P(y)$ → $P(x|y) = \frac{P(x,y)}{P(y)}$
  - If independent, then $P(x|y) = P(x)$

Joint, Conditional, and Total Probability

- Joint Probability: $P(X = x \text{ and } Y = y) = P(x, y)$
  - If x and y are independent, then $P(x, y) = P(x) \cdot P(y)$
- Conditional Probability: $P(x|y)$
  - $P(x, y) = P(x|y)P(y) \rightarrow P(x|y) = \frac{P(x,y)}{P(y)}$
  - If independent, then $P(x|y) = P(x)$
- Total Probability and Marginals

Discrete case:

$\sum_x P(x) = 1$

$P(x) = \sum_y P(x, y) = \sum_y P(x|y)P(y)$

Continuous case:

$\int_x p(x)dx = 1$

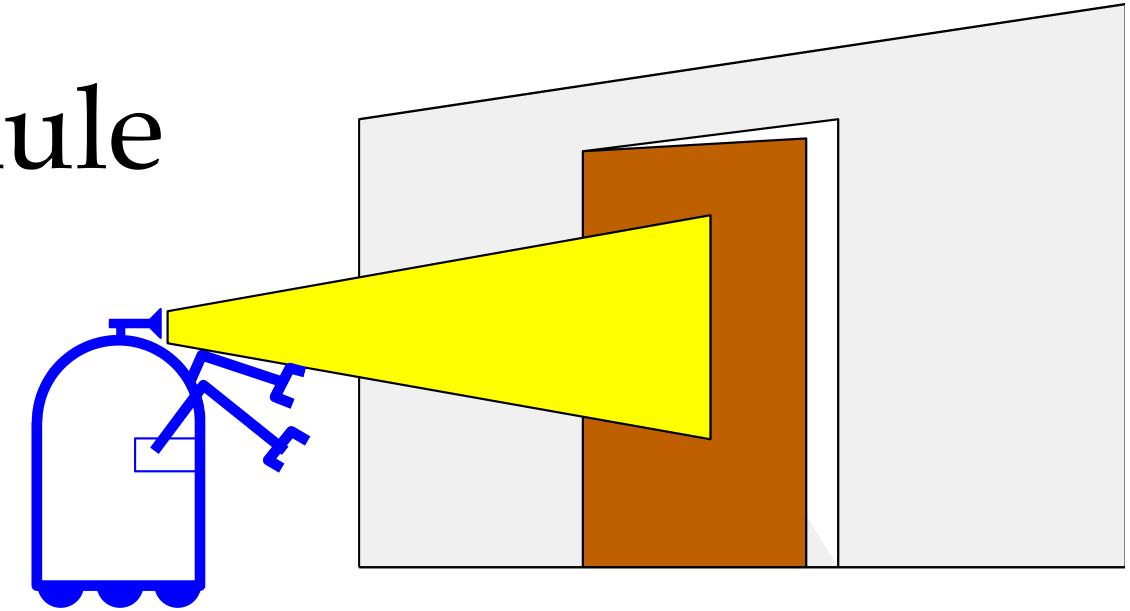
$p(x) = \int p(x, y)dy = \int p(x|y)p(y)dy$

