Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu
(slides adapted from Vivek Thangavelu)

Fast Robots

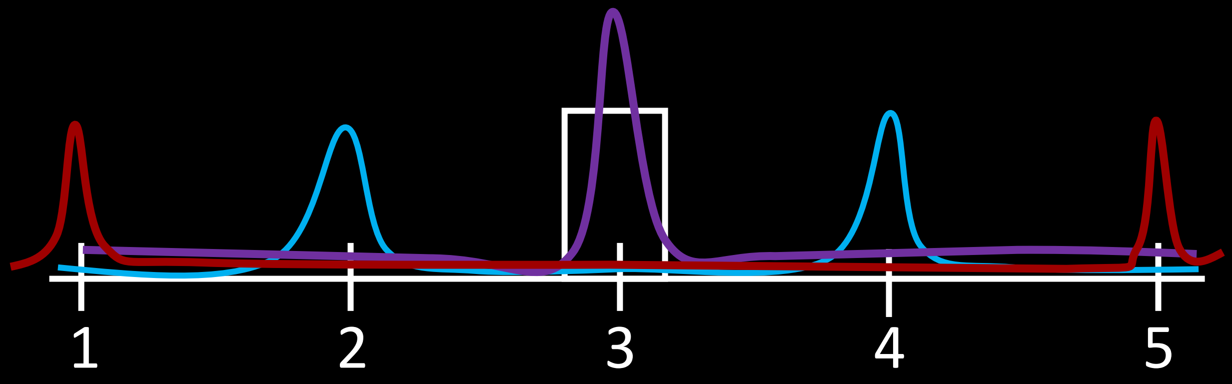
Markov Processes
&
Bayes Filter

Recap

- Random variable
  - $X \colon \Omega \rightarrow \mathbb{R}$
- The probability that the random variable X has value x:
  - $P(X = x)\ or\ p(x)$
- Probabilities sum to 1
  - $\sum_x P(X = x) = 1$
- Probabilities are always greater than 0
  - $P(X{=}x) \geq 0$
- Joint distribution Y
  - $p(x, y) = P(X = x\ and\ Y = y)$
- Conditional probability
  - $p(x|y) = \frac{p(x,y)}{p(y)}$
- Independence
  - $p(x, y) = p(x)p(y)$
  - $p(x|y) = p(x) = \dfrac{p(x, y)}{p(y)}$
- If X and Y are conditionally independent given Z=z, then
  - $p(x, y|z) = p(x|z)p(y|z)$
- Marginal probability
  - $p(x) = \sum_y p(x|y)p(y)$

Bayesian Inference

- Lost robot example
  - $p(X_0 = 1 \text{ or } 2 \text{ or } 3 \text{ or } 4 \text{ or } 5) = 1/5$
  - $p(x|y)$ can be hard to compute
  - What is $p(y|x)$?
  - If Y=1, where are you most likely to be?
  - If Y=0, where are you most likely to be?
  - If Y=2, where are you most likely to be?



