AI for *Robotics*

(*Just Bayes Filter and a bit more*)

Guest Lecture

Shihan Lu

Haptics Robotics and Virtual Interaction Lab

Department of Computer Science

Robotics at USC

Robotics and Autonomous Systems Center
(rasc.usc.edu)

Interaction Lab

RASC

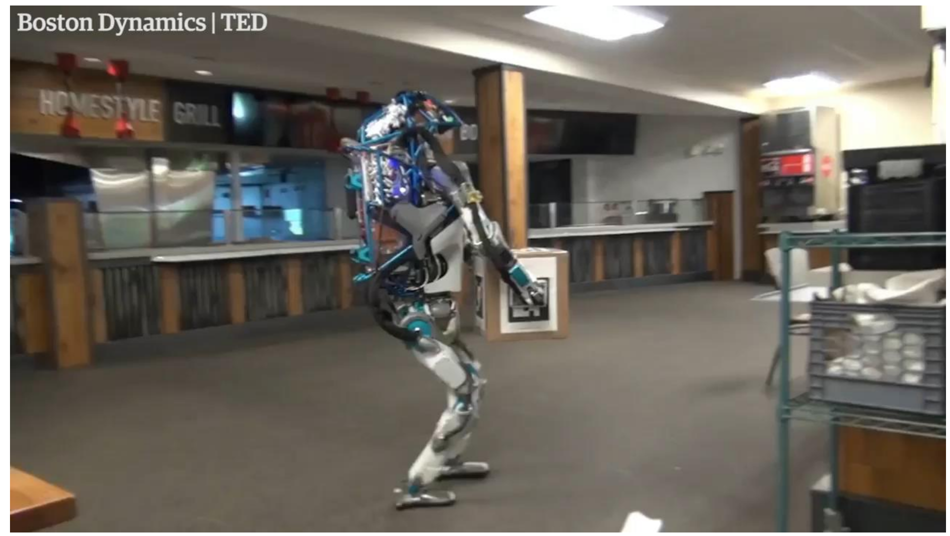
HARVI
HAptics Robotics and Virtual Interaction Lab

USC Robotic Embedded Systems Laboratory

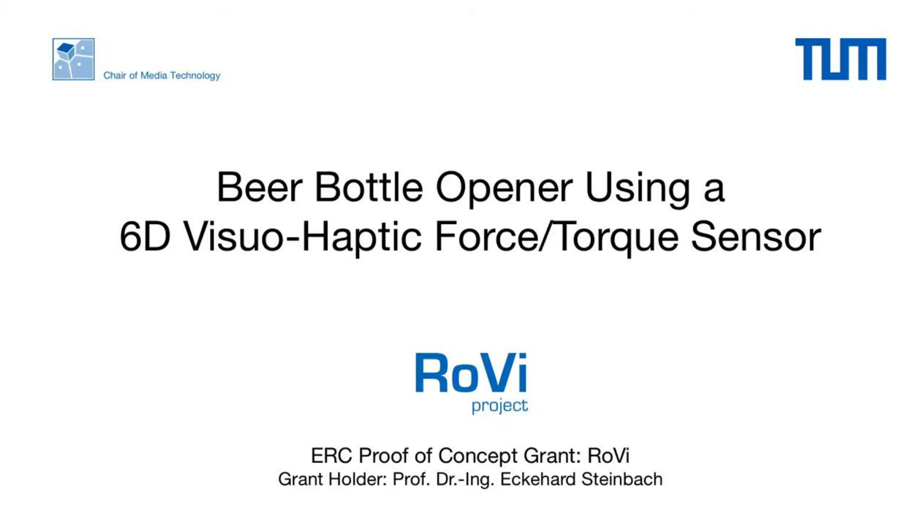
ICAROS

What is a Robot?

What is a Robot?

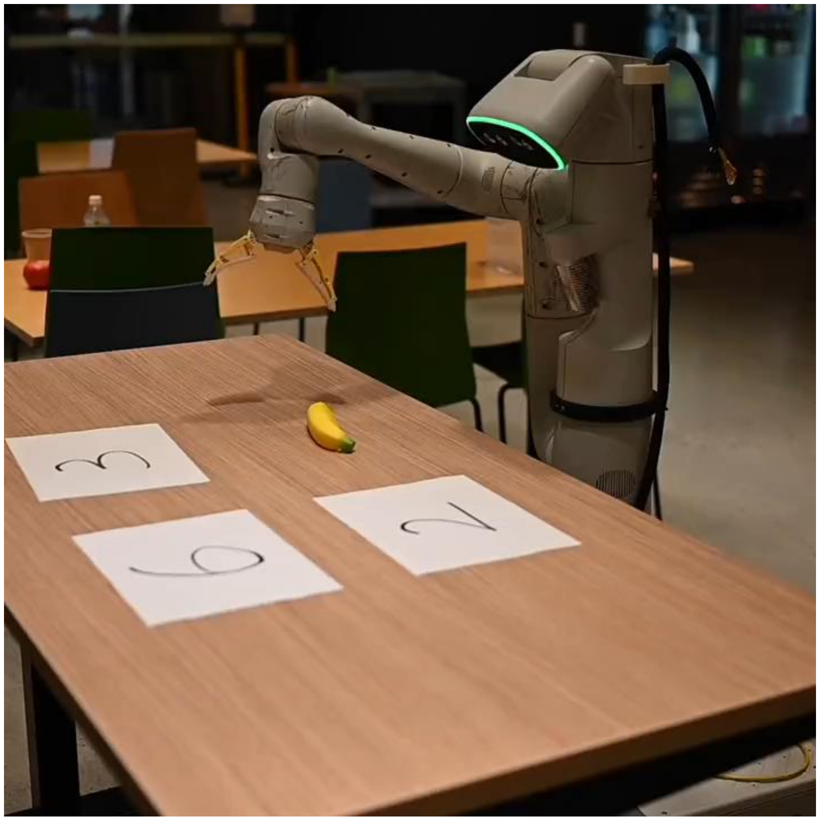


What is a Robot?