Bayes's Formula

Recall: $P(x, y) = P(x|y)P(y) = P(y|x)P(x)$

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

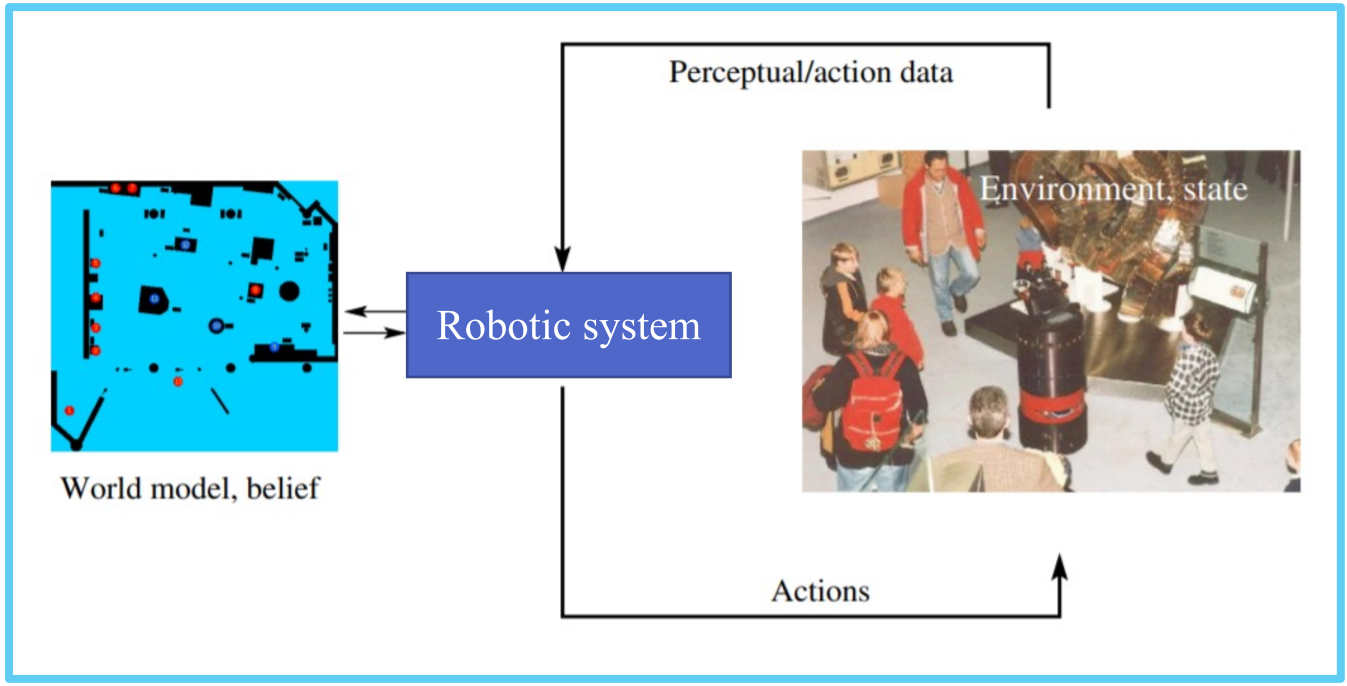
Door example of Bayes Rule



Suppose a robot obtains measurement z.
What is $P(\text{open}|z)$?

Robot's Belief over States

Belief: Robot's knowledge about the state of the environment

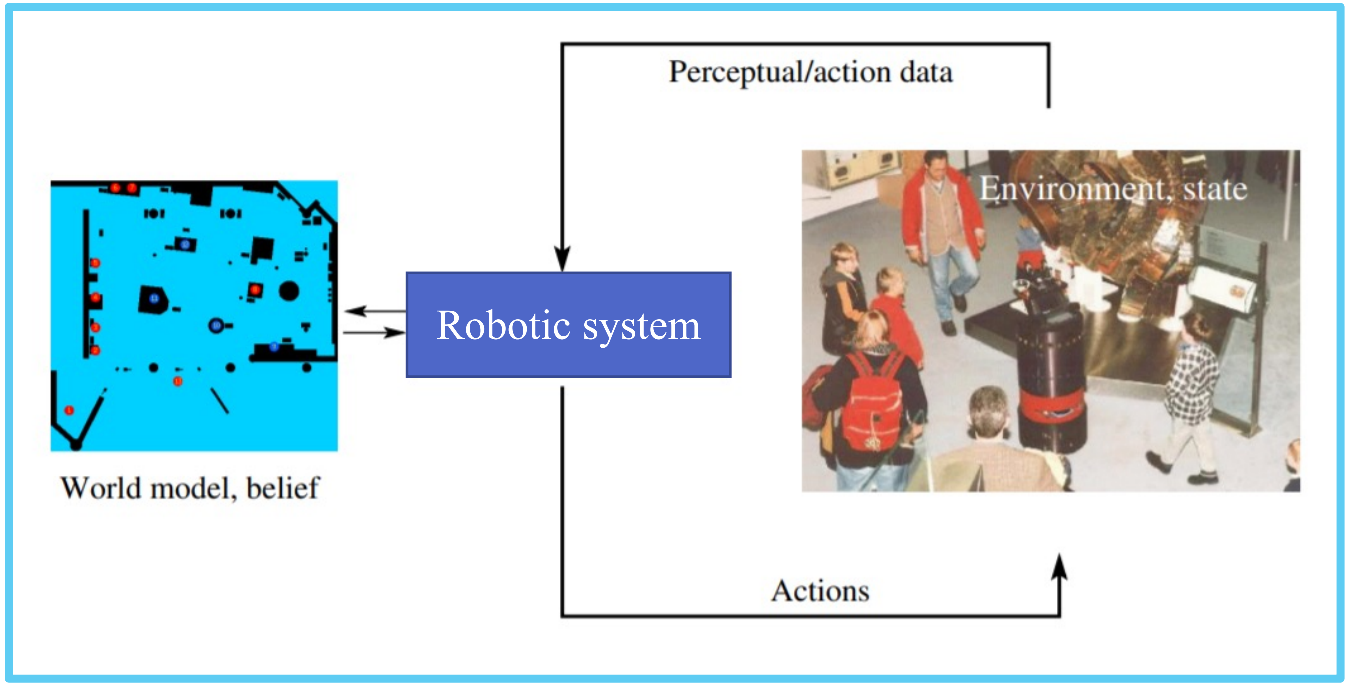


Robot's Belief over States

Belief: Robot's knowledge about the state of the environment

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

Posterior distribution over state at time *t* given all past measurements and control