likelihood *prior*

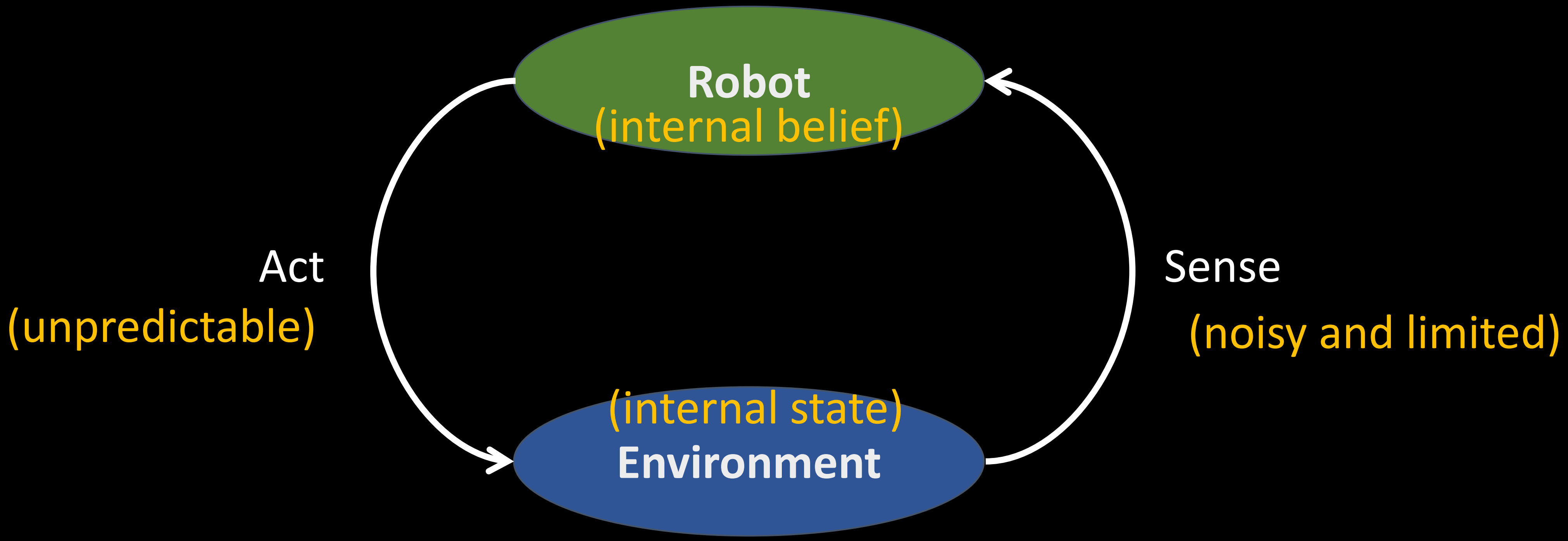
$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

posterior *marginal likelihood (constant)*

- y = Sensor data
- x = Robot state/ location

Robot-Environment Interaction

Robot-Environment Interaction



- Two fundamental types of interaction between a robot and its environment:
  - Sensor Measurements/Observations
  - Control Actions

Robot-Environment Model

- Helps us express a robot-environment interaction using probability
  - Typically modeled as a discrete time system
    - The state at time t will be denoted by as x_t
    - A sensor measurement at time t will be denoted as z_t
    - A control action will be denoted by u_t
      - Induces a transition from state x_{t-1} to x_t

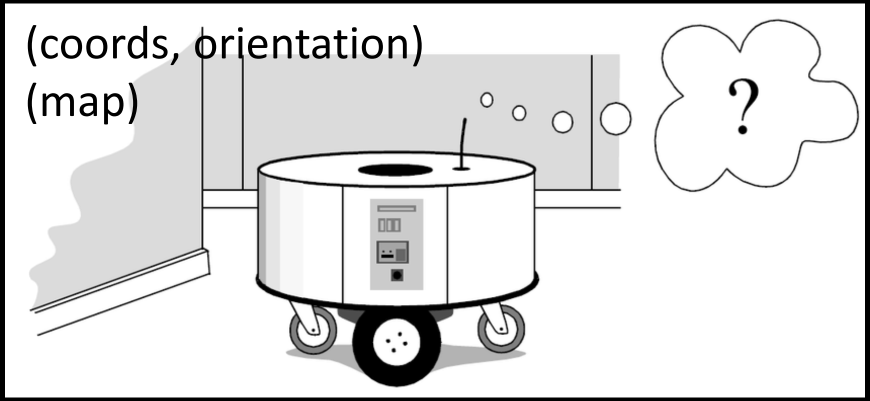
Conventions as per Siegwart, R., Nourbakhsh, I.R. and Scaramuzza, D., 2011. Introduction to autonomous mobile robots. MIT press.

