CSE 574 Lecture 18: POMDP

(Slides adapted from Kaebling et. al., Geoff Hollinger CMU lecture, and POMDP tutorial)

Professor: Stephanie Gil

Email: sgil@asu.edu (Office hours M 12-1pm BYENG 386)

TAs: Sushmita Bhattacharya sbhatt55@asu.edu (Office hours M 5-6 BYENG 392)

Weiying Wang wwang239@asu.edu (Office hours Th 2:30-3:30 BYENG 392)

Last Time

- HMM
- Inference with hidden state
- Viterbi algorithm and the most likely sequence

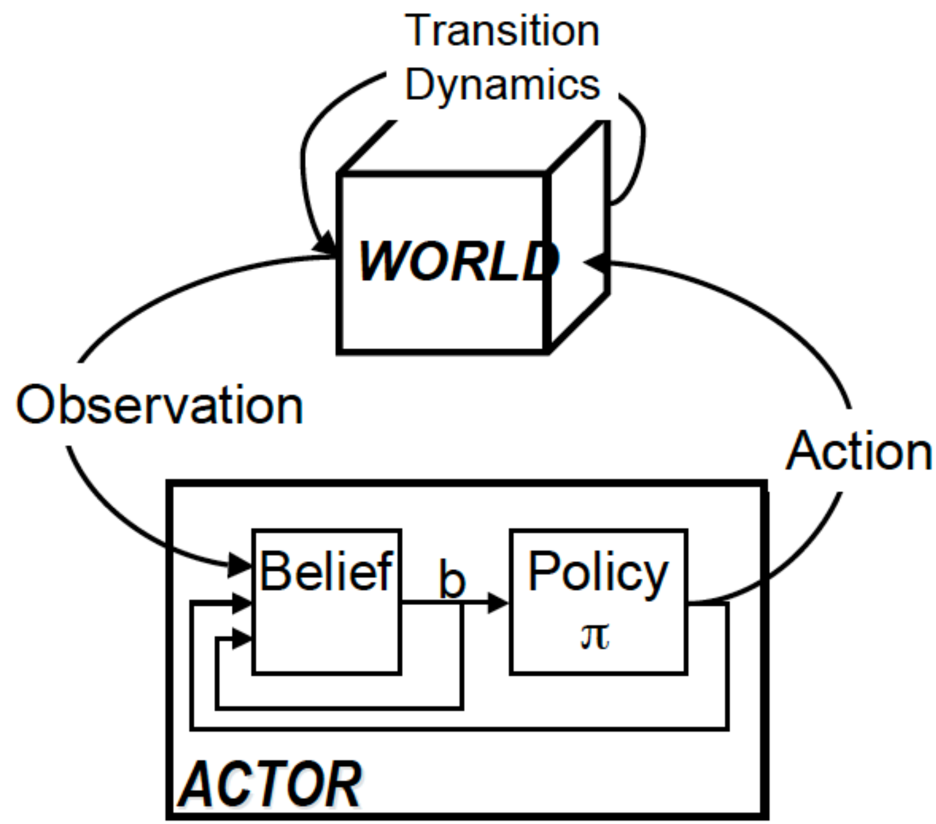
Today

- Partially Observable Markov Decision Processes
  - Updating belief state using observations and actions
  - Acting under uncertainty

- References
  - A POMDP tutorial: https://www.techfak.uni-bielefeld.de/~skopp/Lehre/STdKI_SS10/POMDP_tutorial.pdf
  - “Planning and Acting in Partially Observable Stochastic Domains,” Kaebling et. al.

Agent Model

- Set of states
- Set of actions
- Transition and reward probabilities
- Observation function
- Belief state
- Policy



Goal of a POMDP

- As before, our goal is to maximize long-term total discounted reward

- Find a policy that maximizes the value (expected future reward) of each state *s*:

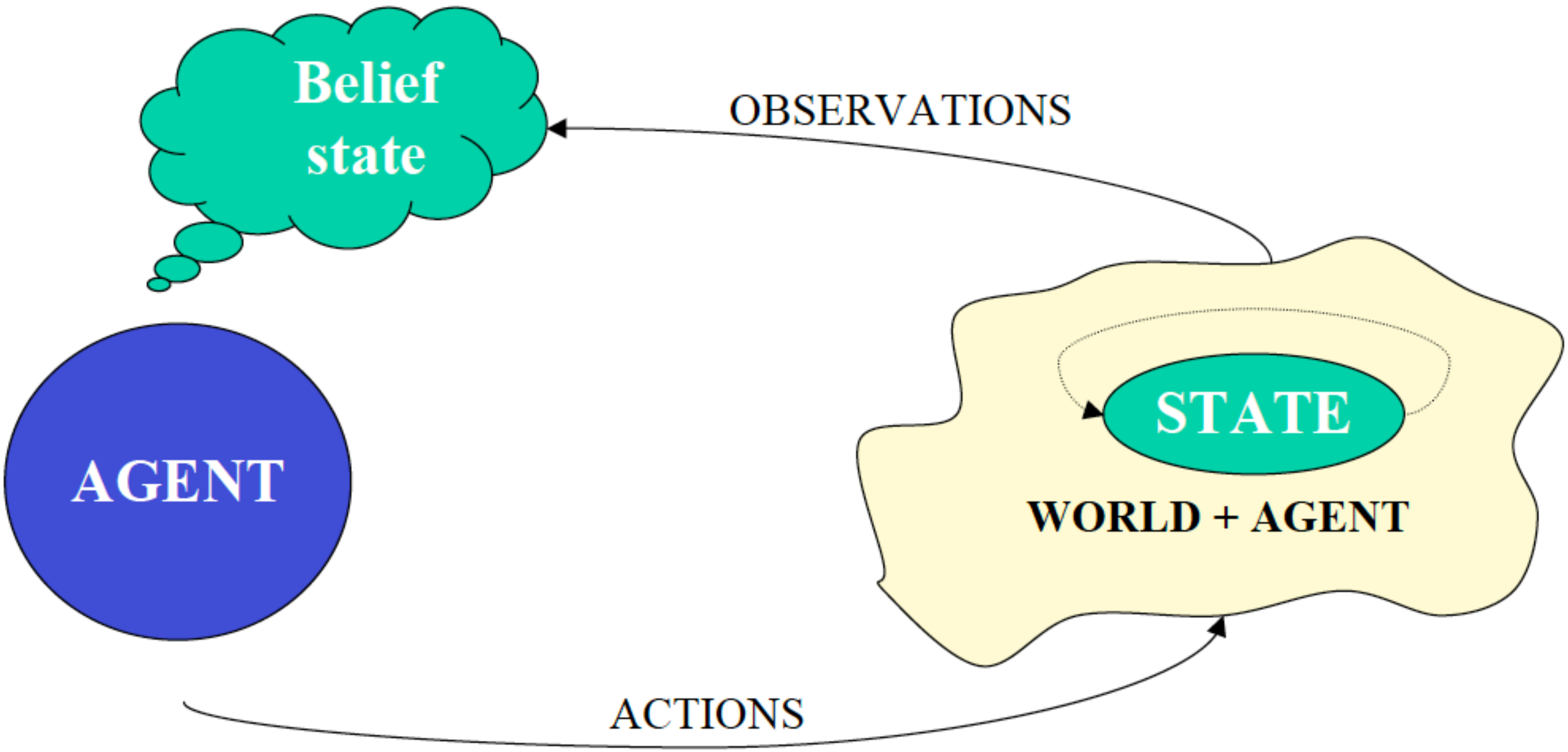
$$V^{\pi}(s) = E\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots \mid s_t = s, \pi\}$$

How Does this Fit in with Past Lectures?

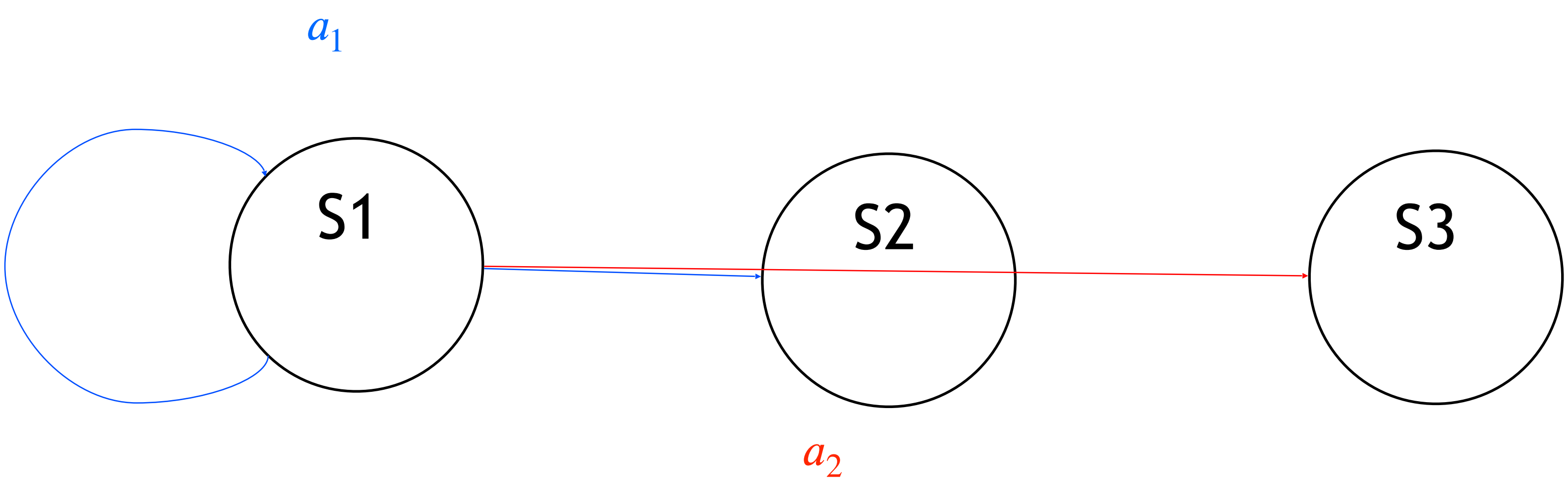
Markov Models		Do we have control over the state transitons?	
		NO	**YES**
Are the states completely observable?	**YES**	**Markov Chain**	**MDP** Markov Decision Process
	NO	**HMM** Hidden Markov Model	**POMDP** Partially Observable Markov Decision Process