Robot’s Belief over States

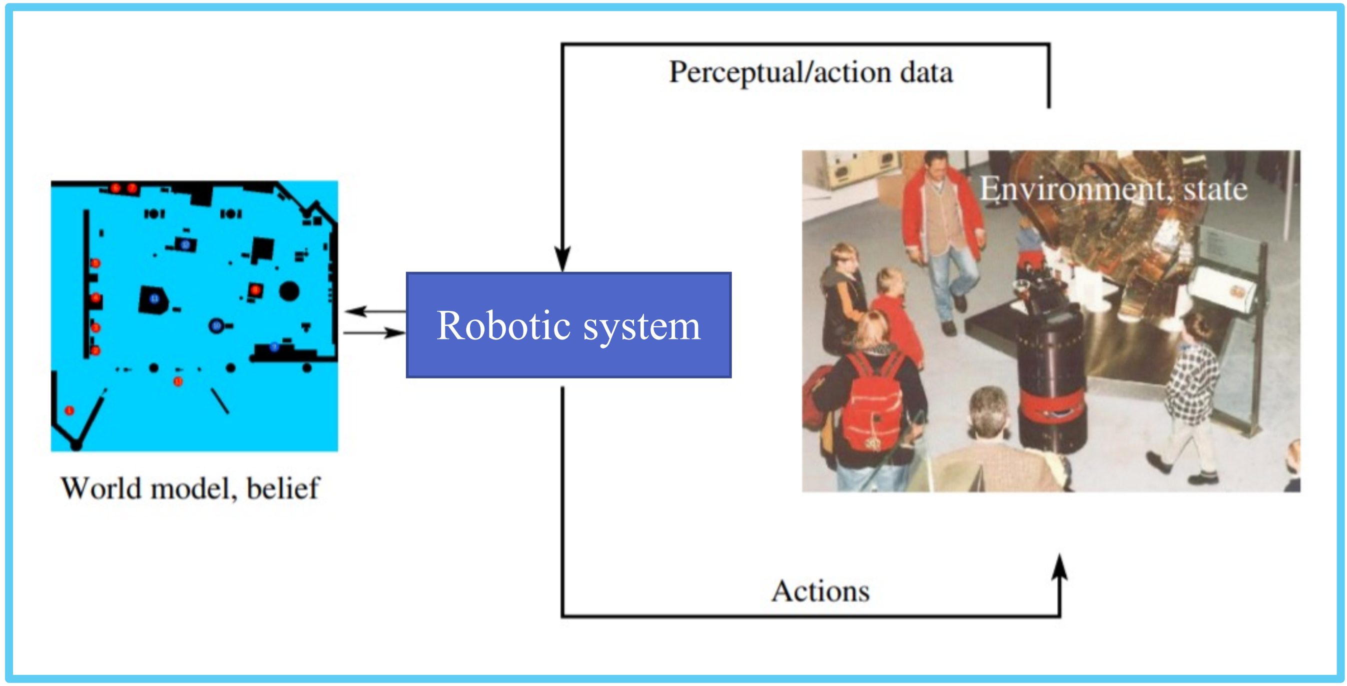
Belief: Robot’s knowledge about the state of the environment

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

Posterior distribution over state at time *t* given all past measurements and control

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

Calculating $bel(x_t)$ from $\overline{bel}(x_t)$ is called **correction** or **measurement update**



Today's Plan

- What is filtering, mapping, and localization?
  - Probability review!
- Bayes Filters (discrete)

Notation and Definitions

- Discrete time model

  $x_{t_1:t_2} = x_{t_1}, x_{t_1+1}, x_{t_1+2}, \ldots, x_{t_2}$ sequence of states t_1 to t_2

- Robot takes one measurement at a time

  $z_{t_1:t_2} = z_{t_1}, \ldots, z_{t_2}$ sequence of all measurements from t_1 to t_2

- Control also exercised at discrete steps

  $u_{t_1:t_2} = u_{t_1}, u_{t_1+1}, u_{t_1+2}, \ldots, u_{t_2}$ sequence control inputs

State Evolution / Models

Evolution of the state and measurements are governed by probabilistic laws:

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

If the state is *complete,* we can succinctly state:

$$p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t | x_{t-1}, u_t)$$

State Evolution / Models

Evolution of the state and measurements are governed by probabilistic laws:

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

If the state is *complete,* we can succinctly state:

$$p(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t|x_{t-1}, u_t)$$

Measurement process given by:

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

Similarly, if measurement is complete:

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

Discrete Bayes Filter Algorithm: Setup

- Evolution of the state is governed by probabilistic state transition:

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

- Measurement process given by:

