EECE 5550
Mobile Robotics

Lecture 7: Probabilistic Robotics

Derya Aksaray

Assistant Professor

Department of Electrical and Computer Engineering

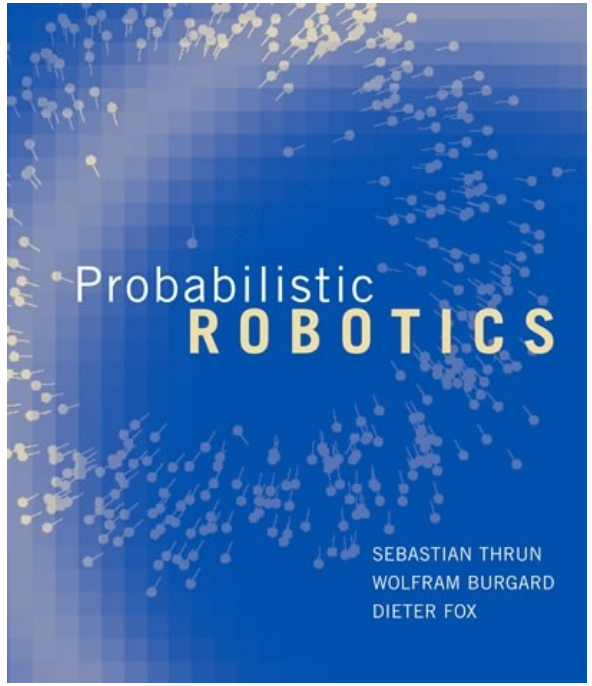
Northeastern University

Plan of the day

- Uncertainty in robotics
- Probabilistic robotics
- Bayesian networks
- State estimation and dynamic Bayes nets
- Recursive Bayesian estimation & the Bayes Filter

References

http://stefanosnikolaidis.net/course-files/CS545/Lecture3.pdf



Chapters 1 & 2

Motivation: The problem of uncertainty

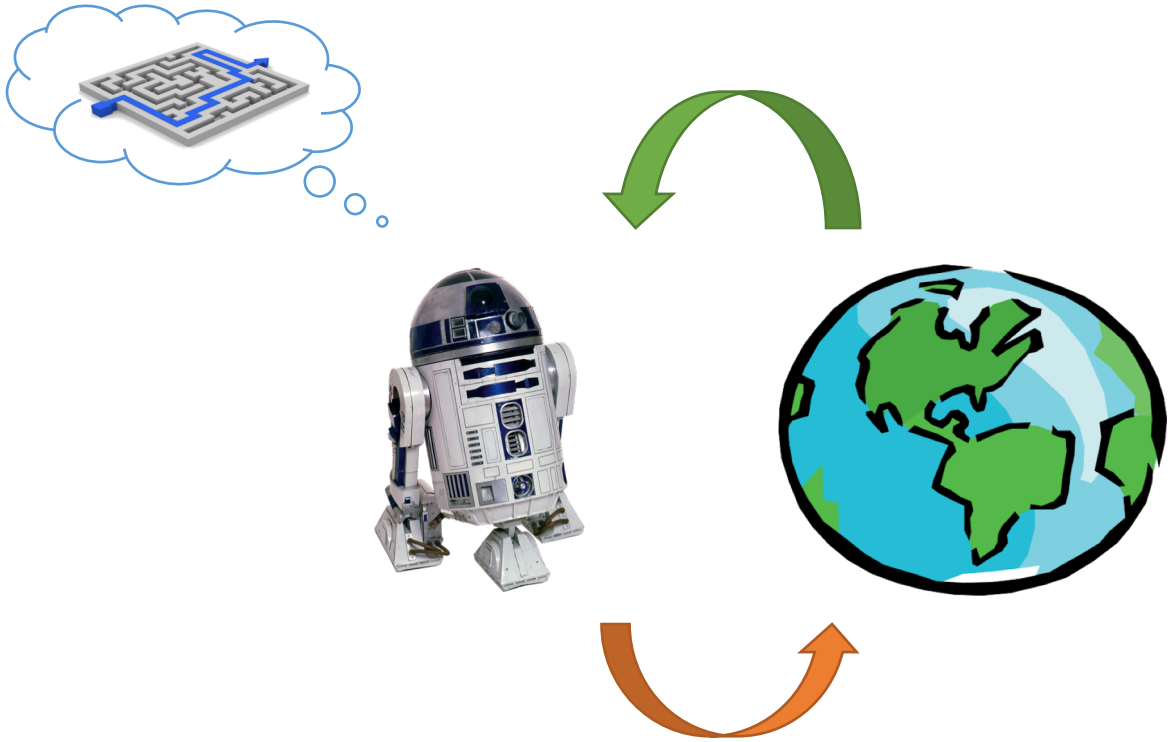
The Goal: Sense -> Think -> Act

The Problem: Uncertainty

What happens if:

- Our models of the world
- Our models of the robot

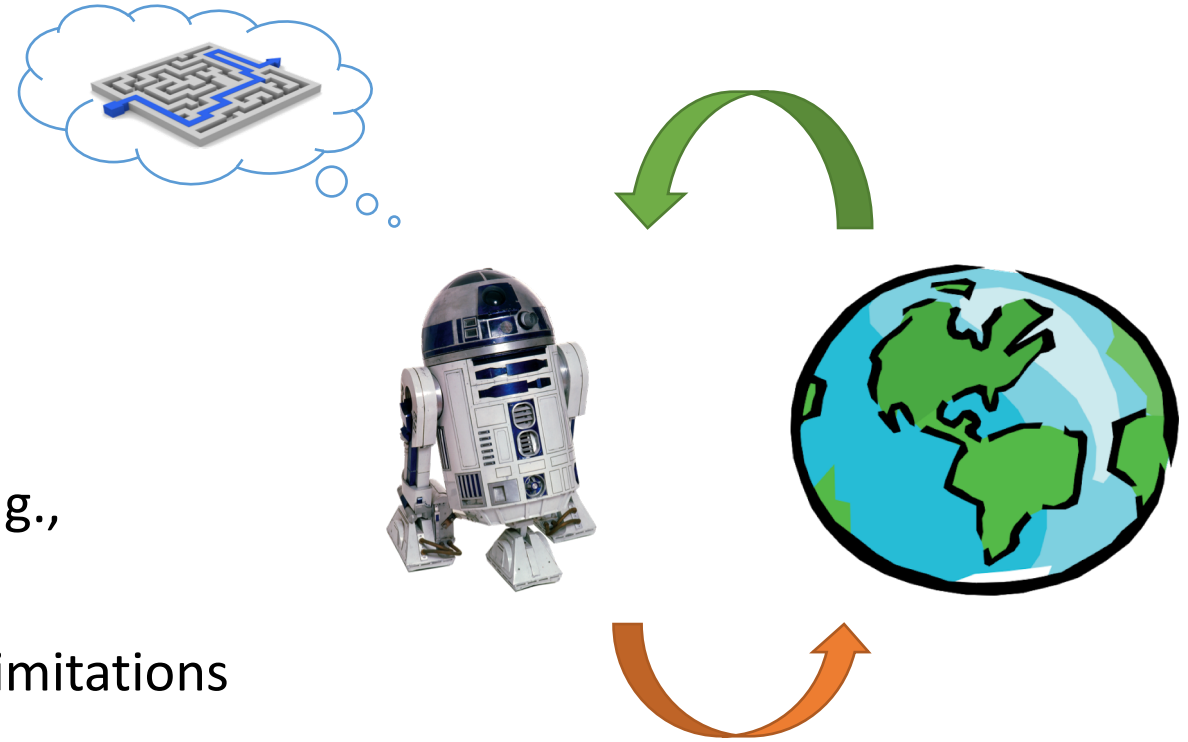
are *wrong*?