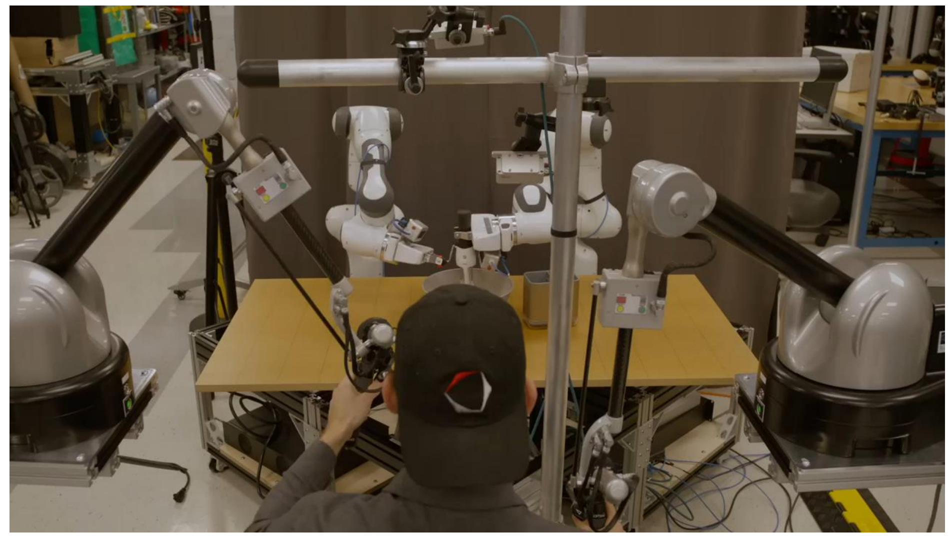


What is a Robot?

Robotics with LLM



Robotics with Learning from Demonstration



What is a Robot?

- Is robot just a shell of computer vision and natural language?
- Will robotics be solved by CV and NLP?
- Generalist vs Specialist
- Data-driven vs physics-based
- What is your thought on future of robotics?

What is a Robot?

Do it yourself!

What is a Robot?

A robot is an *autonomous* system
which exists in the *physical* world,
can *sense* its environment,
and *acts* in it
to *achieve some goals*.

au·ton·o·mous *adjective* \ȯ-ˈtä-nə-məs\

: existing or acting separately from other things or people

: having the power or right to govern itself

Full Definition of AUTONOMOUS

Cite! g+1 Like

1 : of, relating to, or marked by autonomy

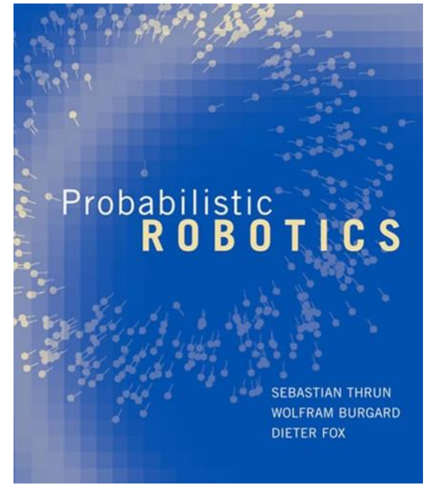
2 a : having the right or power of self-government

b : undertaken or carried on without outside control **:** SELF-CONTAINED <an *autonomous* school system>

Sensing and *Estimation* in Robotics

Contents adapted from Prof. Gaurav Sukhatme's lecture on *Fundamentals of Robotics* and Book *"Probabilistic Robotics"*

Probabilistic Robotics



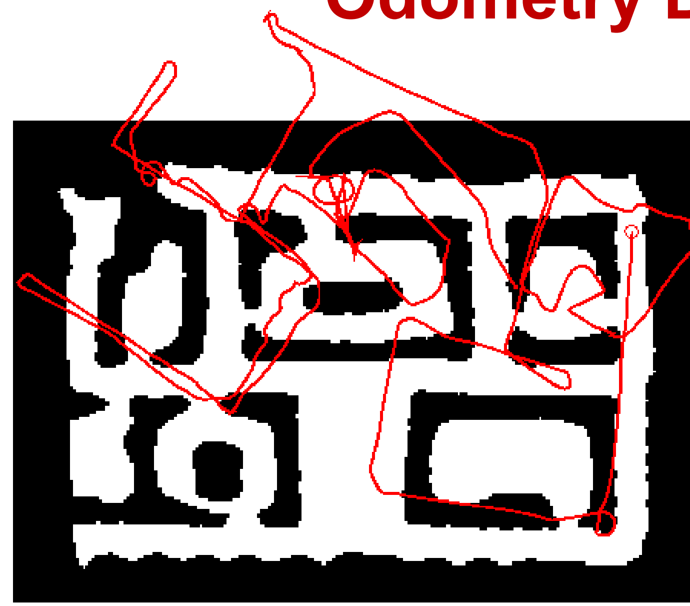
Key idea: Explicit representation of uncertainty using the calculus of probability theory

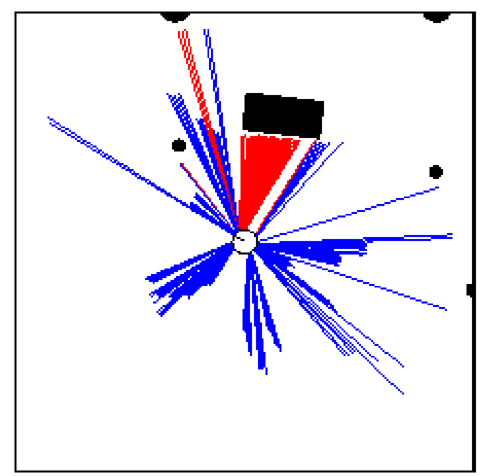
- Perception = state estimation
- Action = utility optimization

Uncertainty is Inherent/Fundamental

- Uncertainty arises from four major factors:
  - Environment is stochastic, unpredictable
  - Robot's actions are stochastic
  - Sensors are limited and noisy
  - Models are inaccurate, incomplete

Odometry Data





Range Data

Advantages and Pitfalls

- Can accommodate inaccurate models
- Can accommodate imperfect sensors
- Robust in real-world applications
- Best known approach to many hard robotic problems
- Computationally demanding
- False assumptions
- Approximate