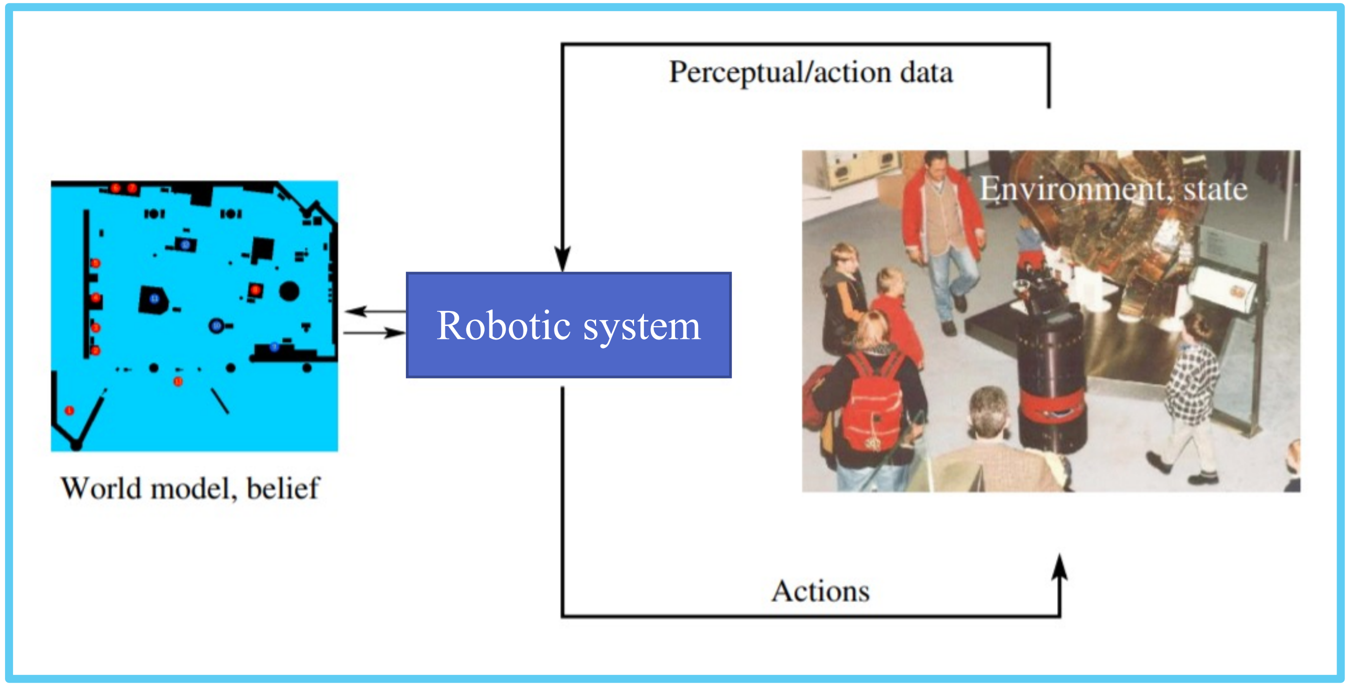
  $$p(z_t|x_t)$$

Robot's Belief over States

Belief: Robot's knowledge about the state of the environment

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

Posterior distribution over state at time *t* given all past measurements and control



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

Calculating $bel(x_t)$ from $\overline{bel}(x_t)$ is called **correction** or **measurement update**

Recursive Bayes Filter

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 $bel(x_t)$

Recursive Bayes Filter

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

for all x_t do:

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

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

end for

return $bel(x_t)$

1

2

3

Recursive Bayes Filter

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

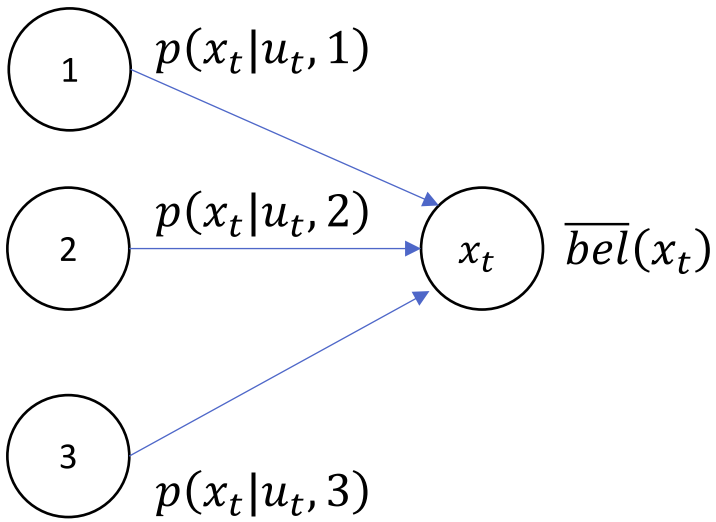
for all x_t do:

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

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

end for

return $bel(x_t)$



Recursive Bayes Filter

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

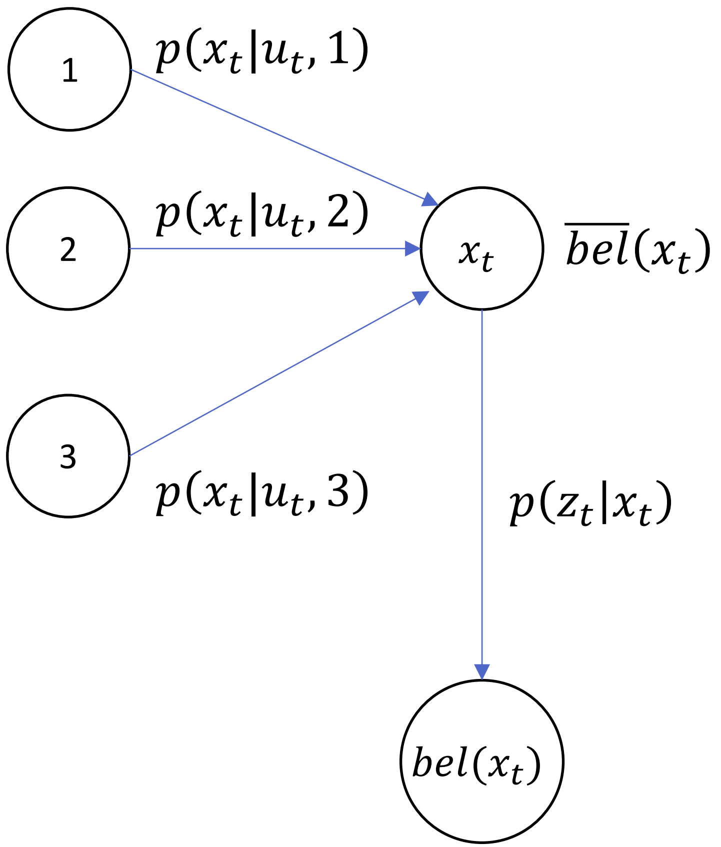
for all x_t do:

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

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

end for

return $bel(x_t)$



Bayes Filters (1)

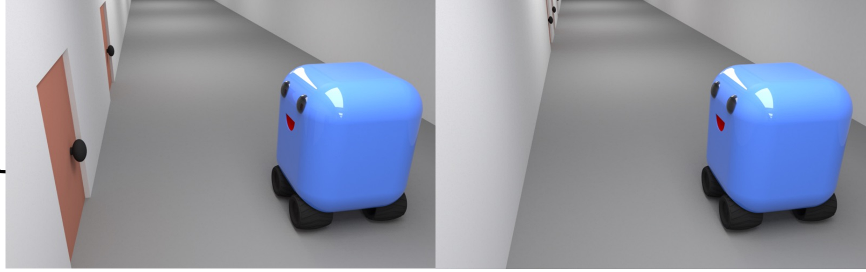
Bayes Filters (2)