Motivation: The problem of uncertainty

The Goal: Sense -> Think -> Act

The Problem: Uncertainty

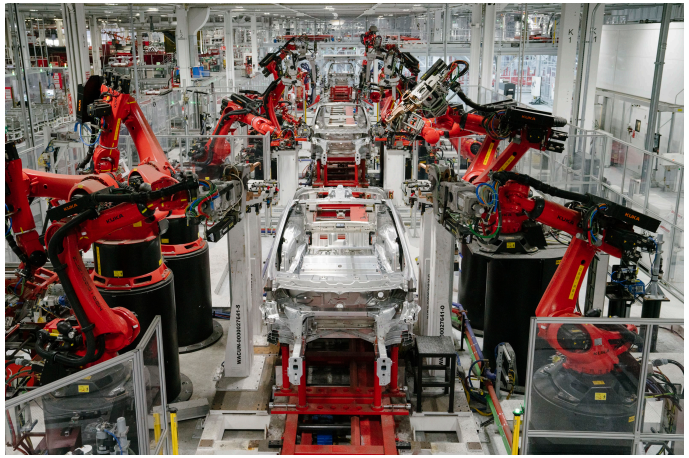


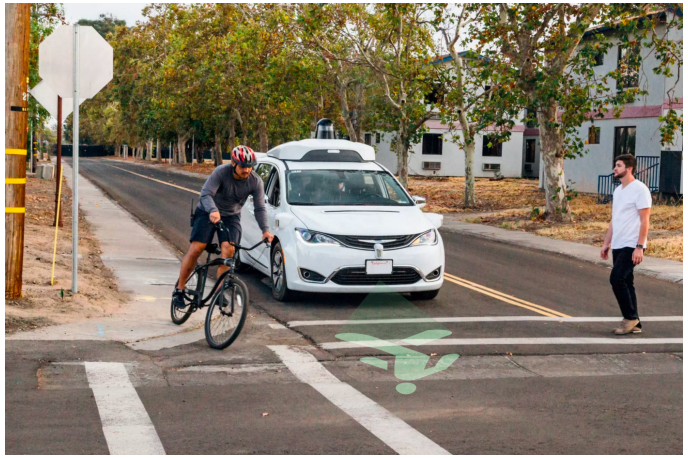
Sources of uncertainty:

- **Environment:** inherently unpredictable (e.g., operating in the proximity of humans)
- **Sensing**: noise, finite resolution, physical limitations
- **Actuation**: process noise, mechanical failures, etc.
- **Modeling:** Models only *approximate* the real world.

There are a lot of things that we don't know!

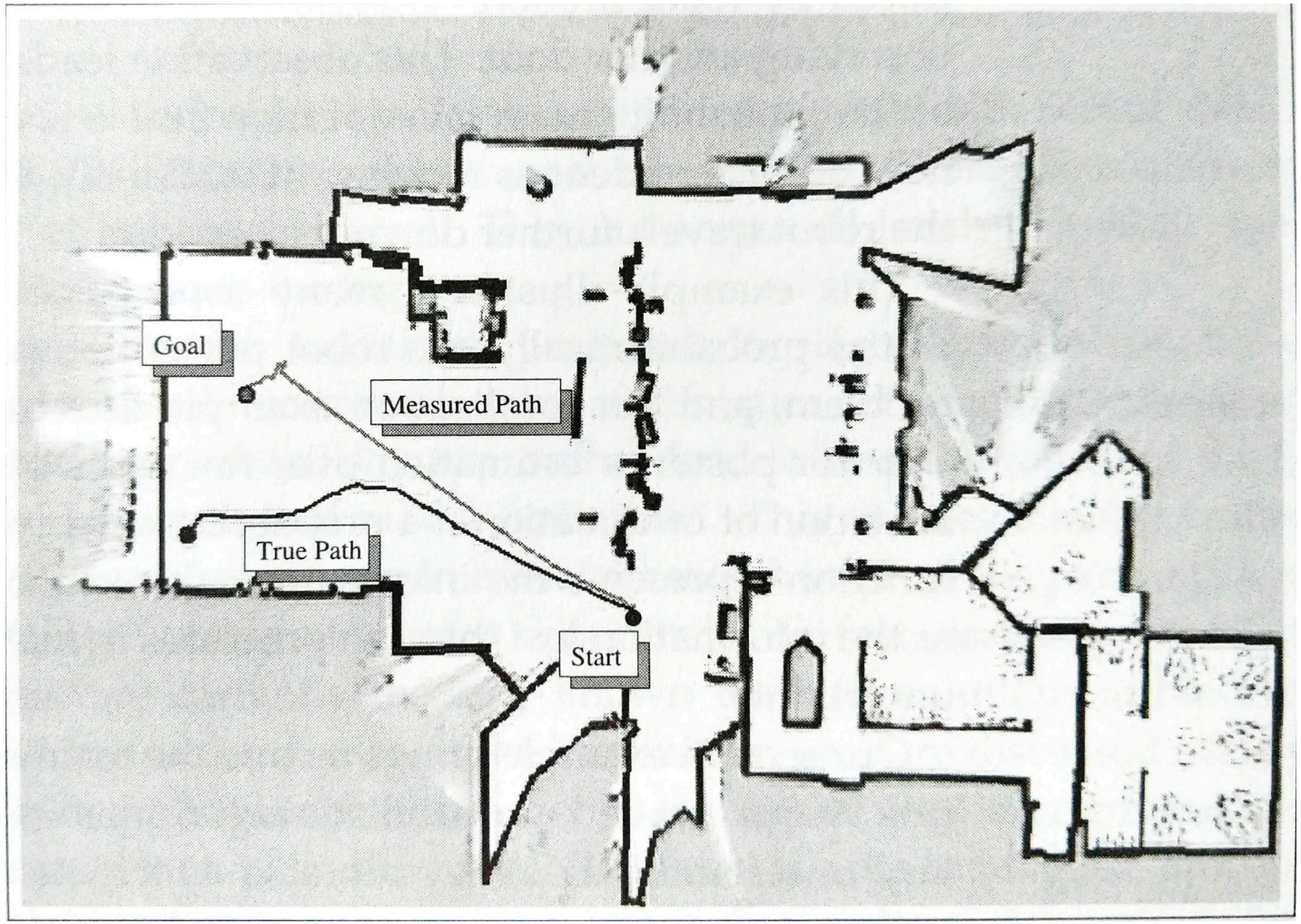
Level of uncertainty





https://www.youtube.com/watch?v=4DKrcpa8Z_E

Example: The effects of noise in robot navigation

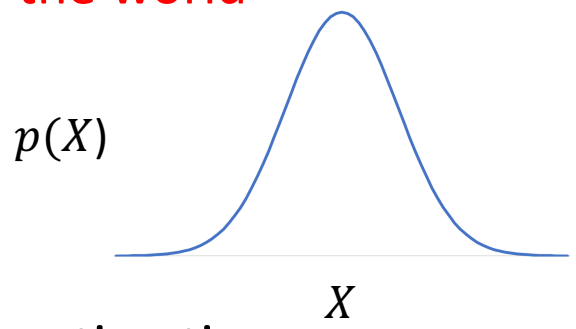


Probabilistic robotics

Main idea: *Explicitly represent* our degree of (un)certainty about the world using the tools of (Bayesian) *probability*

Central object of study: *Belief*

⇒A probability distribution that models our uncertainty over possible states *X* of the world