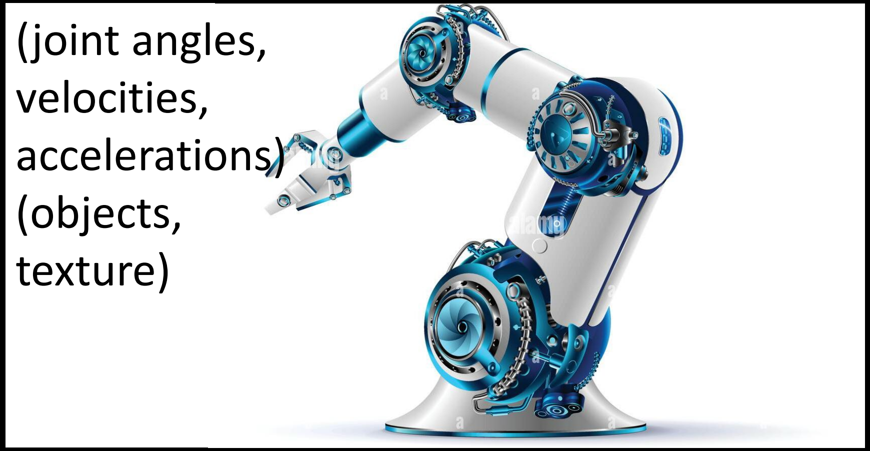
Robot-Environment Model

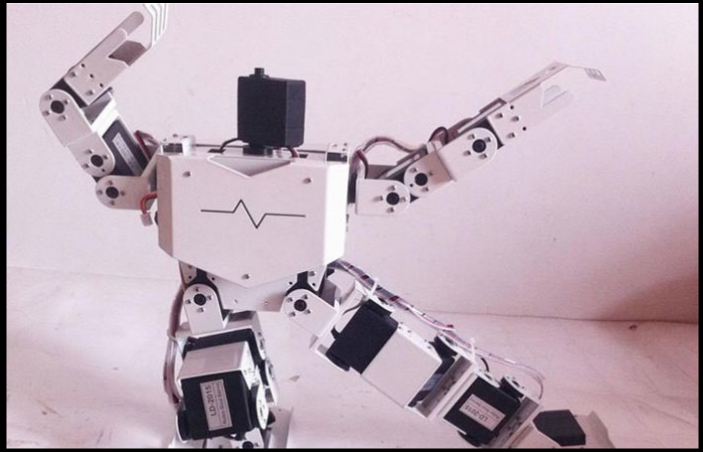
- (Arbitrary) Assumptions
  - The robot executes a control action u_t first and then takes a measurement z_t
  - There is one control action per time step t
  - Control actions include a legal action "*do-nothing*"
  - There is only one measurement z per time step t
  - Shorthand Notation: $x_{t1:t2} = x_{t1}, x_{t1+1}, x_{t1+2}, \ldots, x_{t2}$

Robot State

- The state, x, includes:
  - Robot Specific:
    - Pose, Velocity, Sensor status, etc.
  - Environment Specific:
    - Static variables
      - location of walls
    - Dynamic variables
      - Whereabouts of people in the vicinity of the robot
  - …context-specific







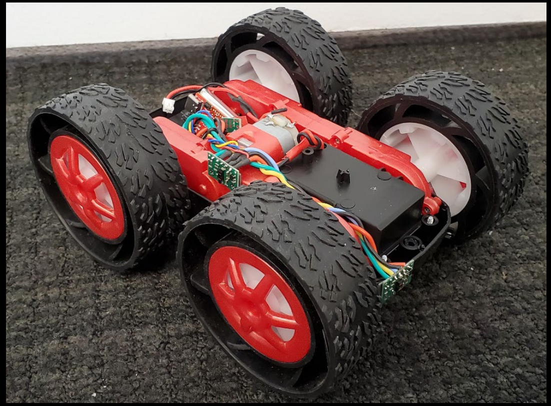
Sensor Measurements/Observations

- z_t
  - Tend to increase the robot's knowledge

balance

Control Actions

- u_t
  - …change the state of the world
  - carry information about the change of the robot state in the time interval (t-1:t]
  - Tends to induce loss of knowledge



Probabilistic Generative Laws

- The evolution of state and measurements is governed by probabilistic laws
  - State: How is x_t generated stochastically?
  - Measurements: How is z_t generated stochastically?

State Generation

- x_t depends on $x_{0:t-1}$, $z_{1:t-1}$ and $u_{1:t}$

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t})$$

...intractable!