Source: Geoff Hollinger POMDP tutorial https://www.cs.cmu.edu/~ggordon/780-fall07/lectures/POMDP_lecture.pdf

Overview of Current Problem



POMDP Model



<u>Components:</u>

Set of states: $s \in S$

Set of actions: $a \in A$

Set of observations: $o \in \Omega$

<u>POMDP parameters:</u>

Initial belief: $b_0(s)=\Pr(S=s)$

Belief state updating: $b'(s')=\Pr(s'|o, a, b)$

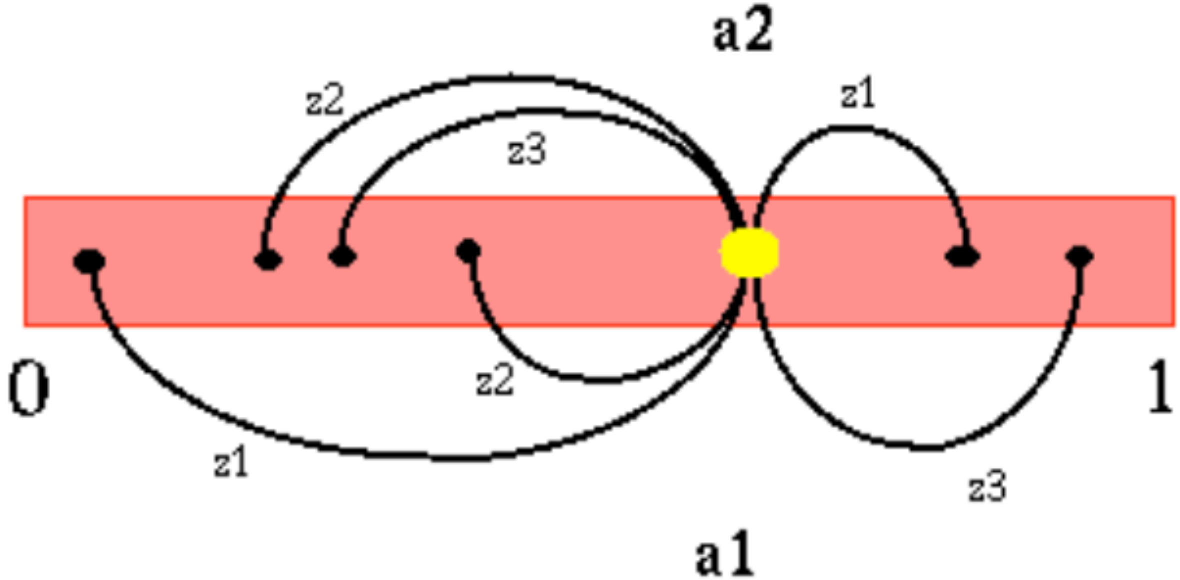
Observation probabilities: $O(s',a,o)=\Pr(o|s',a)$

$$\left.\begin{array}{l}\text{Transition probabilities: } T(s,a,s')=\Pr(s'|s,a) \\ \text{Rewards: } R(s,a)\end{array}\right\} \text{MDP}$$

Belief State

- The agent does not know what state it is in
- Definition ***belief state***: which states of the world are currently possible
  - Actions: belief state b={s1,s2}
  - Transitions:
- Reward function
- Transition function

Visualization of Belief State



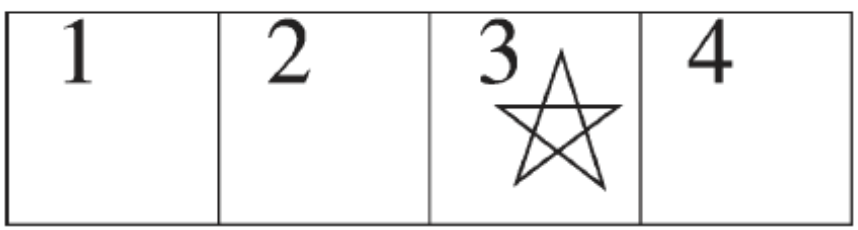
Z_i = *Observations*

$$b'(s_j) = \mathrm{P}(s_j \mid o,a,b) = \underbrace{\frac{P(o \mid s_j,a)\sum_{s_i \in S} P(s_j \mid s_i,a)\, b(s_i)}{\sum_{s_j \in S} P(o \mid s_j,a)\sum_{s_i \in S} P(s_j \mid s_i,a)\, b(s_i)}}_{\alpha}$$

Belief state update

Computing the Belief State

Simplified Gridworld



$$b'(s_j) = \mathrm{P}(s_j \mid o,a,b) = \frac{P(o \mid s_j,a)\sum_{s_i \in S} P(s_j \mid s_i,a)\,b(s_i)}{\sum_{s_j \in S} P(o \mid s_j,a)\sum_{s_i \in S} P(s_j \mid s_i,a)\,b(s_i)}$$