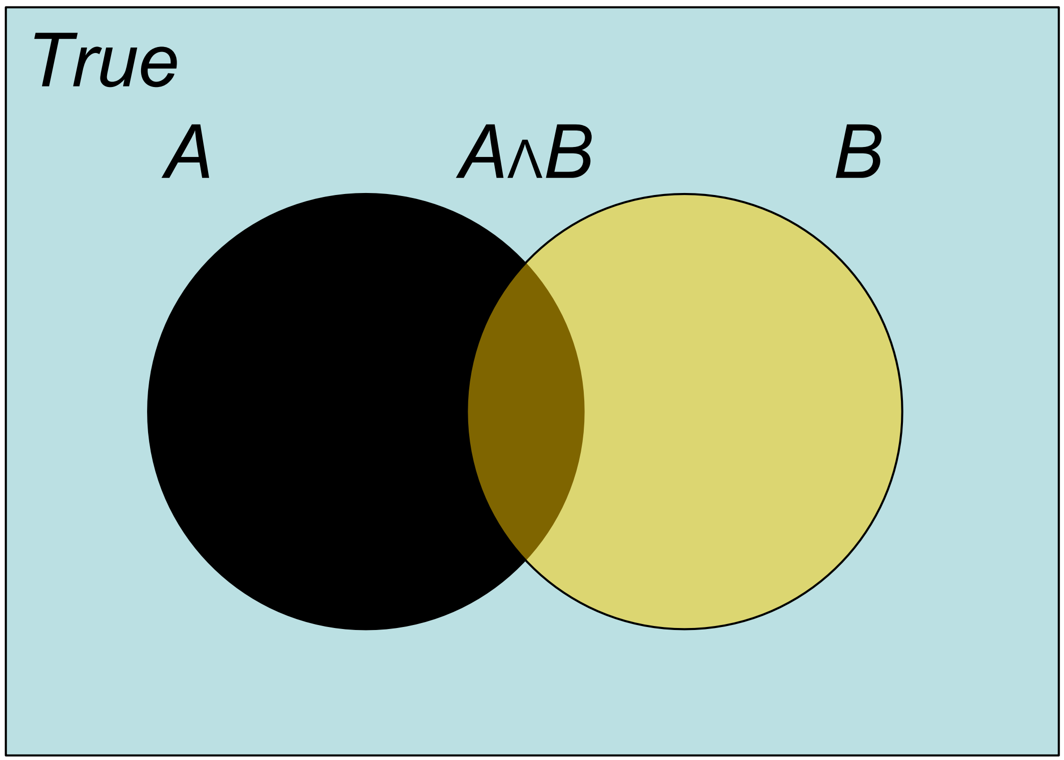
Axioms of Probability Theory

Pr(*A*) denotes probability that proposition *A* is true.

- $0 \leq \Pr(A) \leq 1$
- $\Pr(True) = 1$, $\Pr(False) = 0$
- $\Pr(A \vee B) = \Pr(A) + \Pr(B) - \Pr(A \wedge B)$

A Closer Look at Axiom 3

$$\Pr(A \vee B) = \Pr(A) + \Pr(B) - \Pr(A \wedge B)$$



Using the Axioms

$$
\begin{aligned}
\Pr(A \vee \neg A) &= \Pr(A) + \Pr(\neg A) - \Pr(A \wedge \neg A) \\
\Pr(\mathit{True}) &= \Pr(A) + \Pr(\neg A) - \Pr(\mathit{False}) \\
1 &= \Pr(A) + \Pr(\neg A) - 0 \\
\Pr(\neg A) &= 1 - \Pr(A)
\end{aligned}
$$

Discrete Random Variables

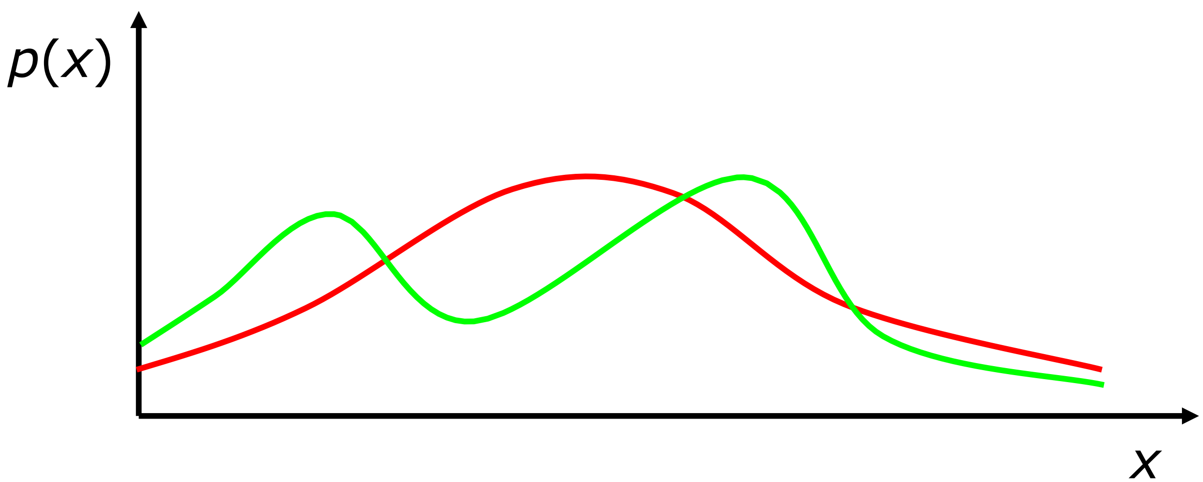
- X denotes a random variable.
- X can take on a finite number of values in $\{x_1, x_2, \ldots, x_n\}$.
- $P(X=x_i)$, or $P(x_i)$, is the probability that the random variable X takes on value x_i.
- $P(\cdot)$ is called probability mass function.
- E.g. $P(\text{Room}) = \{0.7, 0.2, 0.08, 0.02\}$

Continuous Random Variables

- X takes on values in the continuum.
- $p(X=x)$, or $p(x)$, is a probability density function.

$$\Pr(x \in [a,b]) = \int_a^b p(x)\,dx$$

- E.g.



Joint and Conditional Probability

- $P(X = x \text{ and } Y = y) = P(x, y)$

- If X and Y are independent then

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

- $P(x \mid y)$ is the probability of x given y

  $$P(x \mid y) = P(x, y) \,/\, P(y)$$

  $$P(x, y) = P(x \mid y)\, P(y)$$

- If X and Y are independent then

  $$P(x \mid y) = P(x)$$

Law of Total Probability

Discrete case

$$\sum_x P(x) = 1$$

Marginalization:

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

Conditioning:

$$P(x) = \sum_y P(x \mid y) P(y)$$

Continuous case

$$\int p(x)dx = 1$$

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

$$p(x) = \int p(x \mid y)p(y)dy$$

Reverend Thomas Bayes, FRS (1702-1761)

Clergyman and mathematicians first used probability inductively and established a mathematical basis for probability inference

Bayes Formula

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

$$\Downarrow$$

$$\boxed{P(x|y) = \frac{P(y|x)P(x)}{P(y)} = \frac{\text{likehood prior}}{\text{evidence}}}$$

Normalization

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

$$\eta = P(y)^{-1} = \frac{1}{\sum_x P(y|x)P(x)}$$

Algorithm:

$$\forall x: \mathrm{aux}_{x|y} = P(y|x)P(x)$$

$$\eta = \frac{1}{\sum_x \mathrm{aux}_{x|y}}$$

$$\forall x: P(x|y) = \eta\ \mathrm{aux}_{x|y}$$

Conditioning

- Total probability:

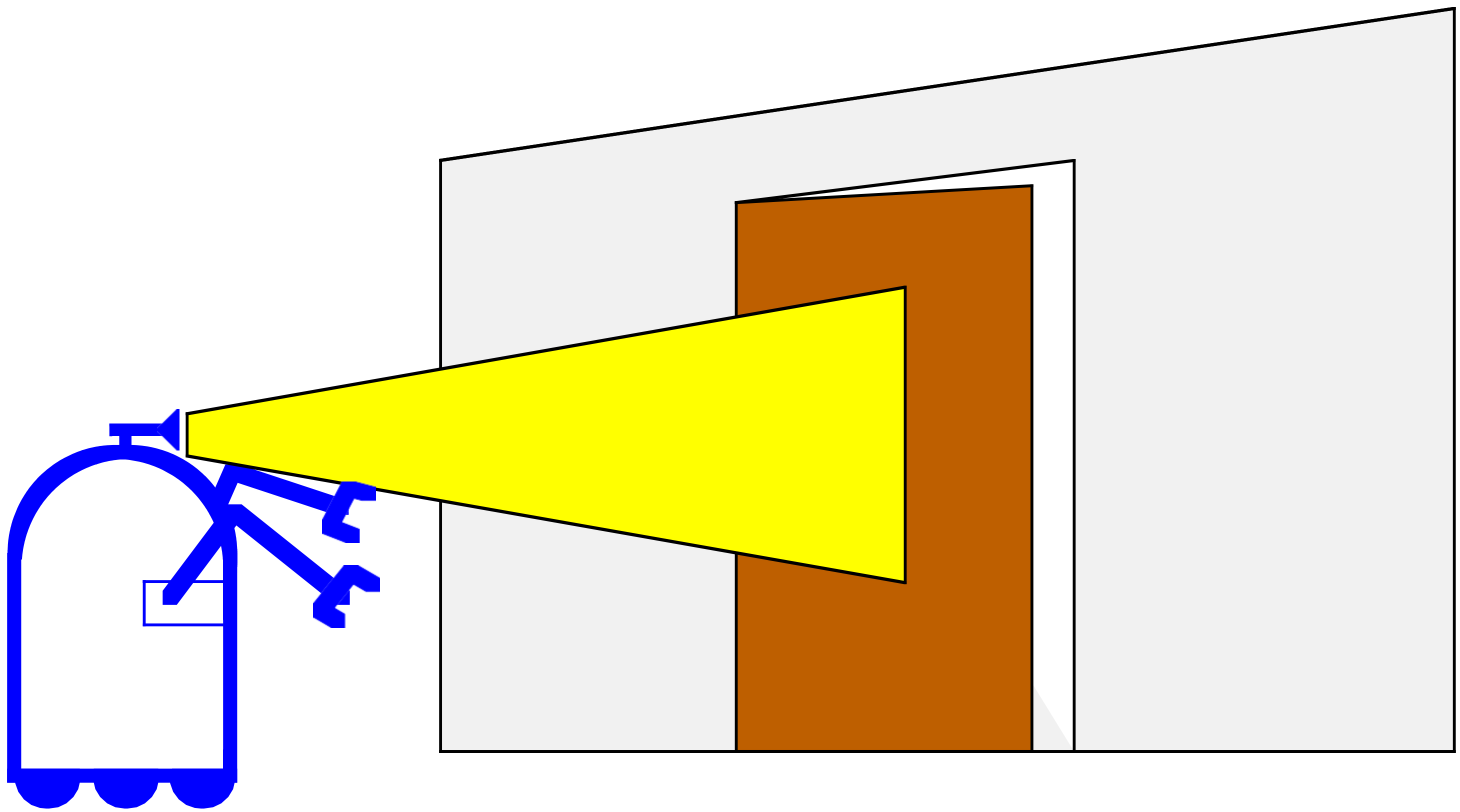
$$P(x|y) = \int P(x|y,z)P(z|y)dz$$

- Bayes rule and background knowledge:

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

Example of Sensing

- Suppose a robot obtains sensor measurement about <u>z = "*door is perceived as open*"</u>
- What is $P(open|z)$?



Causal vs. Diagnostic Reasoning

- $P(open|z)$ is diagnostic.
- $P(z|open)$ is causal, and we call it sensor model.
- Often causal knowledge is easier to obtain.
- Bayes rule allows us to use causal knowledge:

Count frequencies!

$$P(open|z) = \frac{P(z|open)P(open)}{P(z)}$$

Example

- $P(z \mid open) = 0.6 \qquad P(z \mid \neg open) = 0.3$
- $P(open) = P(\neg open) = 0.5$

$$P(open|z) = \frac{P(z|open)P(open)}{P(z|open)P(open) + P(z|\neg open)P(\neg open)}$$

$$= \frac{0.6 \times 0.5}{0.6 \times 0.5 + 0.3 \times 0.5} = \frac{2}{3} = 0.67$$

- *z* raises the probability that the door is open.

Combining Evidence

- Suppose our robot obtains another observation z_2.
- How can we integrate this new information?
- More generally, how can we estimate $P(x \mid z_1,...,z_n)$?

Example: Second Measurement

- $P(z_2 \mid open) = 0.5 \qquad P(z_2 \mid \neg open) = 0.6$
- $P(open \mid z_1) = 2/3$

$$P(open|z_2, z_1) = \frac{P(z_2|open)P(open|z_1)}{P(z_2|open)P(open|z_1) + P(z_2|\neg open)P(\neg open|z_1)}$$

$$= \frac{\frac{1}{2} \times \frac{2}{3}}{\frac{1}{2} \times \frac{2}{3} + \frac{3}{5} \times \frac{1}{3}} = \frac{5}{8} = 0.625$$

- z_2 lowers the probability that the door is open.

Actions

- Often the world is **dynamic** since
  - **actions carried out by the robot**,
  - **actions carried out by other agents**,
  - or just the **time** passing by

  change the world.

- How can we **incorporate** such **actions**?

Typical Actions

- The robot **turns its wheels** to move
- The robot **uses its manipulator** to grasp an object
- Plants **grow** over **time**…

- Actions are **never carried out with absolute certainty**.
- In contrast to measurements, **actions generally increase the uncertainty**.