Bayes Filters (3)

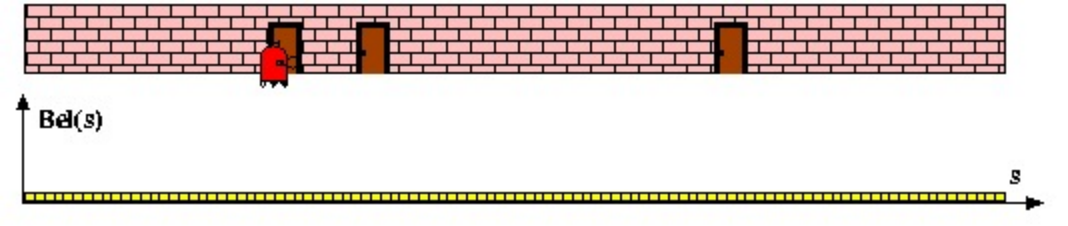
Bayes Filters (4)

Bayes Filters (5)

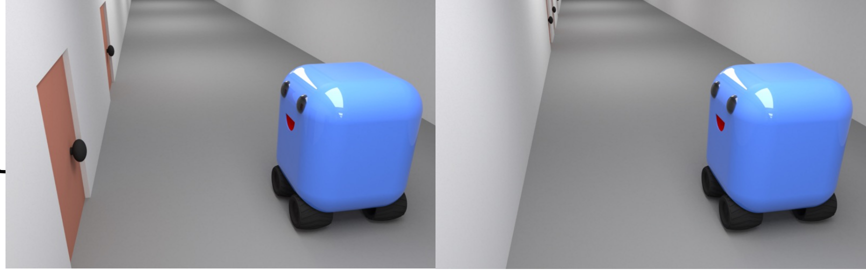
Discrete Bayes Filter - Illustrati



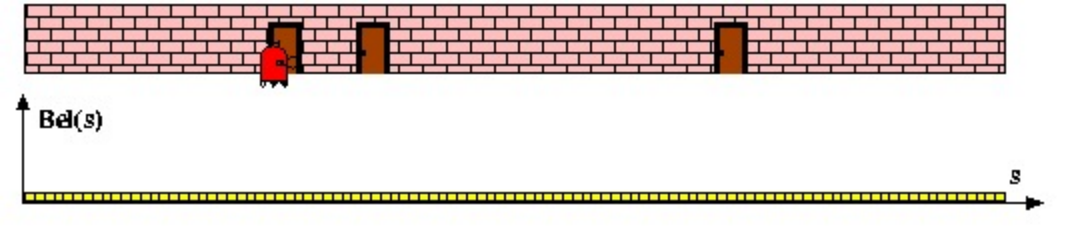
time = 0

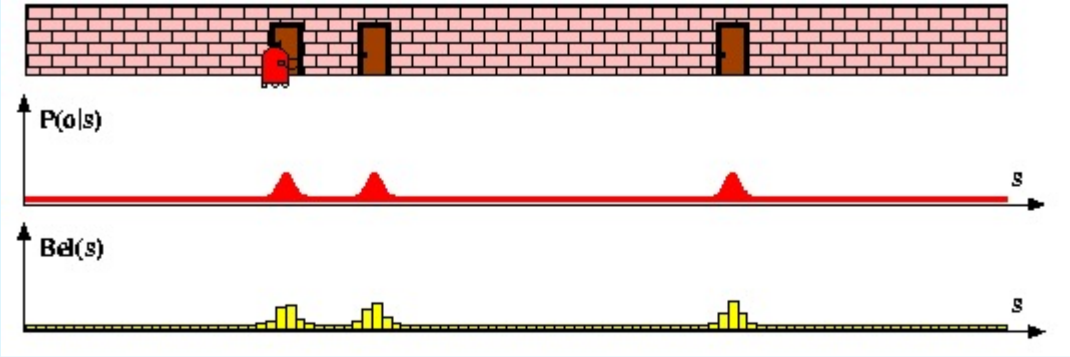


Discrete Bayes Filter - Illustrati

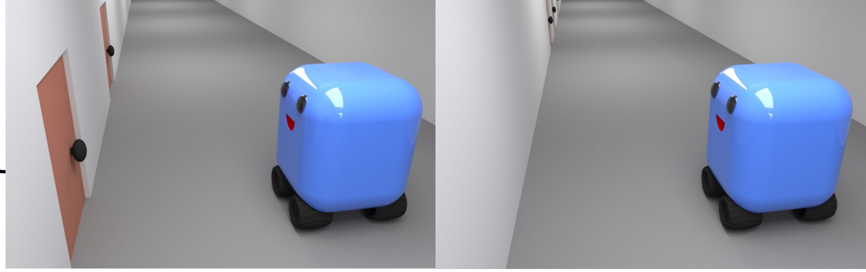


time = 0

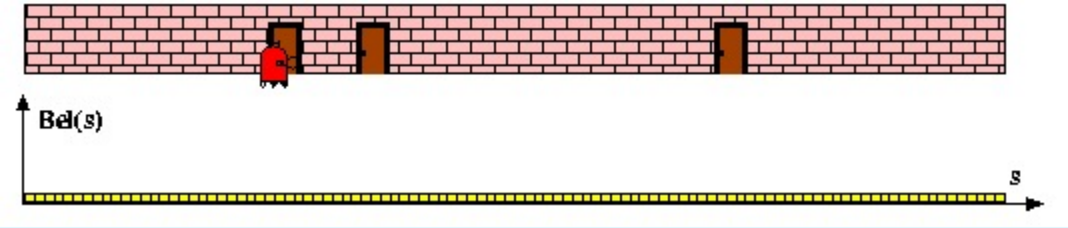


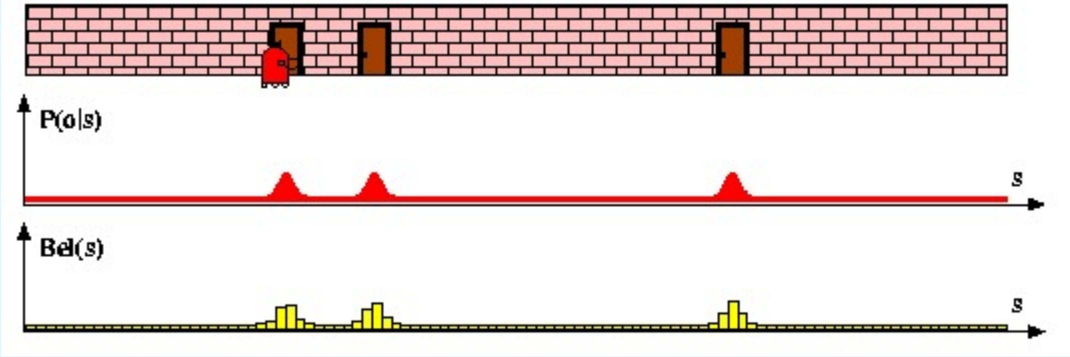


Discrete Bayes Filter - Illustrati

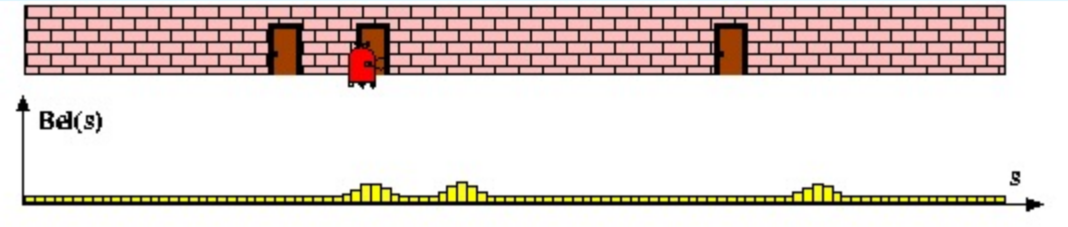


time = 0

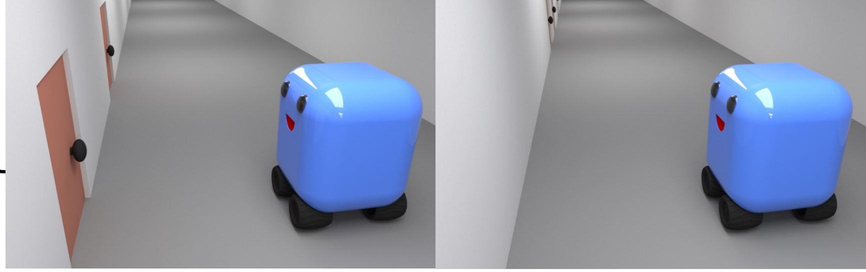




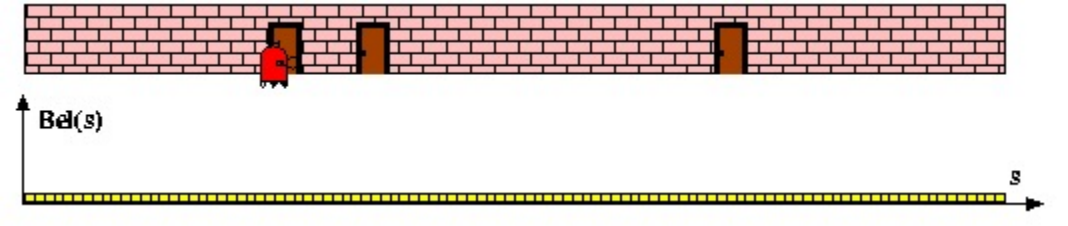
time = 1

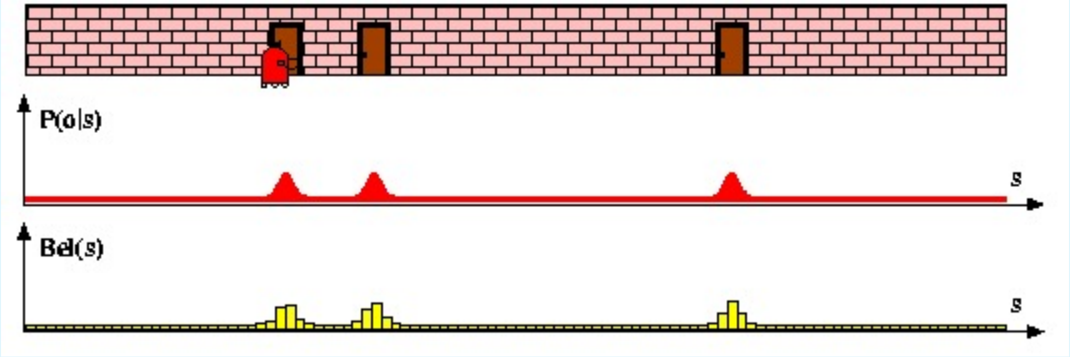


Discrete Bayes Filter - Illustrati