Rev. Thomas Bayes

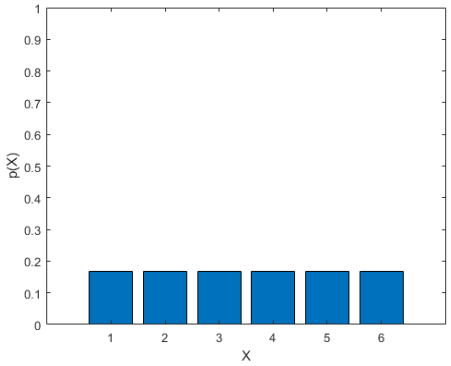
- **Perception:** state estimation
- **Planning:** utility optimization

A simple example

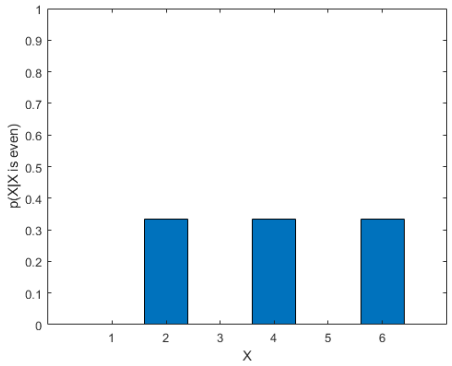
Suppose I roll a fair die, but don't tell you the result *X*

Q1: What should your belief be about *X*?

Q2: Suppose that now I look at *X*, and tell you that its value is even. What should your belief be now?



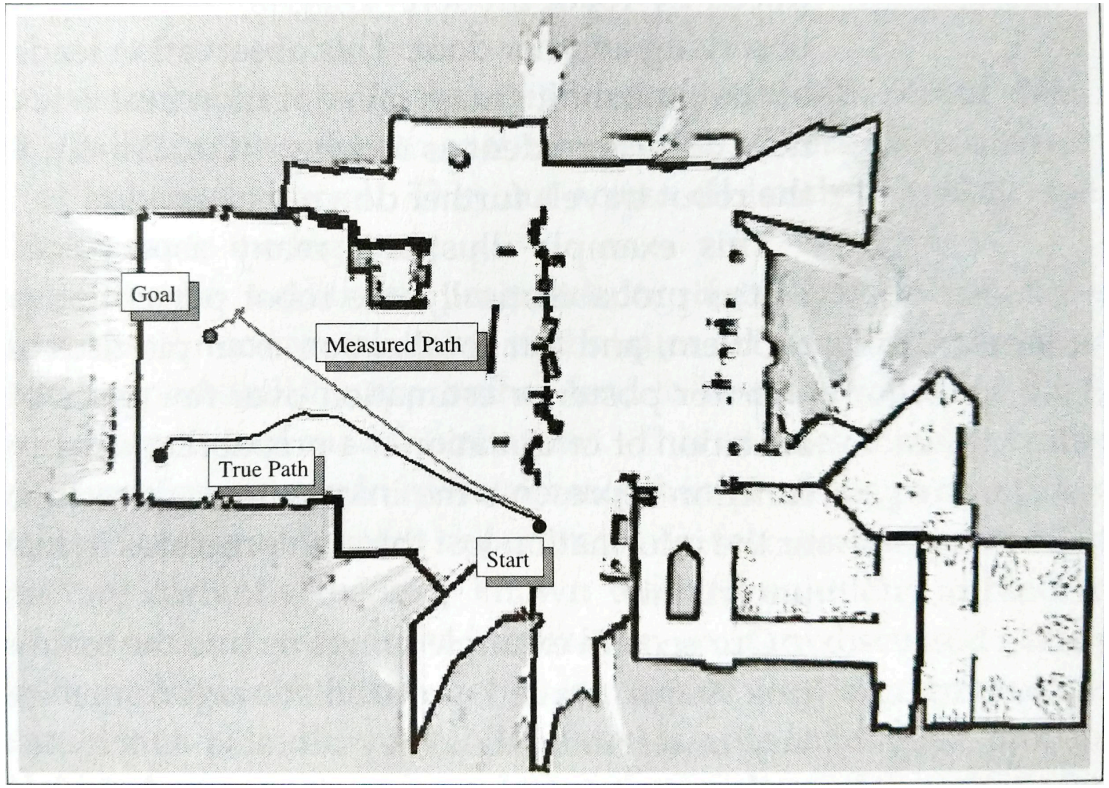
prior



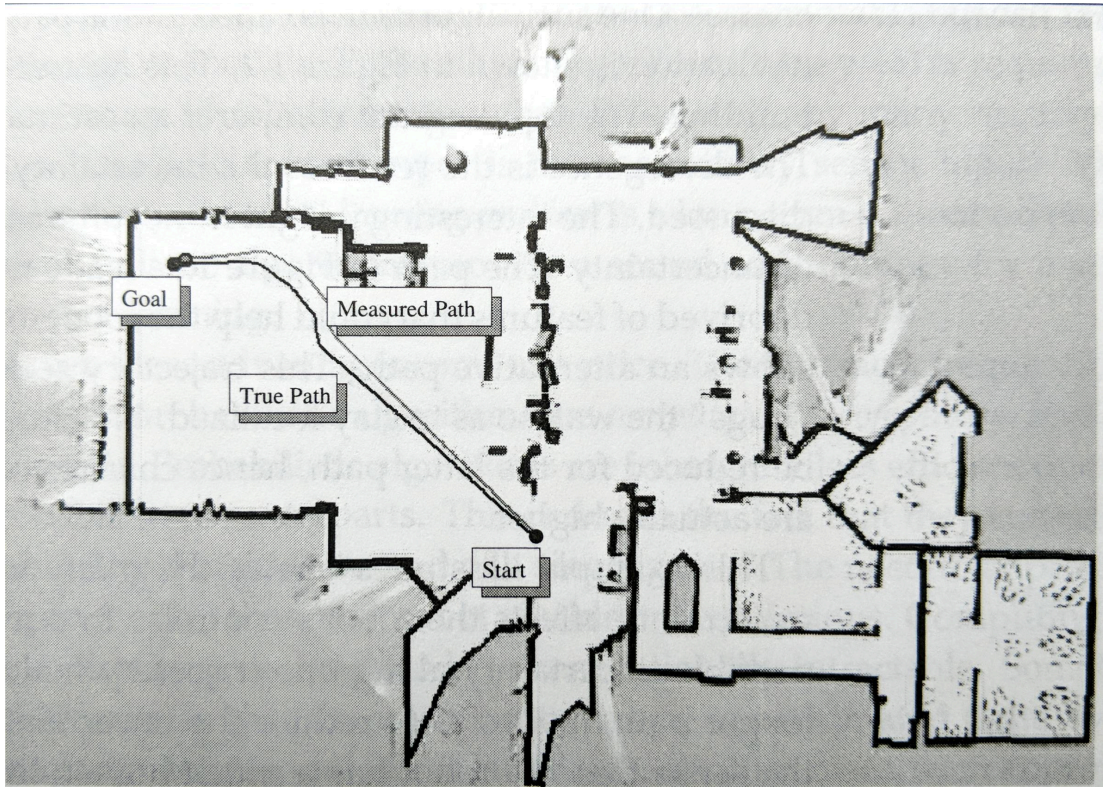
posterior

- Different than the frequency approach!
- In this example, it is *not the world* that is changing, but rather *our information about* the world!
- Beliefs model our *state of knowledge* of the world!

Example robotics application: localization



Dead reckoning



Coastal navigation

Probabilistic robotics

Advantages

- Can accommodate inaccurate models
- Can accommodate imperfect sensors
- Robust in real-world applications

Disadvantages

- Computationally demanding
- False assumptions
- Approximate

Very powerful approach to many hard robotics problems!