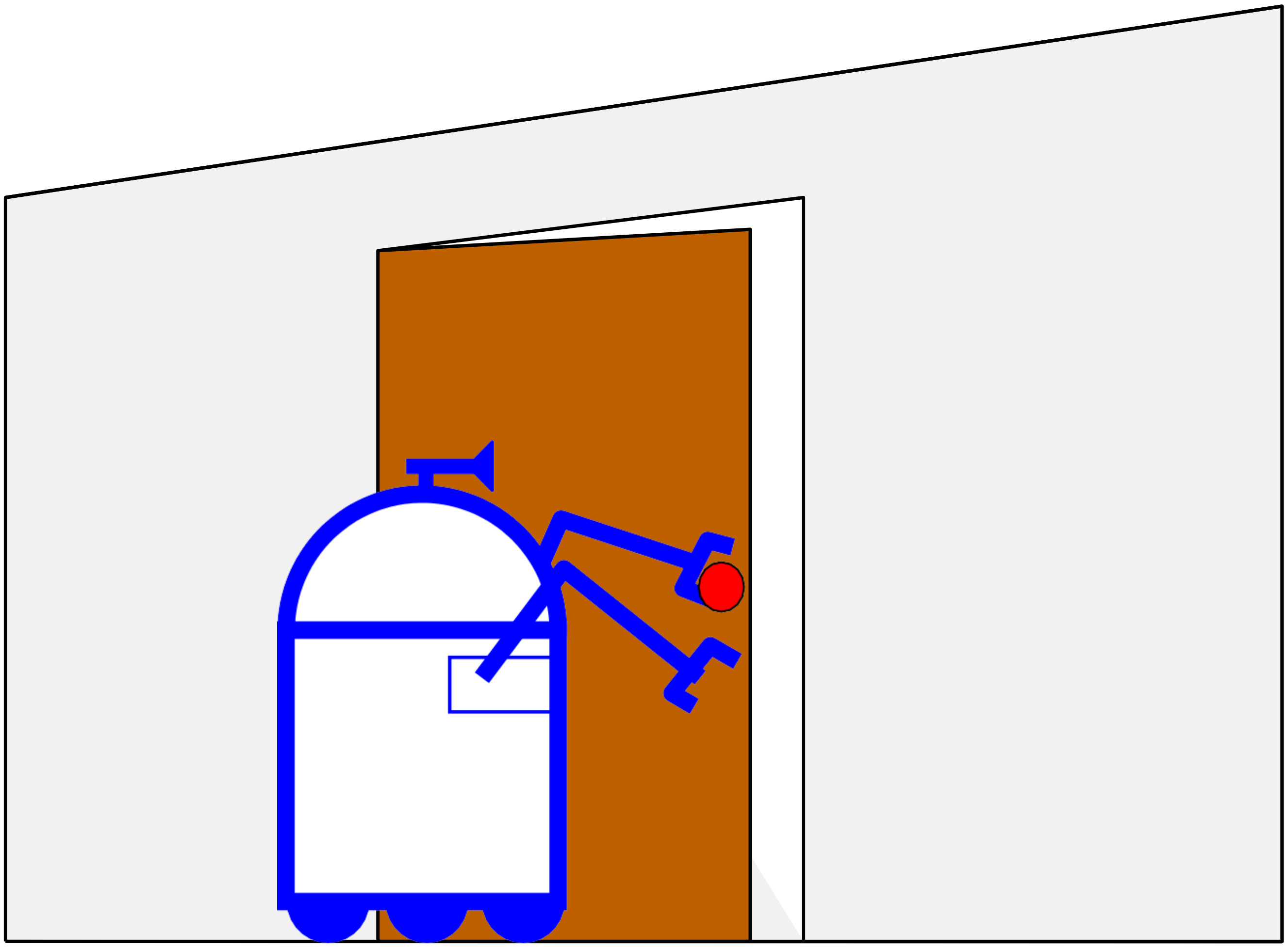
Modeling Actions

- To incorporate the outcome of an action *u* into the current “belief”, we use the conditional pdf

$$P(x|u,x')$$

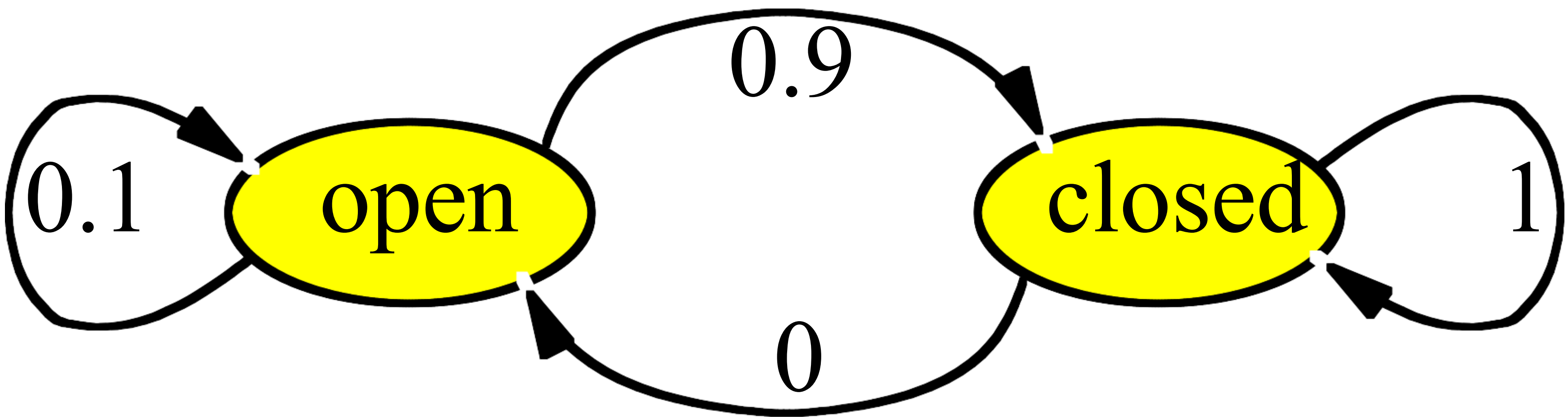
- This term specifies the pdf that executing *u* changes the state from *x’* to *x*.

Example: Closing the door



State Transitions (Action model)

$P(x|u,x')$ for u = “close door”:



If the door is open, the action “close door” succeeds in 90% of all cases.