- Initial belief state

  $[0.333 \quad 0.333 \quad 0.000 \quad 0.333]$

- Action is successful with p=0.9 and opposite with p=0.1

- Agent has two observations, in the goal or not in the goal

- Stage 1: agent takes action EAST and does not observe goal

  $[0.100 \quad 0.450 \quad 0.000 \quad 0.450]$

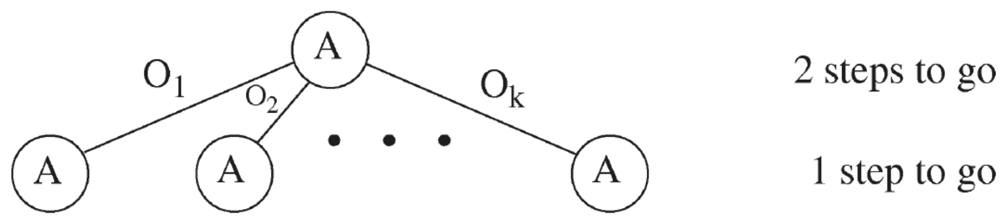
- Stage 2: agent takes action EAST again and does not observe goal

  $[0.100 \quad 0.164 \quad 0.000 \quad 0.736]$

How to Choose an Action?

- If we don’t know our state?
- Unlike having a distribution over states, we cannot take a “partial action”

Policy Tree



- Value of executing a one-step policy tree p

$$V_p(s) = R\big(s, a(p)\big)$$

- Value of executing a t-step policy tree p

$$V_p(s) = R\big(s, a(p)\big) + \gamma \cdot \big(\text{Expected value of the future}\big)$$
$$= R\big(s, a(p)\big) + \gamma \sum_{s' \in \mathcal{S}} \Pr\big(s' \mid s, a(p)\big) \sum_{o_i \in \Omega} \Pr\big(o_i \mid s', a(p)\big) V_{o_i(p)}\big(s'\big)$$

Expected Value of Executing Policy p

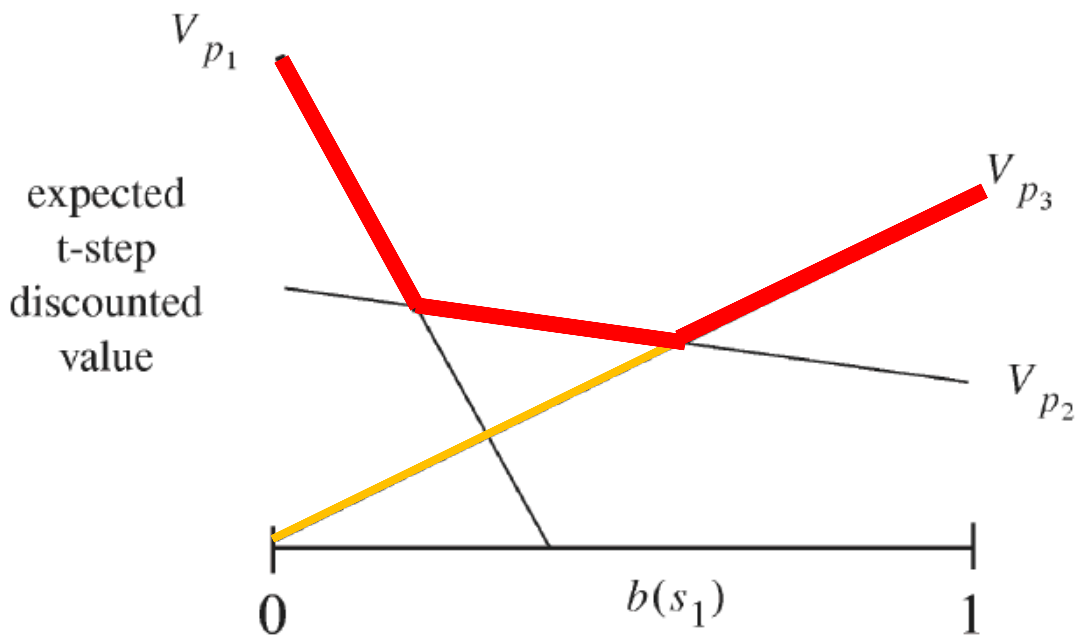
- Must compute the value over beliefs, not states. Build off of the equation on the last slide

$$V_p(b) = \sum_{s \in \mathcal{S}} b(s) \overbrace{V_p(s)}^{\text{Last slide}}$$

- Denote $\alpha_p = \langle V_p(s_1), \ldots, V_p(s_n) \rangle$ then $V_p(b) = b \cdot \alpha_p$
- And the optimal t-step value of starting in belief state b is the value of executing the best policy tree in that belief state