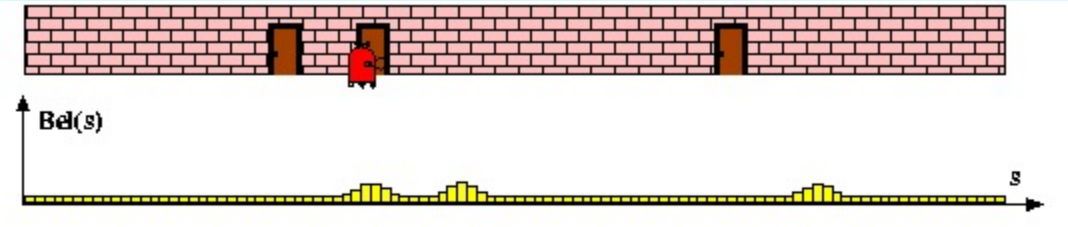


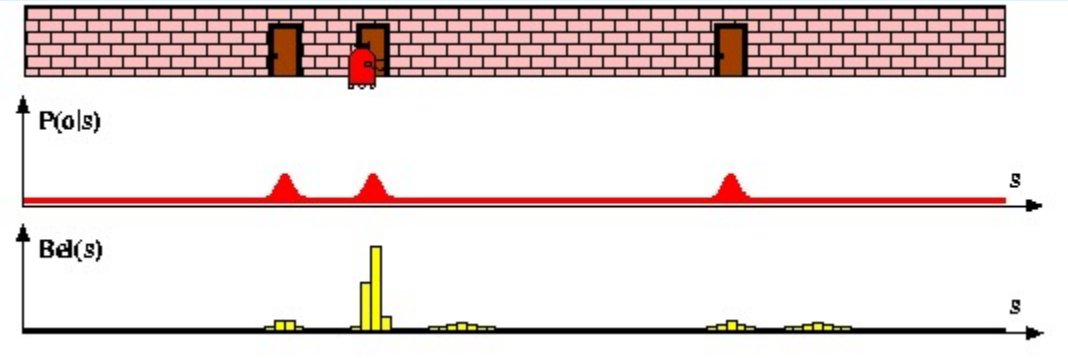
time = 0



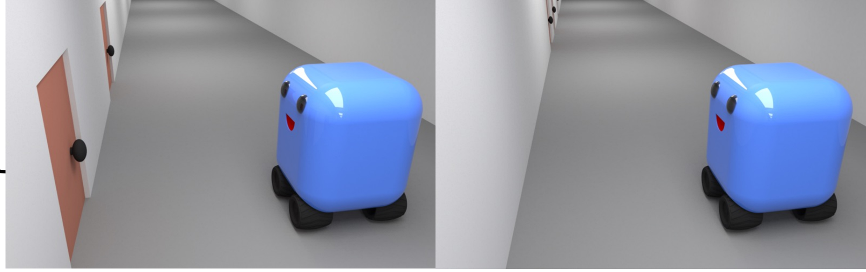


time = 1

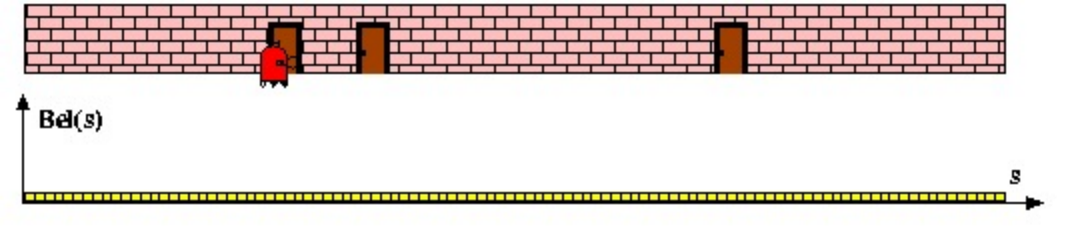


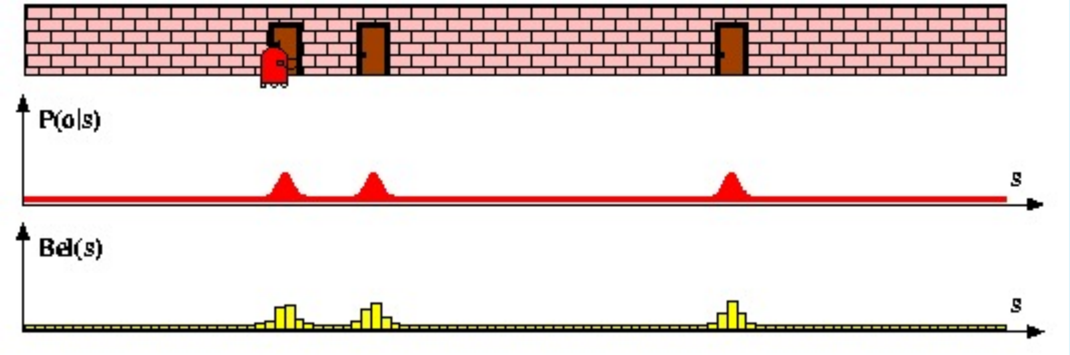


Discrete Bayes Filter - Illustrati

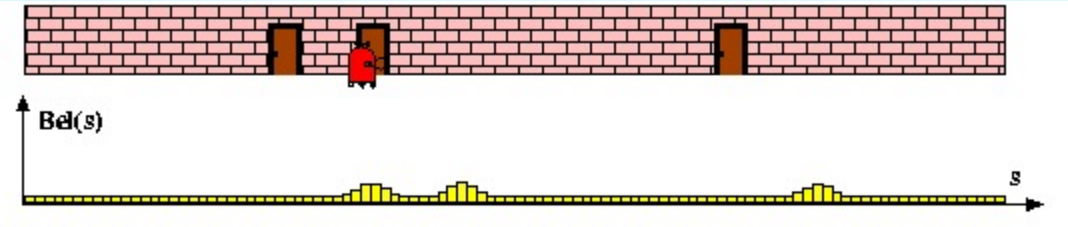


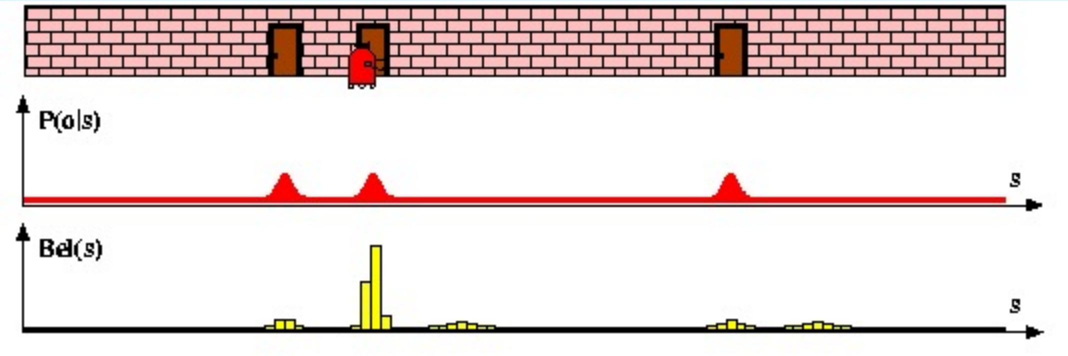
time = 0



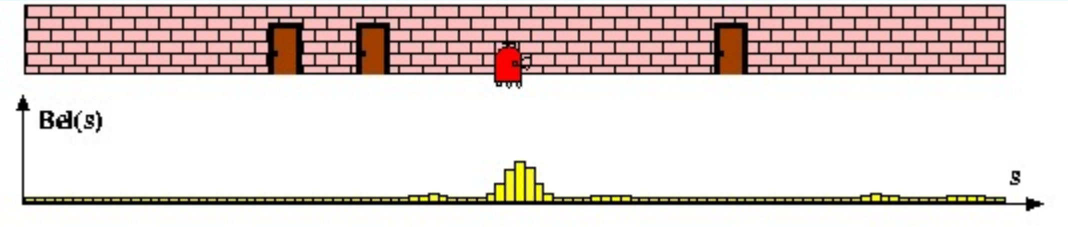


time = 1





time = 2



Bayes Filter Recap

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

- Prediction

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

- Correction

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

What if we have a good model of our (continuous) system dynamics and we assume a Gaussian model for our uncertainty?