Integrating the Outcome of Actions

Continuous case:

$$P(x|u) = \int P(x|u,x')P(x')dx'$$

Discrete case:

$$P(x|u) = \sum_{x'} P(x|u,x')P(x')$$

Example: Resulting belief after action *u* = "*close door*"

P(*open*) = 5/8
P(*closed*) = 3/8
From the calculation of the last example

$$P(closed \mid u) = \sum P(closed \mid u, x')P(x')$$

$$= P(closed \mid u, open)P(open)$$

$$+ P(closed \mid u, closed)P(closed)$$

$$= \frac{9}{10} \times \frac{5}{8} + \frac{1}{1} \times \frac{3}{8} = \frac{15}{16}$$

$$P(open \mid u) = \sum P(open \mid u, x')P(x')$$

$$= P(open \mid u, open)P(open)$$

$$+ P(open \mid u, closed)P(closed)$$

$$= \frac{1}{10} \times \frac{5}{8} + \frac{0}{1} \times \frac{3}{8} = \frac{1}{16}$$

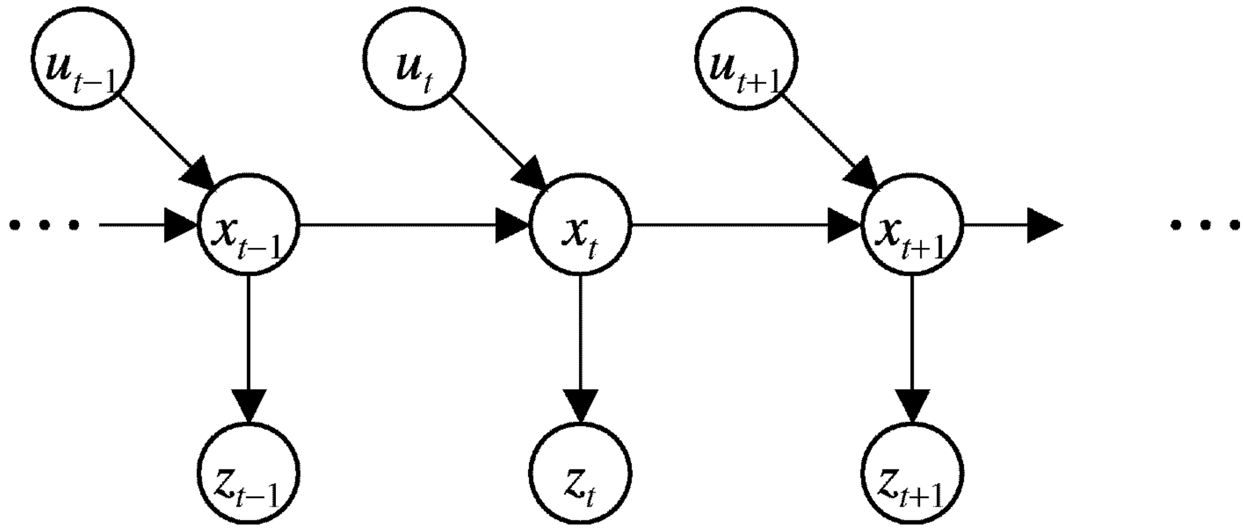
$$= 1 - P(closed \mid u)$$

Bayes Filters: Framework

- **Given:**
  - Stream of sensor observations z and actions u:
    $$d_t = \{ u_t, z_t \}$$
  - Sensor model $P(z|x)$
  - Action model $P(x|u,x')$
  - Prior probability of the state $P(x)$
- **Wanted:**
  - Estimate of the state x of a dynamic system
  - The posterior of the state is also called **Belief**:

$$\begin{aligned} Bel(x_t) &= P(x_t \mid d_1, \ldots, d_t) \\ &= P(x_t \mid u_1, z_1, \ldots, u_t, z_t) \end{aligned}$$

Markov Assumption



Correction-> $P(z_t \mid x_{1:t}, z_{1:t-1}, u_{1:t}) = P(z_t \mid x_t)$ Sensor Model

Prediction-> $P(x_t \mid x_{1:t-1}, z_{1:t-1}, u_{1:t}) = P(x_t \mid x_{t-1}, u_t)$ Action Model

Underlying Assumptions

- Static world
- Independent noise
- Perfect model, no approximation errors

Recursive Bayesian Updating

- Perceptual (Observation) Update for z_t

Markov assumption: z_t is independent of $d_1,\ldots,d_{t-1}, u_t$ **if we know x_t**

$$d_t = \{ u_t, z_t \}$$

$$\begin{aligned}
Bel(x_t) &= P(x_t | d_1, \ldots, d_{t-1}, d_t) \\
&= P(x_t | d_1, \ldots, d_{t-1}, u_t, z_t) \\
&= \frac{P(z_t | x_t, d_1, \ldots, d_{t-1}, u_t) P(x_t | d_1, \ldots, d_{t-1}, u_t)}{P(z_t | d_1, \ldots, d_{t-1}, u_t)} \\
&= \frac{P(z_t | x_t) P(x_t | d_1, \ldots, d_{t-1}, u_t)}{P(z_t | d_1, \ldots, d_{t-1}, u_t)} \\
&= \eta P(z_t | x_t) P(x_t | d_1, \ldots, d_{t-1}, u_t) \\
&= \eta P(z_t | x_t) \boxed{\overline{Bel}(x_t)}
\end{aligned}$$

Intermediate Belief before receiving z_t

Recursive Bayesian Updating

- Action (Dynamic) Update for u_t

$$d_t = \{ u_t , z_t \}$$

$$\boxed{\overline{Bel}(x_t)} = P(x_t|d_1, ..., d_{t-1}, u_t)$$

Continue from last page

$$= \int P(x_t|d_1, ..., d_{t-1}, u_t, x_{t-1})P(x_{t-1}|d_1, ..., d_{t-1}, u_t)dx_{t-1}$$

$$= \int P(x_t|u_t, x_{t-1})P(x_{t-1}|d_1, ..., d_{t-1})dx_{t-1}$$

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

Belief of last time stamp

Bayes Filter Algorithm

Algorithm **Bayes_Filter**($Bel(x),d$):

$\eta = 0$

If d is a perceptual data z then

    For all x do

$$Bel'(x) = P(z|x)Bel(x)$$

$$\eta = \eta + Bel'(x)$$

    For all x do

$$Bel'(x) = \eta^{-1} Bel'(x)$$

Else if d is an action data u then

    For all x do

$$Bel'(x) = \int P(x|u,x')Bel(x')dx'$$

Return $Bel'(x)$

Discrete Bayes Filter Algorithm

1. Algorithm **Discrete_Bayes_Filter**($Bel(x), d$):
2. $\eta = 0$
3. If d is a perceptual data z then
4. For all x do
5. $Bel'(x) = P(z|x)Bel(x)$
6. $\eta = \eta + Bel'(x)$
7. For all x do
8. $Bel'(x) = \eta^{-1} Bel'(x)$
9. Else if d is an action data u then
10. For all x do
11. $Bel'(x) = \sum_{x'} P(x|u,x')Bel(x')$
12. Return $Bel'(x)$