Markov Assumption

Markov Assumption

The Markov assumption postulates that past and future data are independent if one knows the current state

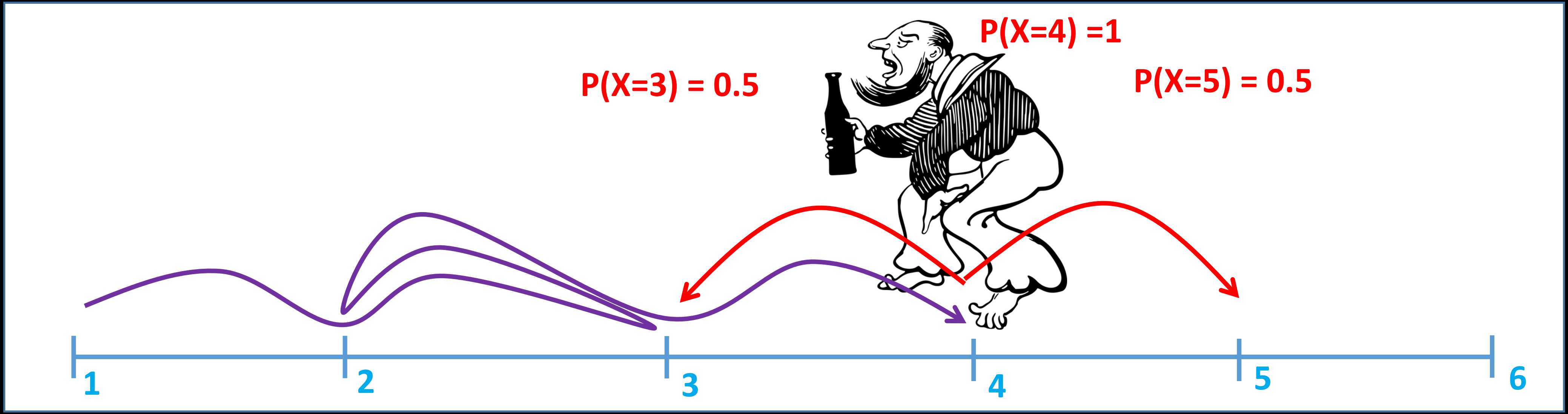
- A stochastic model/process that obeys the Markov assumption is a Markov model
- (This does not mean that x_t is deterministic based on x_{t-1})
- Iff we can model our robot as a Markov process…
  - We can recursively estimate x_t using
    - x_{t-1}, z_t, u_t
    - *But not* $x_{0:t-1}$, $z_{1:t-1}$, $u_{1:t}$*!*
    - Tractable!

???

Andrey Markov (1856–1922) was a Russian mathematician best known for his work on stochastic processes

Drunkard's walk!

- Random walk on the number line
  - At each step, the position may change by +1 or −1 with equal probability
- The transition probabilities depend only on the current position, not on the manner in which the position was reached
- This is a Markov Process!



Coin Purse

- Contents
  - 5 quarters (25¢)
  - 5 dimes (10¢)
  - 5 nickels (5¢)
- Draw coins randomly, one at a time and place them on a table
- Example:
  - X_n = total value of coins on the table after n draws
  - The sequence $\{ X_n : n \epsilon \mathbb{N} \}$ is a stochastic process

- First, I draw a nickel
- What is $X_1 =$ 5¢
- Next, I draw a dime
- What is $X_2 =$ 15¢

$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow$

Coin Purse

- Contents
  - 5 quarters (25¢)
  - 5 dimes (10¢)
  - 5 nickels (5¢)
- Draw coins randomly, one at a time and place them on a table
- Example:
  - X_n = total value of coins on the table after n draws
  - The sequence $\{ X_n : n \in \mathbb{N} \}$ is a stochastic process

- Suppose…
  - In the first six draws, you pick all 5 nickels and 1 quarter
  - $X_6 = 50¢$
- What can we say about X_7?
  - $P(X_7 \geq 0.55) = 1$
- Can you do better?
  - Can you draw a nickel in the 7th draw?
  - $P(X_7 \geq 0.6) = 1$
- Exercise
  - Is this a Markov Model?
  - If not, can you tweak the definition of X_n to make it one?