Probabilistic state estimation

- Localization
- Mapping

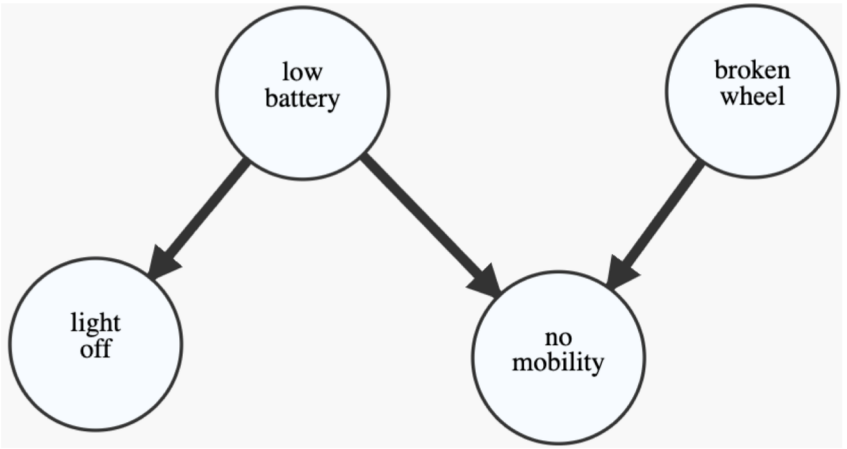
Probabilistic decision-making

- Planning
- Exploration

Bayesian networks

A Bayesian network is a directed acyclic graph (DAG) that models the factorization of a probability distribution $p(X_1, \ldots, X_n)$ as a product of *conditional* distributions.

Usually when the robot has sufficient battery, the robot's light is turned on, and it can move. If the robot's battery is low, then its light is turned off and it loses its mobility as well. A broken wheel could also result in the robot's loss of mobility, even if the battery is full.



Each node: Random variable
Each edge: Parent-descendent relationship

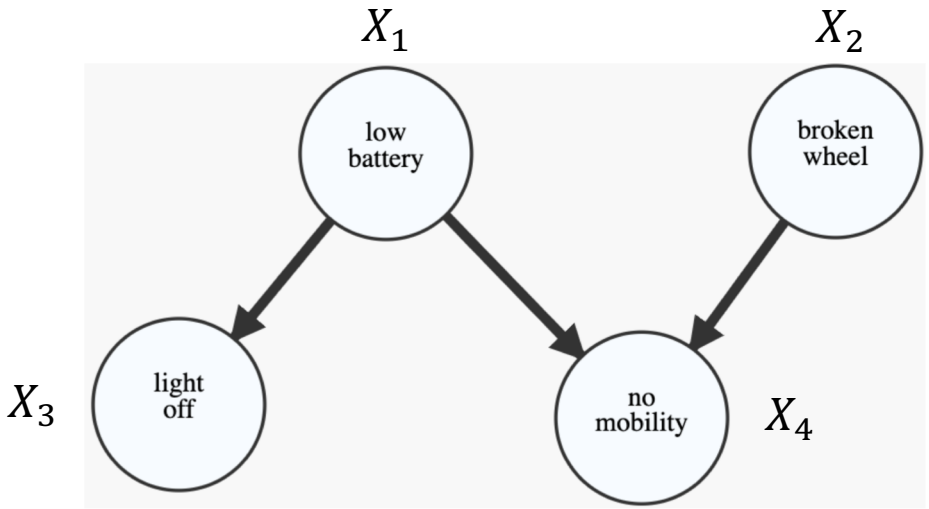
Bayesian network

A Bayesian network is a directed acyclic graph (DAG) that models the factorization of a probability distribution $p(X_1, \dots, X_n)$ as a product of *conditional* distributions.

Joint probability dist.: 2^n assignments!

Bayes nets are a convenient language for building complex joint distributions from simple parts

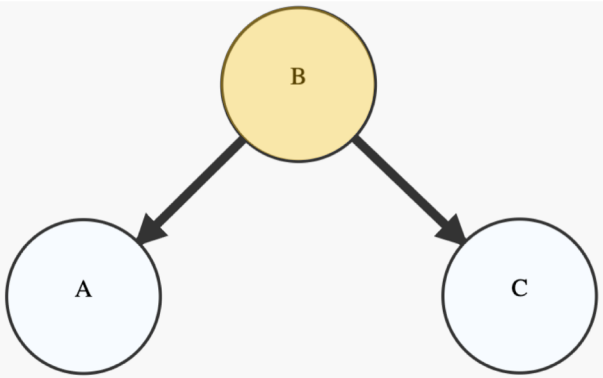
$$p(X_1, \dots, X_n) = \prod_{i=1}^{n} p(X_i | pa(X_i))$$



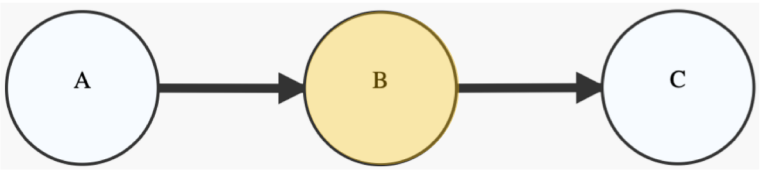
$$p(X_1, X_2, X_3, X_4) = p(X_1)p(X_2)p(X_3|X_1)p(X_4|X_1, X_2)$$

Bayesian network – conditional independence

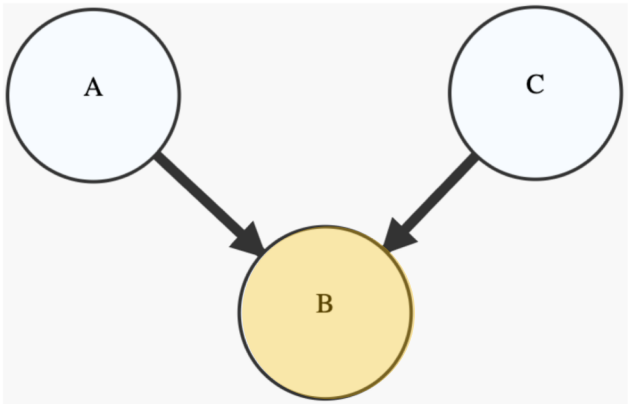
Every node in a Bayesian network is conditionally independent of its non-descendants, given its parents.



Common parent
Fixing B decouples A and C



Cascade
Knowing B decouples A and C



V-structure
Knowing B couples A and C

Probability recap

- Conditional probability $P(X|Y) = \frac{P(X,Y)}{P(Y)}$
- Product rule $P(X,Y) = P(X|Y)P(Y)$
- Chain rule $P(X_1, X_2, \dots, X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2) \dots$
  $= \prod_{i=1}^{n} P(X_i|X_1, \dots, X_{i-1})$
- X and Y independent iff $P(X,Y) = P(X)P(Y)$
- X and Y conditionally independent given Z iff $P(X,Y|Z) = P(X|Z)P(Y|Z)$ $X \perp Y|Z$