$$V_t(b) = \max_{p \in \mathcal{P}} b \cdot \alpha_p$$

Pictorial Representation of the Optimal t-step Value for Belief b

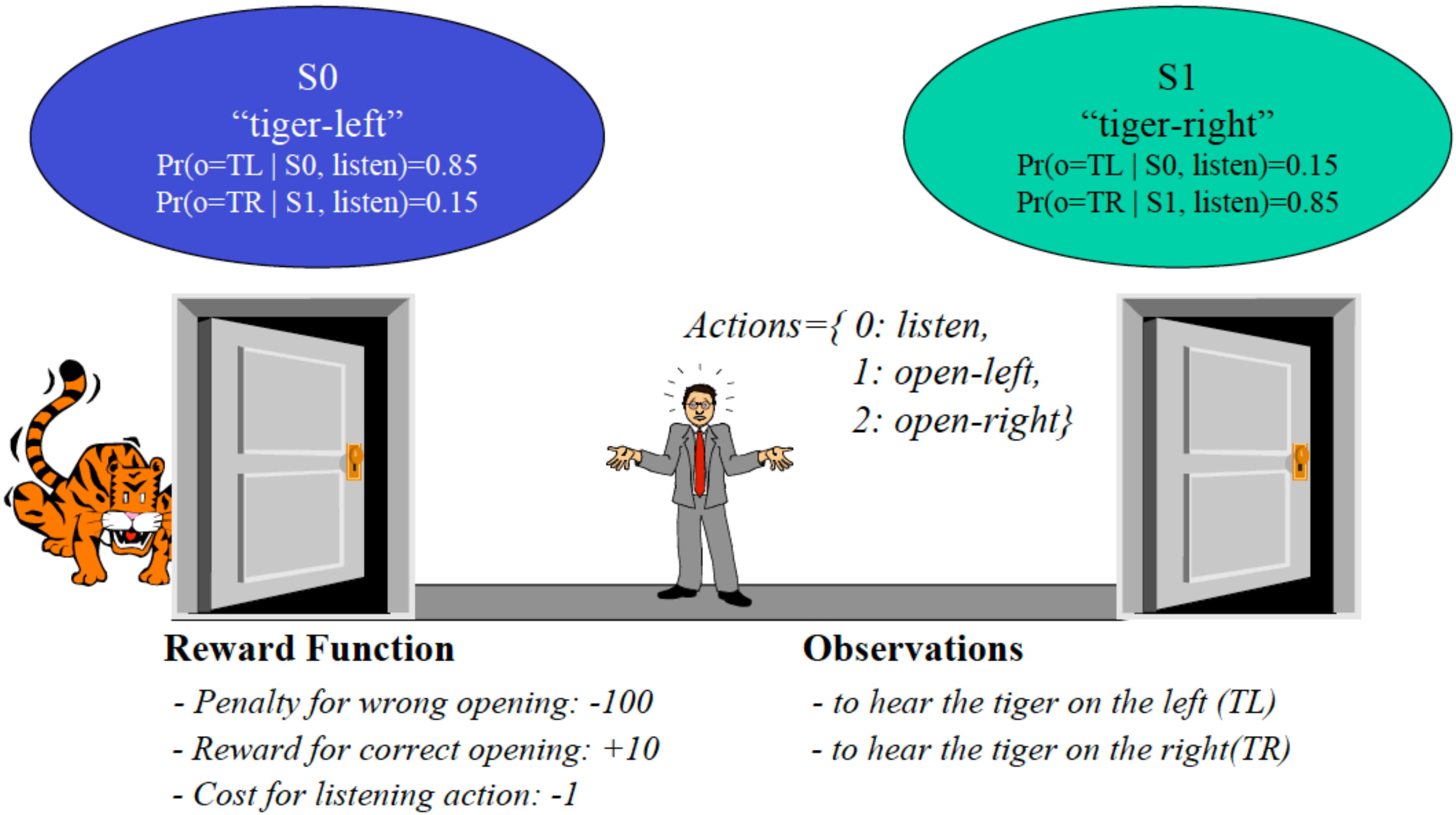


$$\alpha_p = \langle V_p(s_1), \ldots, V_p(s_n) \rangle$$

$$V_p(b) = b \cdot \alpha_p$$

$$V_t(b) = \max_{p \in \mathcal{P}} b \cdot \alpha_p$$

POMDP Example



Source: https://www.techfak.uni-bielefeld.de/~skopp/Lehre/STdKI_SS10/

Example: Tiger Problem (cont)

- Transition probabilities

Prob. (LISTEN)	Tiger: left	Tiger: right
Tiger: left	1.0	0.0
Tiger: right	0.0	1.0

Doesn't change Tiger location

Prob. (LEFT)	Tiger: left	Tiger: right
Tiger: left	0.5	0.5
Tiger: right	0.5	0.5

Problem reset

Prob. (RIGHT)	Tiger: left	Tiger: right
Tiger: left	0.5	0.5
Tiger: right	0.5	0.5

Source: https://www.techfak.uni-bielefeld.de/~skopp/Lehre/STdKI_SS10/

Example: Tiger Problem (cont)