Coin Purse

- Contents
  - 5 quarters (25¢)
  - 5 dimes (10¢)
  - 5 nickels (5¢)
- Draw coins randomly, one at a time and place them on a table
- Example:
  - X_n = total value of coins on the table after n draws
  - The sequence $\{ X_n : n \epsilon \mathbb{N} \}$ is a stochastic process

- Markov model
  - X_n = {number of quarters, number of dimes, number of nickels} drawn
  - First you pick a nickel
    - $X_1 = \{0,0,1\}$
    - $X_6 = \{1,0,5\}$
  - Now, what can you say about X_7?
    - $p(X_7 \geq 0.6) = 1$

- State space: 6*6*6 = 216 possible states
- …but independent of the number of draws

Robot-Environment Model

State Generative Model

- x_t is generated stochastically from the state x_{t-1}
- x_t depends on $x_{0:t-1}$, $z_{1:t-1}$ and $u_{1:t}$

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t-1})$$

- If state x_t is modeled under the **Markov Assumption**, then

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

(conditional independence)

- Knowledge of only the previous state x_{t-1} and control u_t is sufficient to predict x_t

Tractable!

Measurement Generative Model

- Similarly, the process by which measurements are generated are of importance

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

- If x_t conforms to the **Markov Assumption**, then

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

(conditional independence)

- The state x_t is sufficient to predict the (potentially noisy) measurements
- Knowledge of any other variable, such as past measurements, controls, or even past states, is irrelevant under the Markov Assumption

Robot-Environment Model

+

Markov Assumption

+

Bayes Theorem

=

Bayes Filter

Robot Belief

- Probabilistic robotics represents beliefs through *posterior conditional probability distributions*
  - probability distributions over state variables conditioned on available data
  - The belief of a robot is the posterior distribution over the state of the environment, given all past sensor measurements and all past controls
    - Belief over a state variable x_t is denoted by $bel(x_t)$:

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

  - The (prior) belief is the belief before incorporating the latest measurement z_t

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

Bayes Filter

- It is a recursive algorithm that calculates the belief distribution from measurements and control data

1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t, z_t)$:
2. for all x_t do
3. $\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t|u_t, x_{t-1})\ bel(x_{t-1})$
4. $bel(x_t) = \eta\, p(z_t|x_t)\ \overline{bel}(x_t)$
5. endfor
6. return $bel(x_t)$

Bayes Filter

1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t, z_t)$:
2. for all x_t do
3. $\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t | u_t, x_{t-1})\ bel(x_{t-1})$ (Prediction step)
4. $bel(x_t) = \eta\, p(z_t | x_t)\ \overline{bel}(x_t)$
5. endfor
6. return $bel(x_t)$

Transition probability /action model

Bayes Filter

1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t, z_t)$:
2. for all x_t do
3. $\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t|u_t, x_{t-1}) \; bel(x_{t-1})$ (Prediction step)
4. $bel(x_t) = \eta \, p(z_t|x_t) \; \overline{bel}(x_t)$ (Update/measurement Step)
5. endfor
6. return $bel(x_t)$

Transition probability /action model

Measurement Probability / Sensor Model

Kalman Filter Implementation

State estimate: $\mu(t)$
State uncertainty: $\Sigma(t)$
Process noise: Σ_u
Kalman filter gain: K_{KF}
Measurement noise: Σ_z

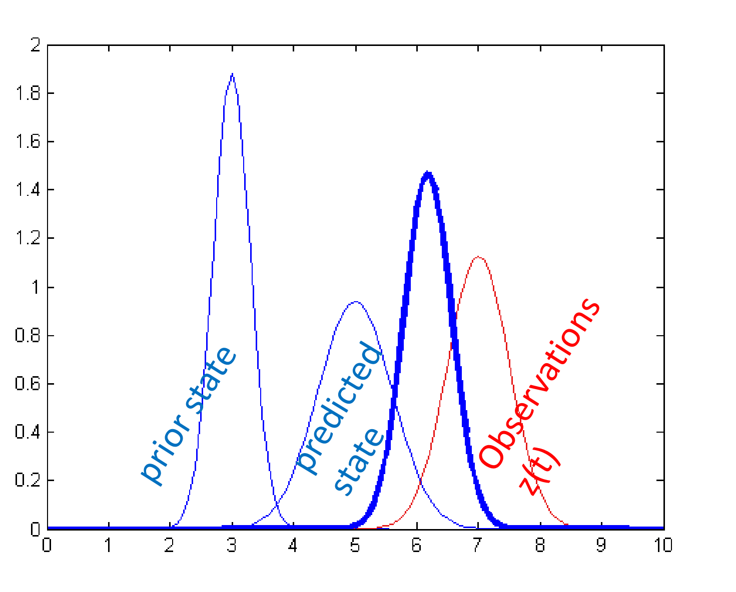
Kalman Filter ($\mu(t-1)$, $\Sigma(t-1)$, u(t), z(t))