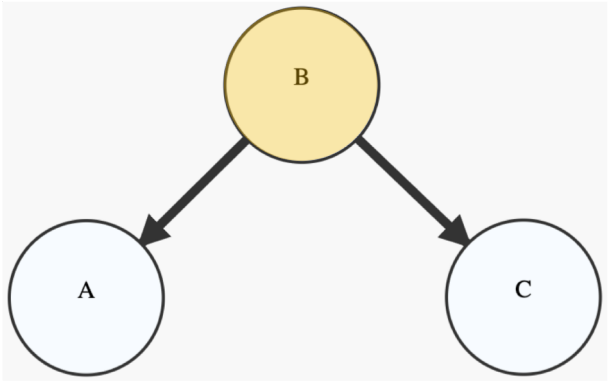
Common parent

- From Bayes network:

$$\Pr(A, B, C) = \Pr(B)\Pr(A|B)\Pr(C|B)$$

$$\Pr(A, C|B) = \frac{\Pr(A, C, B)}{\Pr(B)}$$

$$= \Pr(A|B)\Pr(C|B)$$



Common parent
Fixing B decouples A and C

Cascade

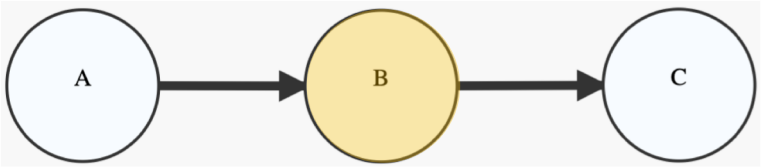
- From Bayes network:

$$\Pr(A, B, C) = \Pr(A)\Pr(B|A)\Pr(C|B)$$

$$\Pr(C|A, B) = \frac{\Pr(A, B, C)}{\Pr(A, B)}$$

$$= \frac{\Pr(A)\Pr(B|A)\Pr(C|B)}{\Pr(A)\Pr(B|A)}$$

$$= \Pr(C|B)$$



Cascade

Knowing B decouples A and C

V-structure

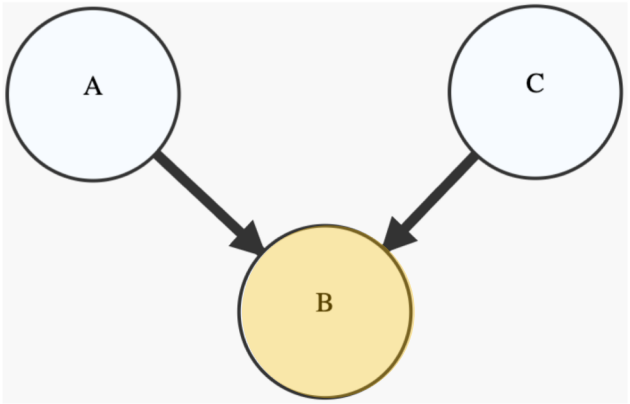
Suppose that A, B, C Boolean variables.

If we know that the grass is wet (B is true) and the sprinkler didn't go on (C is false), then the probability that A is true must be one, because that is the only other possible explanation.

Hence, A and C are not independent given B.

Rain
(true or false)

Sprinkler on
(true or false)



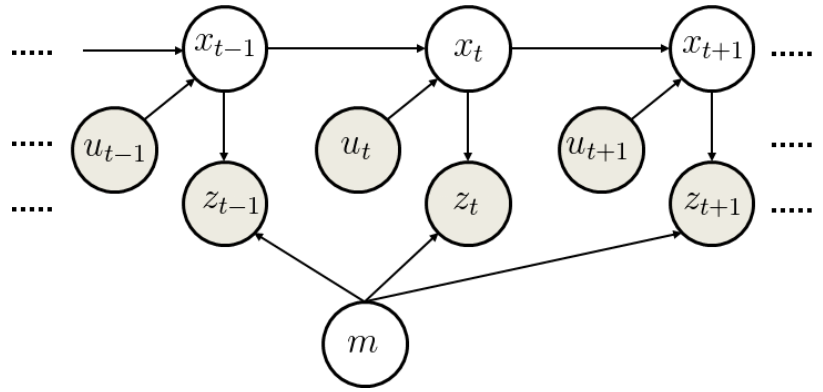
Grass wet (true or false)

V-structure
Knowing B couples A and C

Markov assumption

Every node in a Bayesian network is conditionally independent of its non-descendants, given its parents.

E.g., If we know the current state of the robot, past and future states are conditionally independent of one another.



If we know where the robot is now, then knowing where the robot was 5 minutes ago doesn't give us any more information than we already have, given the current state.

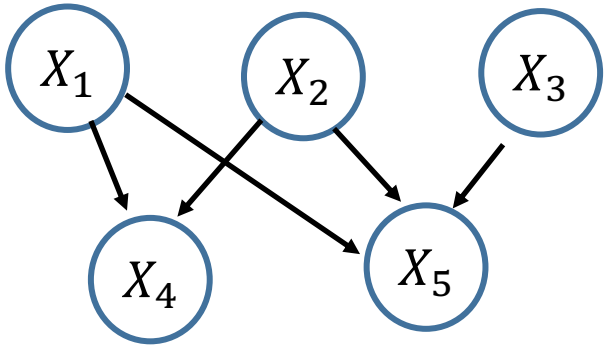
*Static world assumption
**Independent noise assumption

Bayesian networks as generative models

Bayesian networks provide a convenient *generative* description of a probability distribution p: that is, they tell us how to *draw samples from it.*

Example for the graph at right: X_1, X_2, X_3, X_4, X_5

Given this ordering, we can generate a sample $X = (X_1, \ldots, X_n)$ from the *joint* distribution by sampling *each element* in topological order



Ancestral sampling:

1. Sample from $\Pr(X_1)$
2. Sample from $\Pr(X_2)$
3. Sample from $\Pr(X_3)$
4. Sample from $\Pr(X_4|X_1, X_2)$
5. Finally, sample from $\Pr(X_5|X_1, X_2, X_3)$

=> $\{X_1, X_2, X_3, X_4, X_5\}$ is a sample from the joint distribution.

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|pa(X_i))$$

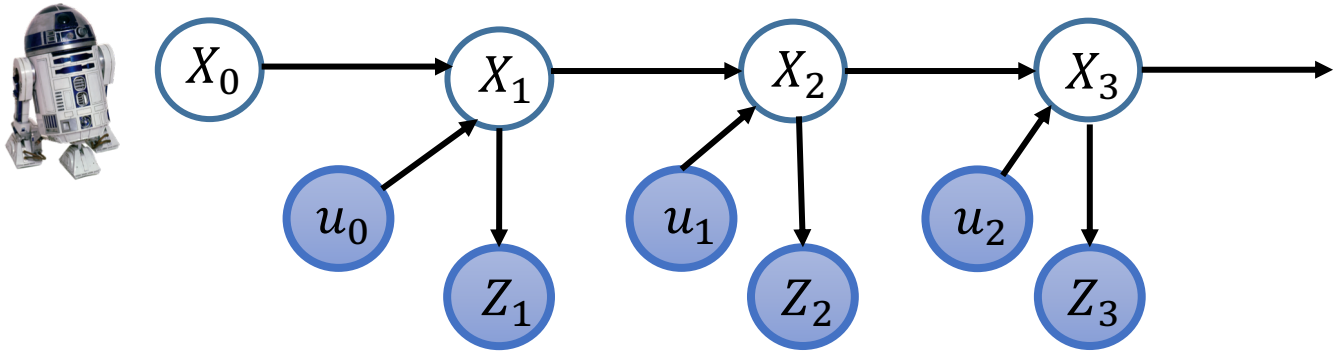
State estimation and dynamic Bayes nets

Consider a navigating robot. Starting at some initial position X_0, the robot repeatedly:

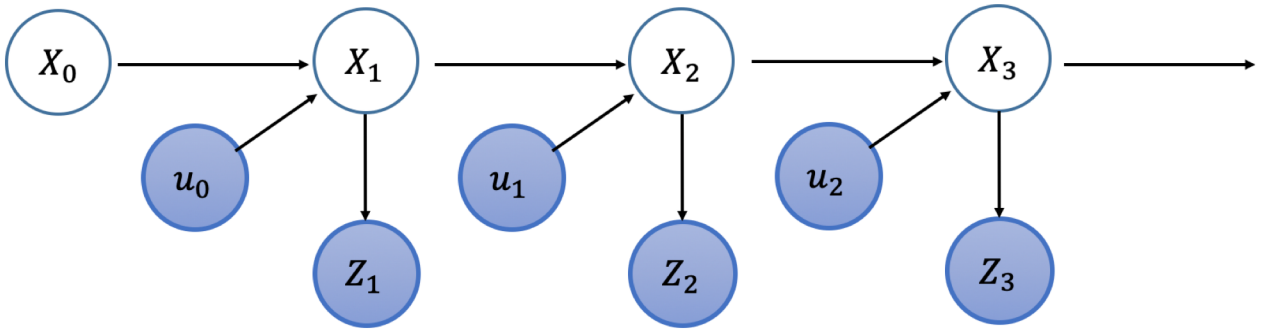
- Collects an observation Z_t about its current position X_t
- Applies a control u_t to move to its next position

Goal: Estimate a belief $p(X_t|u_{0:t-1}, Z_{1:t})$ over the robot's current position X_t, given all previous controls $u_{0:t-1}$ and observations $Z_{1:t}$

We can model this scenario using a dynamic Bayes net



State estimation and dynamic Bayes nets

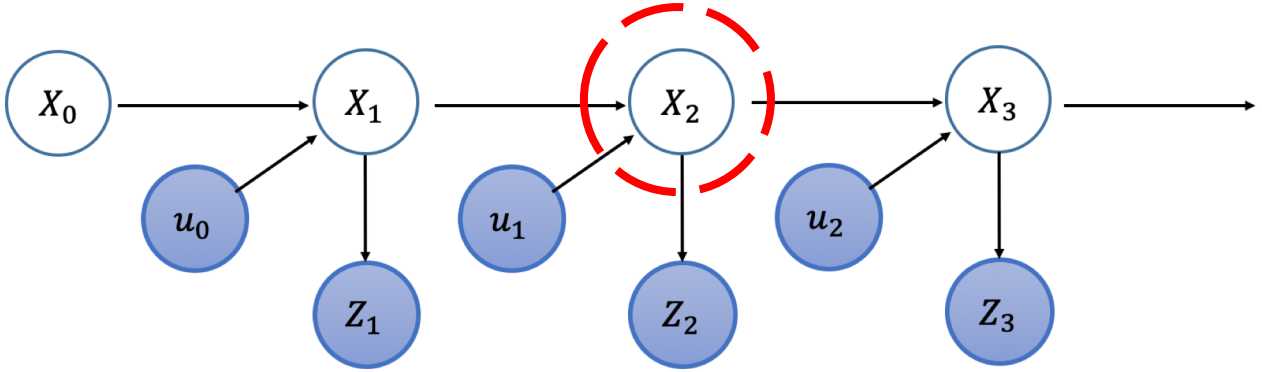


Problem: The size of this Bayes net is increasing as the robot explores. ($p(X_t|u_{0:t-1}, Z_{1:t})$)

This is a problem for:

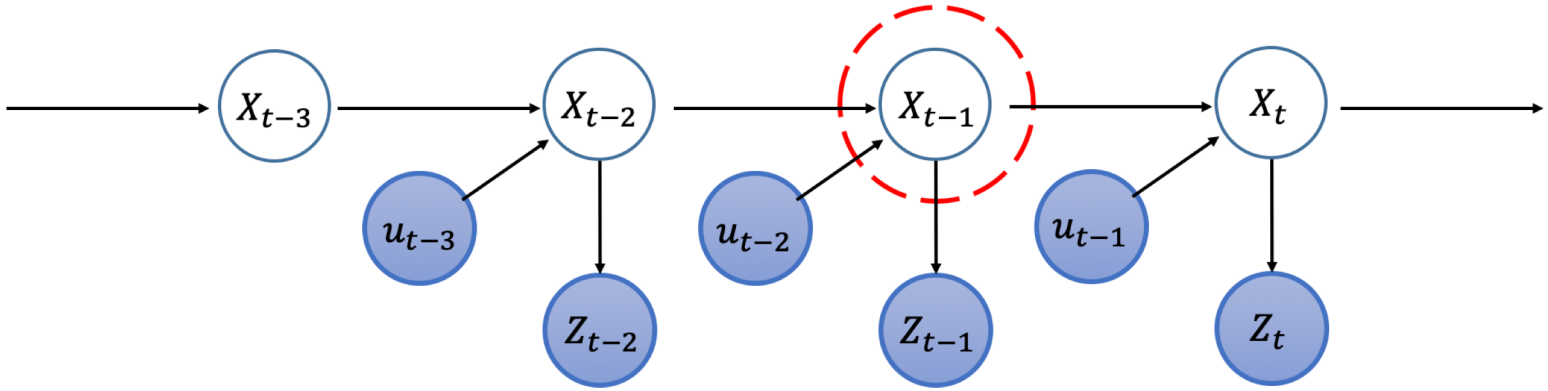
- **Storage**: Need to keep track of more variables, measurements, motor commands
- **Compute**: Need to solve a larger problem to estimate current state

State estimation and dynamic Bayes nets



But: Notice that by the local Markov property, X_3 is conditionally independent of X_0 and X_1 given X_2!

State estimation and dynamic Bayes nets



More generally: X_t is conditionally independent of $X_{0:t-2}$ given X_{t-1}.

This suggests that we need to only know $p(X_{t-1}|u_{0:t-2}, Z_{1:t-1})$ – the belief over the *previous* position – to compute $p(X_t|u_{0:t-1}, Z_{1:t})$ – the belief at the *current* position.

If so, we could discard information about all of the earlier states $X_{1:t-2}$! (This would be a *huge* savings!)

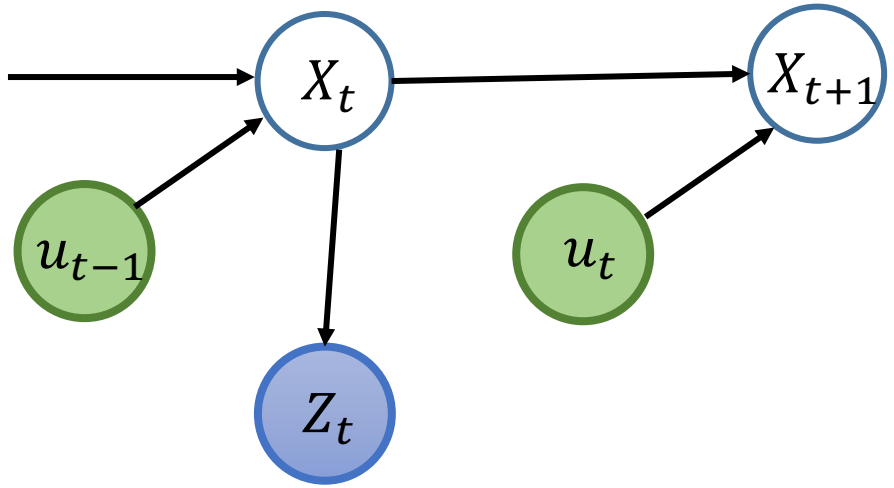
Let's see if we can make this precise …

Recursive Bayesian estimation

Suppose that we *already have* $p(X_t|u_{0:t-1}, Z_{1:t})$, the belief for the robot's current position X_t given all previous controls $u_{0:t-1}$ and measurements $Z_{1:t}$.