→ Kalman Filters!

Summary

- **Bayes filters** are a probabilistic tool for **estimating the state of dynamic systems**
  - **They are everywhere!** Kalman filters, Particle filters, Hidden Markov models, Dynamic Bayesian networks, Partially Observable Markov Decision Processes (POMDPs), …
  - Bayes rule allows us to compute probabilities that are hard to assess otherwise
  - Recursive Bayesian updating can efficiently combine evidence over time
- Next time: Look at extensions of this basic filtering approach (Kalman filtering and particle filtering)

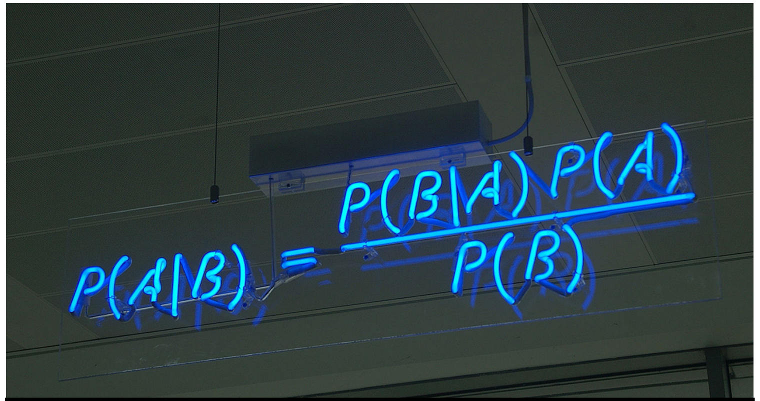
Extra Slides

Fun Fact: Who is Bayes?

Bayes was an English **statistician**, philosopher, and minister who lived from 1701 to 1761, and is known for two works:

1. Divine Benevolence, or an Attempt to Prove That the Principal End of the Divine Providence and Government is the Happiness of His Creatures (1731)
2. An Introduction to the Doctrine of Fluxions, and a Defence of the Mathematicians Against the Objections of the Author of The Analyst (1736), in which he **defended the logical foundation of Isaac Newton's calculus** ("fluxions") against the criticism of George Berkeley, author of The Analyst

Bayes never published his most famous accomplishment **Bayes' Theorem**. These notes were edited and published after his death by Richard Price.



From the HP Autonomy Lab

Probably not Bayes