Bayes Filter

z = observation

u = action

x = state

$$\boxed{Bel(x_t)} = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Bayes $$= \eta \; P(z_t \mid x_t, u_1, z_1, \ldots, u_t) \, P(x_t \mid u_1, z_1, \ldots, u_t)$$

Markov $$= \eta \; P(z_t \mid x_t) \, P(x_t \mid u_1, z_1, \ldots u_t)$$

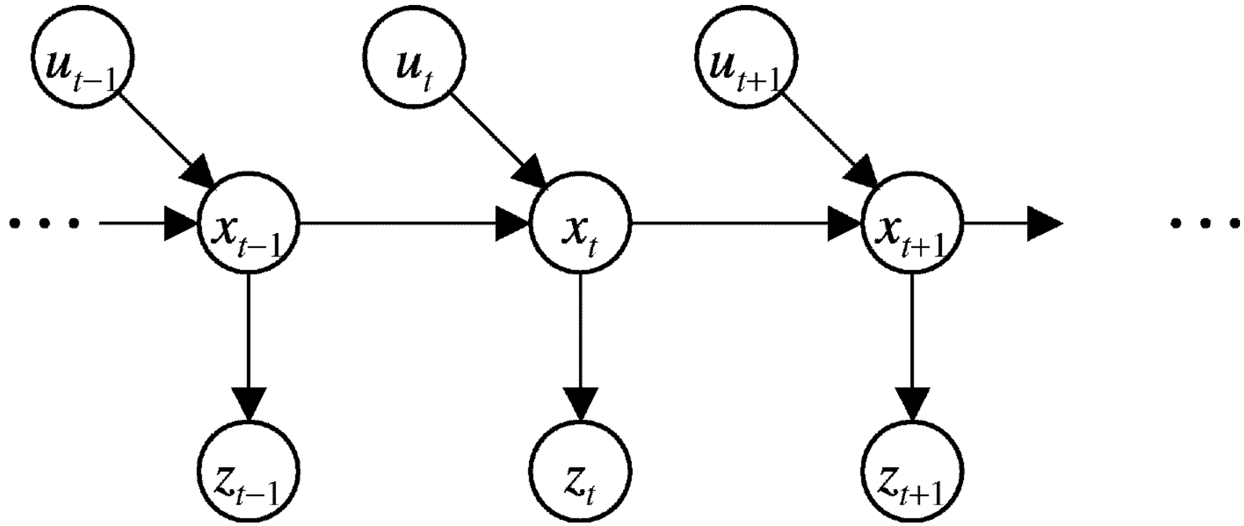
Total prob. $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots u_t) \, dx_{t-1}$$

Markov $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \, P(x_{t-1} \mid u_1, z_1, \ldots u_t) \, dx_{t-1}$$

Markov $$= \eta \, P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \, P(x_{t-1} \mid u_1, z_1, \ldots z_{t-1}) \, dx_{t-1}$$

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

Recap – Bayes Filter

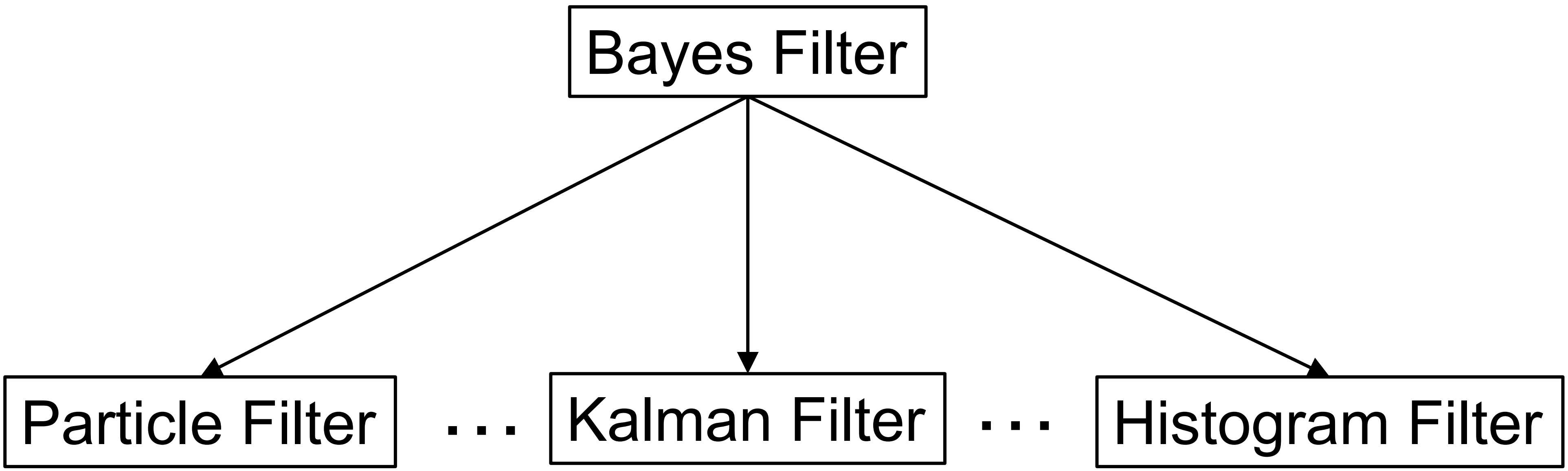


$P(z_t \mid x_{1:t}, z_{1:t\text{-}1}, u_{1:t}) = P(z_t \mid x_t)$ **Sensor Model**

$P(x_t \mid x_{1:t\text{-}1}, z_{1:t\text{-}1}, u_{1:t}) = P(x_t \mid x_{t\text{-}1}, u_t)$ **Action Model**

How do we get *sensor model* and *action model* in a <u>more general</u> way?