1. $\mu_p(t) = A\,\mu(t-1) + B\,u(t)$
2. $\Sigma_p(t) = A\,\Sigma(t-1)\,A^T + \Sigma_u$
3. $K_{KF} = \Sigma_p(t)\,C^T\,(\,C\,\Sigma_p(t)\,C^T + \Sigma_z)^{-1}$
4. $\mu(t) = \mu_p(t) + K_{KF}\,(\,z(t) - C\,\mu_p(t)\,)$
5. $\Sigma(t) = (\,\mathbf{I} - K_{KF}\,C)\,\Sigma_p(t)$
6. Return $\mu(t)$ and $\Sigma(t)$

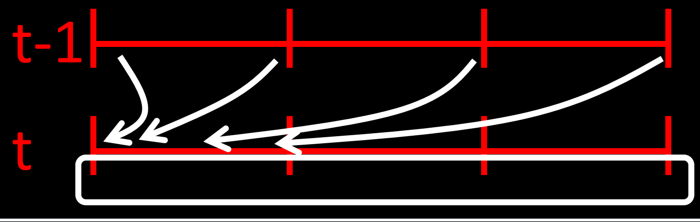
prediction

update

$$\Sigma_u = \begin{bmatrix} {\sigma_1}^2 & 0 & 0 \\ 0 & {\sigma_2}^2 & 0 \\ 0 & 0 & {\sigma_3}^2 \end{bmatrix}, \Sigma_z = \begin{bmatrix} {\sigma_4}^2 & 0 \\ 0 & {\sigma_5}^2 \end{bmatrix}$$



Bayes Filter



1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t, z_t)$:
2. for all x_t do
3. $\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t | u_t, x_{t-1})\ bel(x_{t-1})$ (Prediction step)
4. $bel(x_t) = \eta\, p(z_t | x_t)\ \overline{bel}(x_t)$ (Update/measurement Step)
5. endfor
6. return $bel(x_t)$

Dynamical Stochastic Model

- $p(x_t|x_{t-1}, u_t)$
  - It is known as the state transition probability
  - It specifies how the robot state evolves over time as a function of robot controls u_t
- $p(z_t|x_t)$
  - It is known as the measurement probability
  - It specifies how the measurements are generated from the robot state x_t
  - Informally, you may think of measurements as noisy projections of the state
- Remember that these predictions are stochastic and not deterministic

Bayes Filter - Initial Conditions

- To compute the posterior belief recursively, the algorithm requires an initial belief $bel(x_0)$ at time $t = 0$