- Observation probabilities

Prob. (LISTEN)	O: TL	O: TR
Tiger: left	0.85	0.15
Tiger: right	0.15	0.85

- Immediate rewards

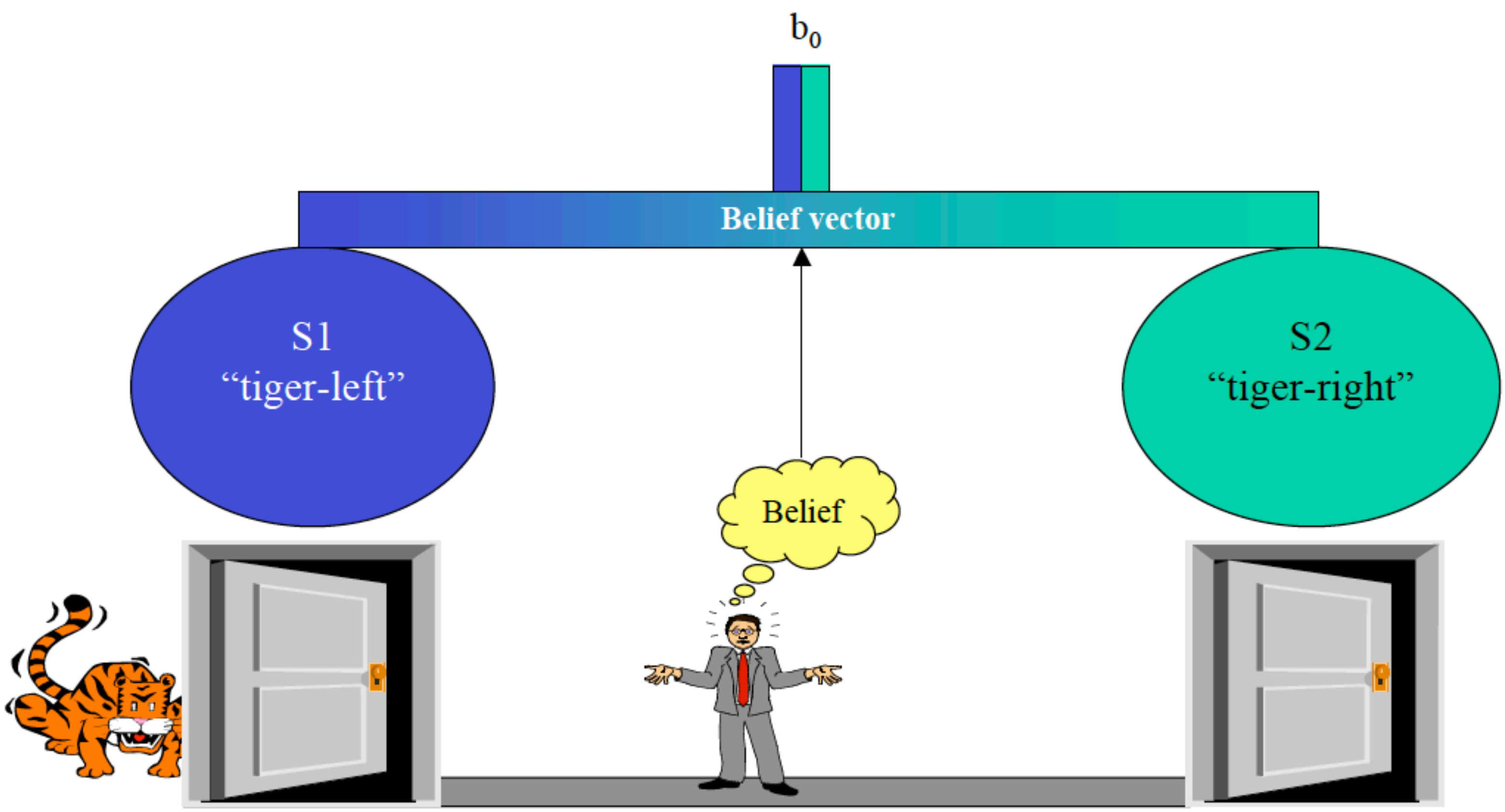
Reward (LISTEN)	
Tiger: left	-1
Tiger: right	-1

Reward (LEFT)	
Tiger: left	-100
Tiger: right	+10

Reward (RIGHT)	
Tiger: left	+10
Tiger: right	-100

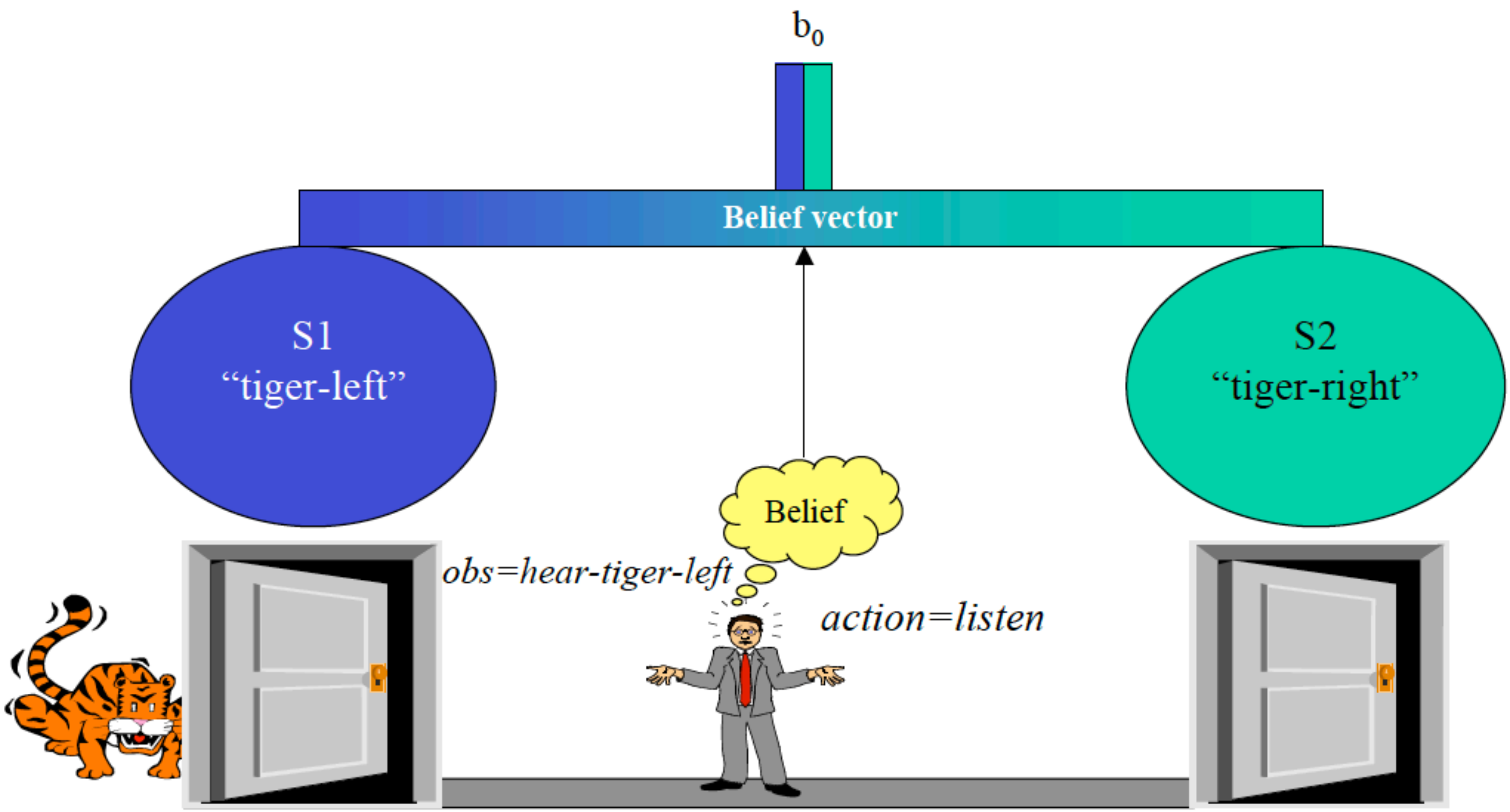
Source: https://www.techfak.uni-bielefeld.de/~skopp/Lehre/STdKI_SS10/