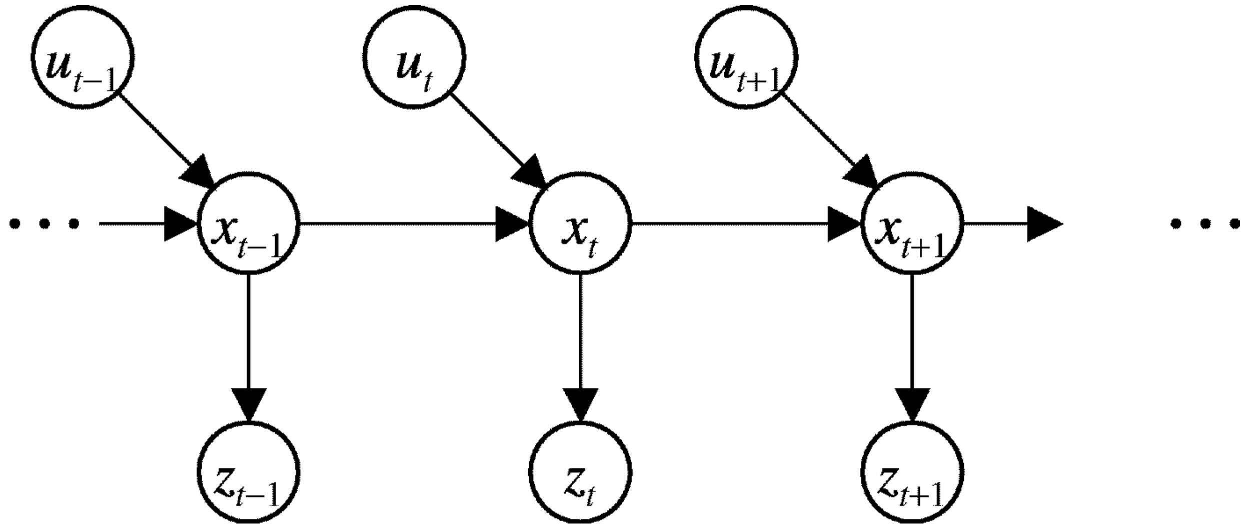
Recap – Bayes Filter



How do we get *sensor model* and *action model* in a more general way?

Kalman Filter

A special case of Bayes Filter with <u>linear Gaussian dynamics and sensory models</u>



$$P(z_t \mid x_{1:t}, z_{1:t-1}, u_{1:t}) = P(z_t \mid x_t)$$ Sensor Model

$$P(x_t \mid x_{1:t-1}, z_{1:t-1}, u_{1:t}) = P(x_t \mid x_{t-1}, u_t)$$ Action Model

Kalman Filter

Prediction-> $$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$ Action Model (Dynamics)

Correction-> $$z_t = C_t x_t + \delta_t$$ Sensor Model (Observation)

$$Bel(x_0) = N(x_0; \mu_0, \Sigma_0)$$ State Initialization

Components of a Kalman Filter

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

$$z_t = C_t x_t + \delta_t$$

$$Bel(x_0) = N(x_0; \mu_0, \Sigma_0)$$

$\boxed{A_t}$ Matrix (n x n) that describes how the state evolves from t-1 to t without actions or noise.

$\boxed{B_t}$ Matrix (n x l) that describes how the action u_t changes the state from t-1 to t.

$\boxed{C_t}$ Matrix (k x n) that describes how to map the state x_t to an observation z_t.

$\boxed{\varepsilon_t}$ $\boxed{\delta_t}$ Random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance R_t and Q_t respectively.

Multivariate Gaussians Properties

$$\left.\begin{array}{l} X \sim N(\mu, \Sigma) \\ Y = AX + B \end{array}\right\} \quad \Rightarrow \quad Y \sim N(A\mu + B, A\Sigma A^T)$$

$$\left.\begin{array}{l} X_1 \sim N(\mu_1, \Sigma_1) \\ X_2 \sim N(\mu_2, \Sigma_2) \end{array}\right\} \Rightarrow p(X_1) \cdot p(X_2) \sim N\left(\frac{\Sigma_2}{\Sigma_1 + \Sigma_2} \mu_1 + \frac{\Sigma_1}{\Sigma_1 + \Sigma_2} \mu_2, \quad \frac{1}{\Sigma_1^{-1} + \Sigma_2^{-1}} \right)$$

- We stay in the "Gaussian world" as long as we start with Gaussians and perform only linear transformations.

Linear Gaussian Systems: Initialization

- Initial belief is normally distributed:

$$Bel(x_0) = N\left(x_0; \mu_0, \Sigma_0\right)$$

Linear Gaussian Systems: Dynamics

- Dynamics are linear function of state and action plus additive noise:

$$\boxed{x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t}$$

$$\boxed{p(x_t \mid u_t, x_{t-1}) = N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right)}$$

$$\boxed{\begin{array}{ccc} \overline{Bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) & & Bel(x_{t-1})\, dx_{t-1} \\ \Downarrow & & \Downarrow \\ \sim N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right) & & \sim N\left(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}\right) \end{array}}$$

Linear Gaussian Systems: Dynamics

$$\overline{Bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \qquad Bel(x_{t-1})\ dx_{t-1}$$

$$\Downarrow \qquad\qquad\qquad \Downarrow$$

$$\sim N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right) \quad \sim N\left(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}\right)$$

$$\Downarrow$$

$$\overline{Bel}(x_t) = \eta \int \exp\left\{-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)\right\}$$

$$\exp\left\{-\frac{1}{2}(x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - \mu_{t-1})\right\} dx_{t-1}$$

$$\overline{Bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

Linear Gaussian Systems: Observations

- Observations are linear function of state plus additive noise:

$$\boxed{z_t = C_t x_t + \delta_t}$$

$$\boxed{p(z_t \mid x_t) = N\left(z_t; C_t x_t, Q_t\right)}$$

$$\boxed{\begin{array}{llll} Bel(x_t) = & \eta & p(z_t \mid x_t) & \overline{Bel}(x_t) \\ & & \Downarrow & \Downarrow \\ & & \sim N\left(z_t; C_t x_t, Q_t\right) & \sim N\left(x_t; \overline{\mu}_t, \overline{\Sigma}_t\right) \end{array}}$$

Linear Gaussian Systems: Observations

$$Bel(x_t) = \eta \quad p(z_t \mid x_t) \qquad \overline{Bel}(x_t)$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$\sim N(z_t; C_t x_t, Q_t) \qquad \sim N\left(x_t; \bar{\mu}_t, \bar{\Sigma}_t\right)$$

$$\Downarrow$$

$$Bel(x_t) = \eta \exp\left\{-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t)\right\} \exp\left\{-\frac{1}{2}(x_t - \bar{\mu}_t)^T \bar{\Sigma}_t^{-1}(x_t - \bar{\mu}_t)\right\}$$

$$Bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases} \quad \text{with} \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

Kalman Filter Algorithm

1. Algorithm **Kalman_Filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

2. Prediction:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. Correction:
6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

9. Return μ_t, Σ_t

Recap - Kalman Filter

A special case of Bayes Filter with linear Gaussian dynamics and sensory models

- Highly computationally efficient
- Optimal for linear Gaussian systems!
- But does this linearity/Gaussian distribution always hold in robotic systems?

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

Action Model

$$z_t = C_t x_t + \delta_t$$

Sensor Model

$$Bel(x_0) = N(x_0; \mu_0, \Sigma_0)$$

Extensions

- Extended Kalman Filter
- Unscented Kalman Filter
- Learning-based Kalman Filter

……