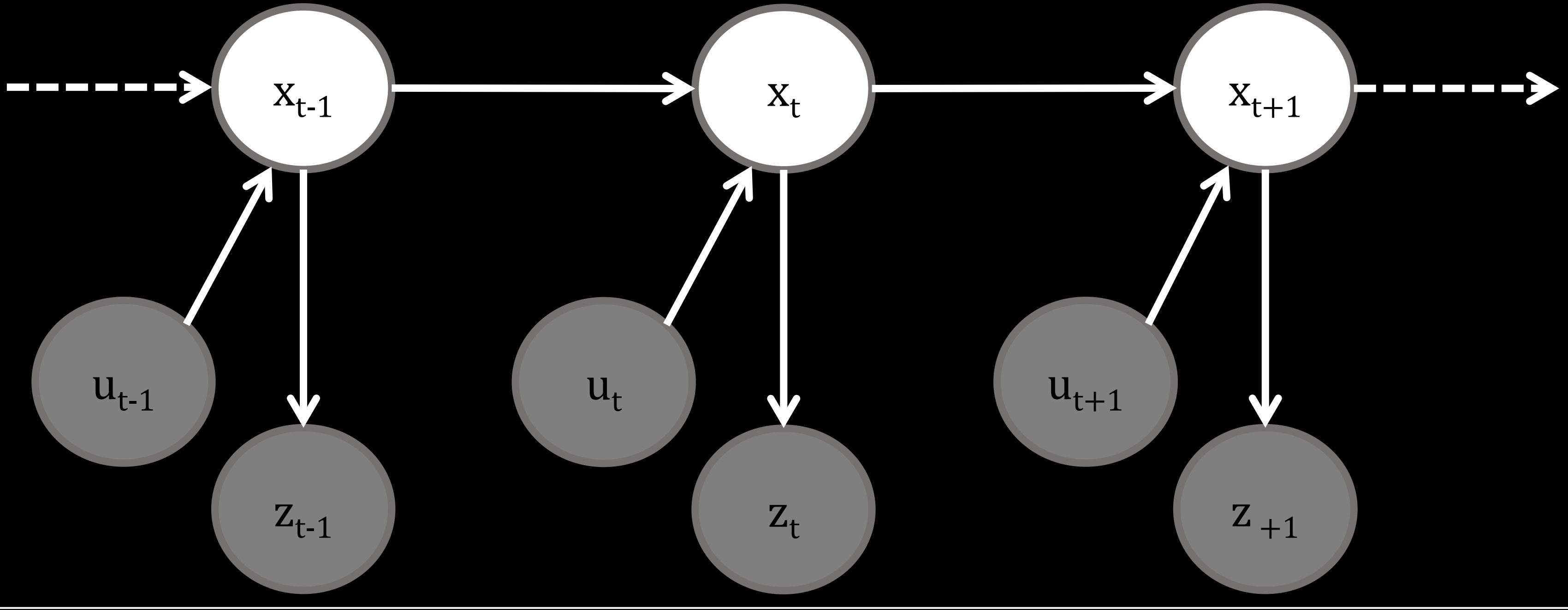
Bayes Filter

1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t, z_t)$:
2. for all x_t do
3. $\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t | u_t, x_{t-1})\ bel(x_{t-1})$ (Prediction step)
4. $bel(x_t) = \eta\, p(z_t | x_t)\ \overline{bel}(x_t)$ (Update/measurement Step)
5. endfor
6. return $bel(x_t)$

Bayes Filter - Initial Conditions

- To compute the posterior belief recursively, the algorithm requires an initial belief $bel(x_0)$ at time $t = 0$
- If we know the initial state with absolute certainty, we can initialize a point mass distribution that centers all probability mass on the correct value of x_0 and assign zero everywhere else
- If we are entirely ignorant of the initial state, we can initialize it with a uniform probability distribution over all the possible states

Dynamical Stochastic Model



Dynamic Bayes Network that characterizes the evolution of controls, states, and measurements.

- $p(x_t \mid x_{t-1}, u_t)$ and $p(z_t \mid x_t)$ together describe the **dynamical stochastic system** of the robot and its environment
- Such a generative model is also known as a Hidden Markov Model (HMM) or Dynamic Bayes Network (DBN)

References

1. Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. Probabilistic robotics. MIT press, 2005.
2. http://gki.informatik.uni-freiburg.de/teaching/ws0607/advanced/recordings/BayesFiltering.pdf
3. http://home.deib.polimi.it/restelli/MyWebSite/pdf/E05_h.pdf
4. https://en.wikipedia.org/wiki/Markov_chain
5. https://en.wikipedia.org/wiki/Markov_model
6. Image Sources:
   a. https://blog.robotiq.com/a-brief-history-of-robots-in-manufacturing
   b. https://northwesturbanist.files.wordpress.com/2015/01/20151104_082523.jpg
   c. https://expresswriters.com/is-the-dress-blue-black-white-or-gold-how-it-went-viral/
   d. https://en.wikipedia.org/wiki/Andrey_Markov