Example: Tiger Problem (cont)



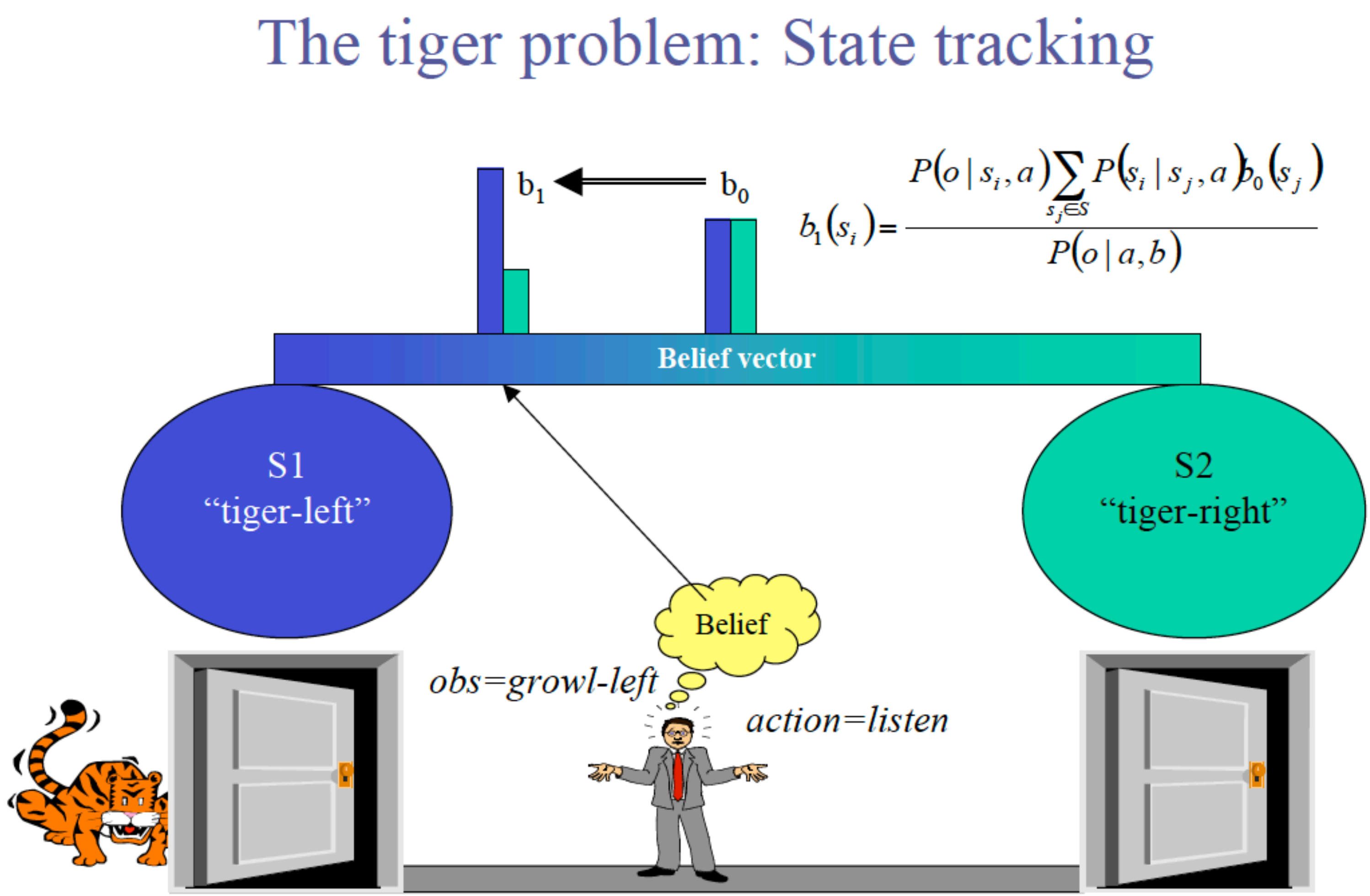
Source: https://www.techfak.uni-bielefeld.de/~skopp/Lehre/STdKI_SS10/