Now we apply the command u_t to move to the next position.

Q: What is $p(X_{t+1}|u_{0:t}, Z_{1:t})$, the belief for the *next* position X_{t+1} given the control u_t?



$$p(X_{t+1}, X_t|u_{0:t}, Z_{1:t}) = p(X_{t+1}|X_t, u_{0:t}, Z_{1:t}) \cdot p(X_t|u_{0:t}, Z_{1:t}) \quad \text{(Chain Rule of probability)}$$

$$= p(X_{t+1}|X_t, u_t) \cdot p(X_t|u_{0:t-1}, Z_{1:t}) \quad \text{(Local Markov property)}$$

Motion model — Belief over prior state

Now we can marginalize over X_t! $\quad p(X_{t+1}|u_{0:t}, Z_{1:t}) = \int p(X_{t+1}, X_t|u_{0:t}, Z_{1:t})\ dX_t$

$$= \int p(X_{t+1}|X_t, u_t) \cdot p(X_t|u_{0:t-1}, Z_{1:t})\ dX_t$$

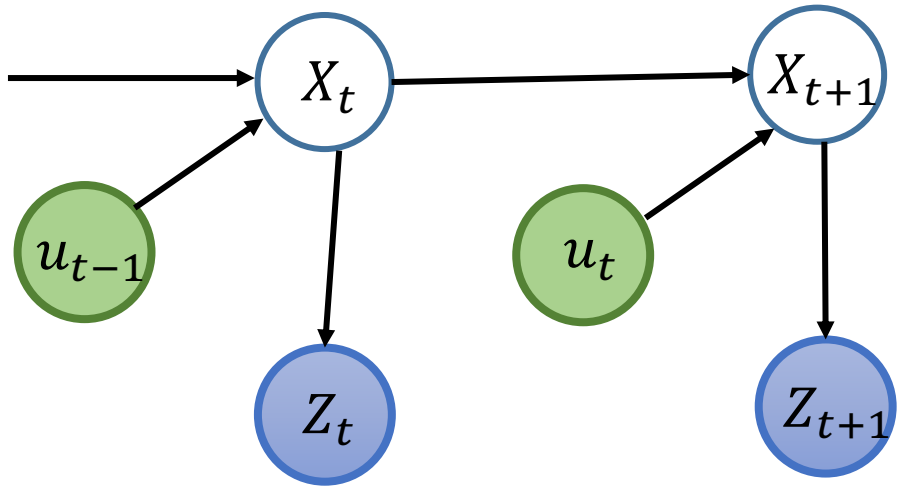
Recursive Bayes estimation

$$P(X \mid Y) = \frac{P(Y \mid X)\, P(X)}{P(Y)}$$

likelihood (of X given Y) · prior · posterior · evidence

Now suppose that we take another measurement Z_{t+1} at position X_{t+1}.

Q: What is $p(X_{t+1}|u_{0:t}, Z_{1:t+1})$, the *updated* belief for X_{t+1} after incorporating the latest measurement Z_{t+1}?



$$p(X_{t+1}|u_{0:t}, Z_{1:t+1}) = \frac{p(Z_{t+1}|X_{t+1}, u_{0:t}, Z_{1:t})p(X_{t+1}|u_{0:t}, Z_{1:t})}{p(Z_{t+1}|\, u_{0:t}, Z_{1:t})}$$

(Conditional Bayes' Rule)

Measurement model

Predicted belief for X_{t+1} (prior)

$$= \frac{p(Z_{t+1}|X_{t+1})p(X_{t+1}|u_{0:t}, Z_{1:t})}{p(Z_{t+1}|\, u_{0:t}, Z_{1:t})}$$

(Local Markov property)

Evidence calculation:
(marginal prob. of observing Z_{t+1} Irrespective of X_{t+1})

$$p(Z_{t+1}|\, u_{0:t}, Z_{1:t}) = \int p(Z_{t+1}|X_{t+1})p(X_{t+1}|u_{0:t}, Z_{1:t})\; dX_{t+1}$$

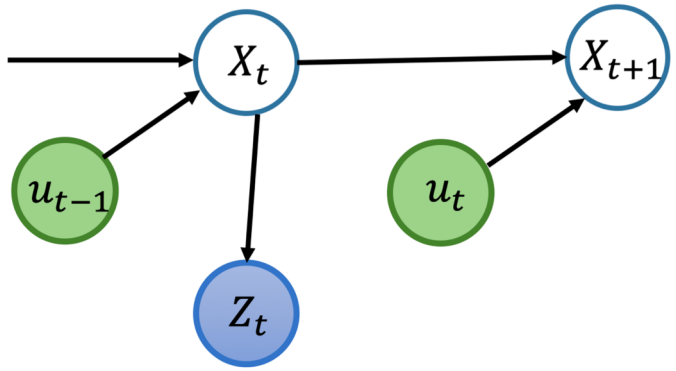
The Bayes filter

The Bayes Filter is a simple and efficient algorithm for recursive Bayesian estimation in dynamic Bayes nets

Given:

- Prior $p(X_0)$ for the initial state X_0
- Sequence of controls $u_{0:t-1}$ and sensor observations $Z_{1:t}$

Find: $p(X_t|u_{0:t-1}, Z_{1:t})$, the posterior belief over the *current* state X_t



Bayes Filter: For t = 1, 2 … repeat the following operations:

- **Predict** belief for current state X_t given previous control u_{t-1}:

$$p(X_t|u_{0:t-1}, Z_{1:t-1}) = \int p(X_t|X_{t-1}, u_{t-1}) \cdot p(X_{t-1}|u_{0:t-2}, Z_{1:t-1})\ dX_{t-1}$$

- **Update** belief after incorporating measurement Z_t at current state X_t:

$$p(X_t|u_{0:t-1}, Z_{1:t}) = \frac{p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})}{\int p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})\ dX_t}$$

Why the Bayes Filter is super cool?

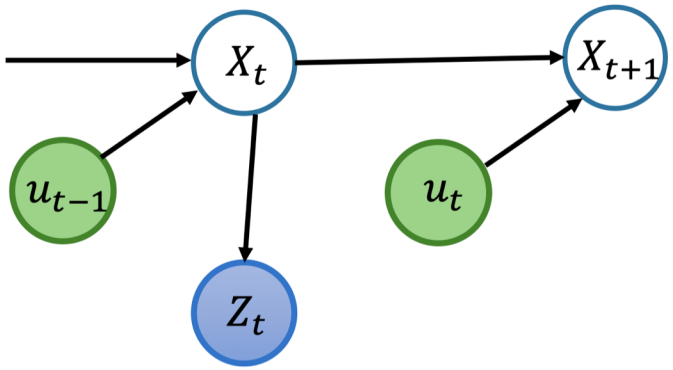
Bayes Filter: For t = 1, 2 … repeat the following operations:

- **Predict** belief for current state X_t given previous control u_{t-1}:

$$p(X_t|u_{0:t-1}, Z_{1:t-1}) = \int p(X_t|X_{t-1}, u_{t-1}) \cdot p(X_{t-1}|u_{0:t-2}, Z_{1:t-1}) \; dX_{t-1}$$

- **Update** belief after incorporating measurement Z_t at current state X_t:

$$p(X_t|u_{0:t-1}, Z_{1:t}) = \frac{p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})}{\int p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1}) \; dX_t}$$



Computational efficiency: While the total # of states, controls, and measurements in the dynamic Bayes net *grows linearly* in time, the Bayes Filter is:

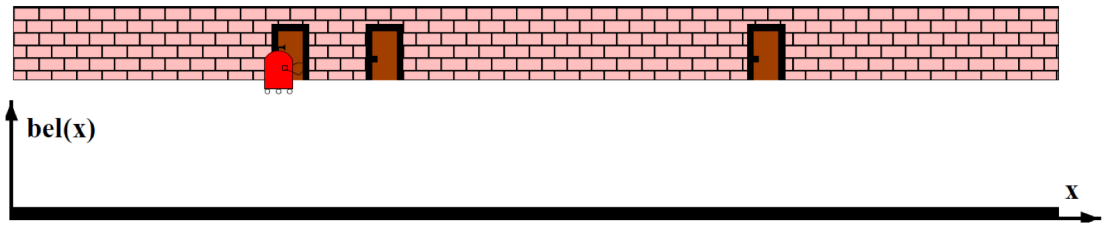
Constant space:

- The Bayes Filter only maintains a belief over the current state X_t – this is a distribution of a *fixed size*
- The control u_t and measurement Z_t are only used for prediction and updating in timestep *t* (i.e. a *single step*)
  ⇒We don't need to remember these after they're applied!

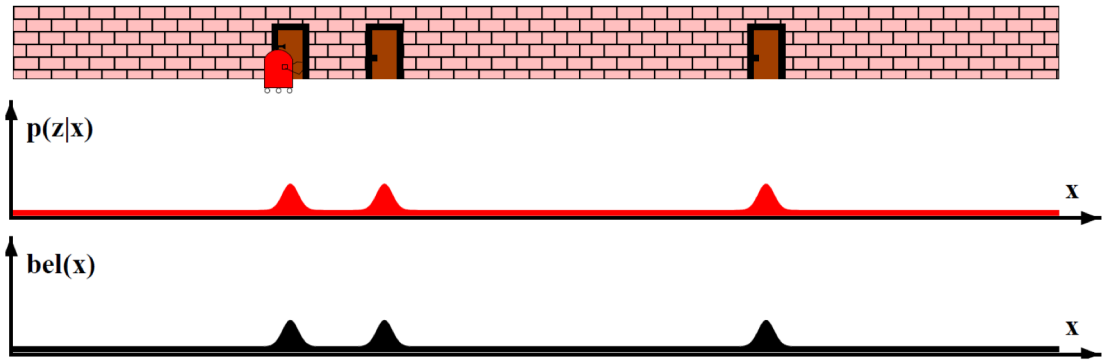
Constant time: Each prediction and update step involves computing integrals over distributions of a *fixed size and type*

⇒ These are *constant-time* operations. This is super important for (real-time!) robotics applications

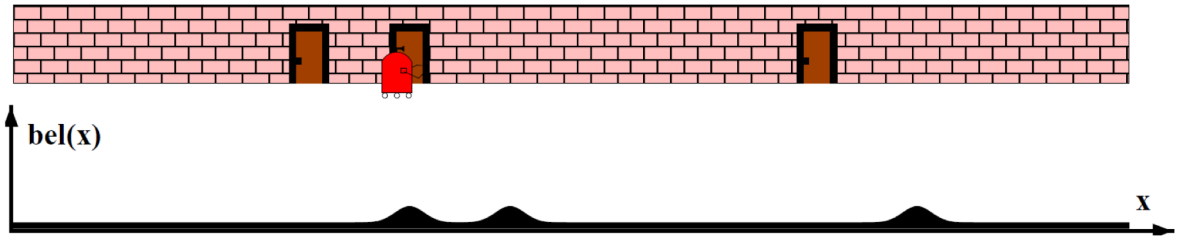
Example application: Robot localization



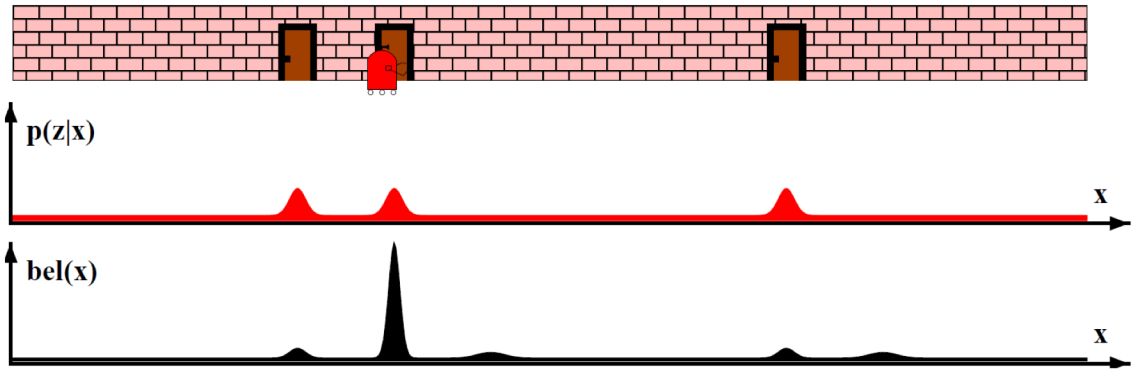
Initial belief



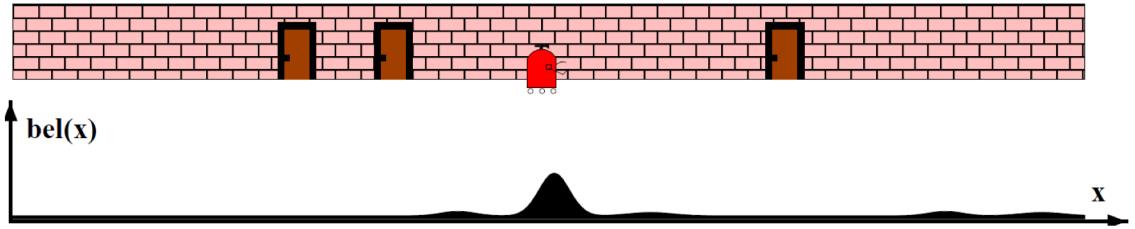
Measurement



Move right



Measurement



Move right

Summary

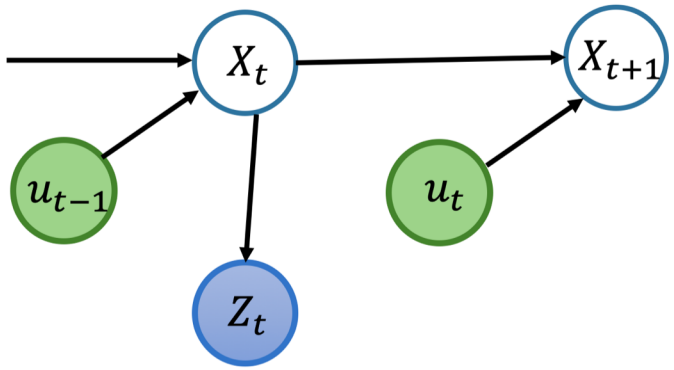
Bayes Filter: For t = 1, 2 … repeat the following operations:

- **Predict** belief for current state X_t given previous control u_{t-1}:

$$p(X_t|u_{0:t-1}, Z_{1:t-1}) = \int p(X_t|X_{t-1}, u_{t-1}) \cdot p(X_{t-1}|u_{0:t-2}, Z_{1:t-1}) \; dX_{t-1}$$

- **Update** belief after incorporating measurement Z_t at current state X_t:

$$p(X_t|u_{0:t-1}, Z_{1:t}) = \frac{p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})}{\int p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1}) \; dX_t}$$



This time:

- Probabilistic robotics
- Bayesian networks
- Recursive Bayesian estimation and the Bayes Filter

Next time: Three specific ways to implement the Bayes Filter