Example: Tiger Problem (cont)



Source: https://www.techfak.uni-bielefeld.de/~skopp/Lehre/STdKI_SS10/

Example: Tiger Problem (cont)

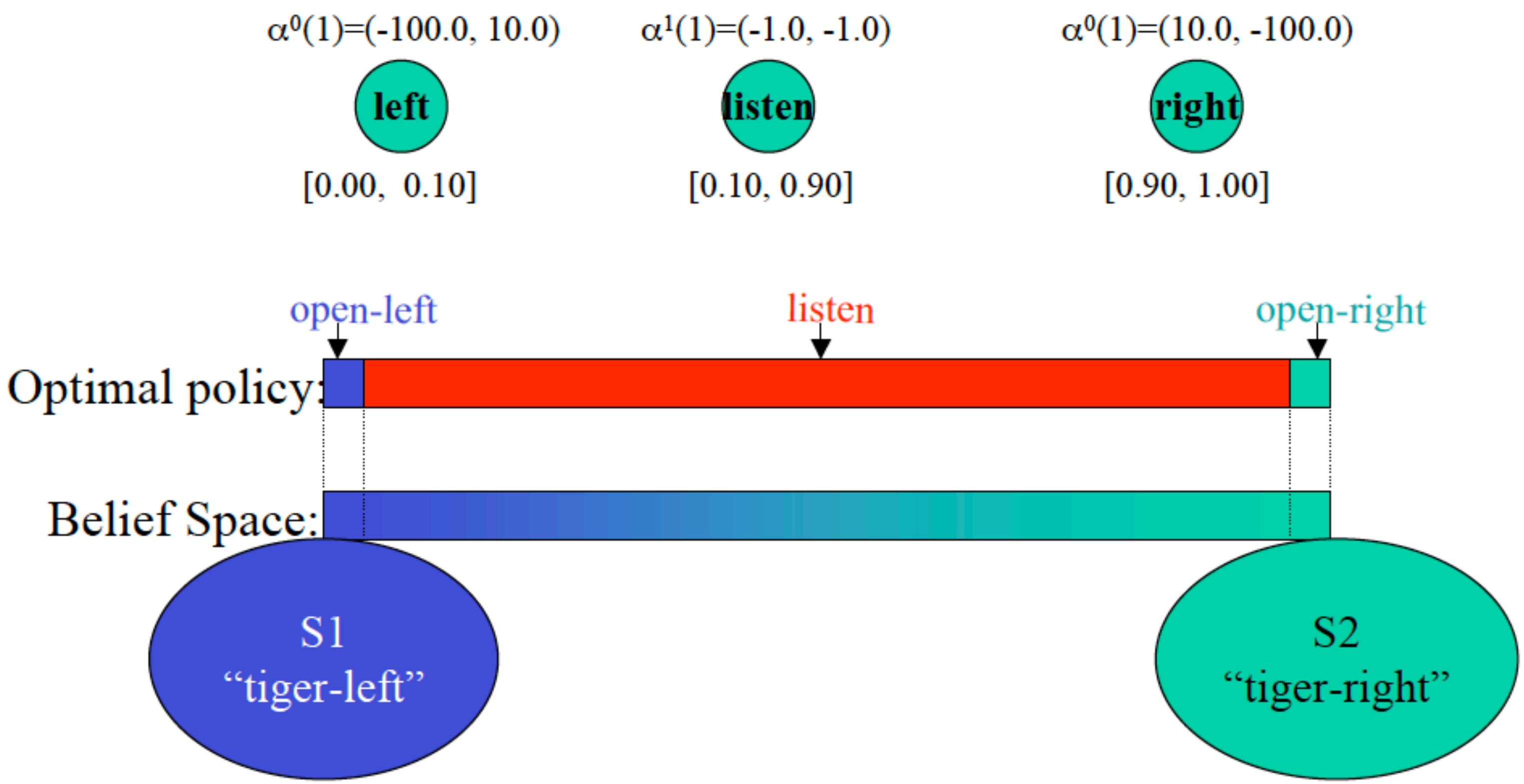


Source: https://www.techfak.uni-bielefeld.de/~skopp/Lehre/STdKI_SS10/

Example: Tiger Problem (cont)

Tiger Example Optimal Policy t=1

- Optimal Policy for t=1



Source: https://www.techfak.uni-bielefeld.de/~skopp/Lehre/STdKI_SS10/

Intro to SLAM

- Simultaneous Localization and Mapping (SLAM) is also a hidden state problem!
- Idea:
  - Given

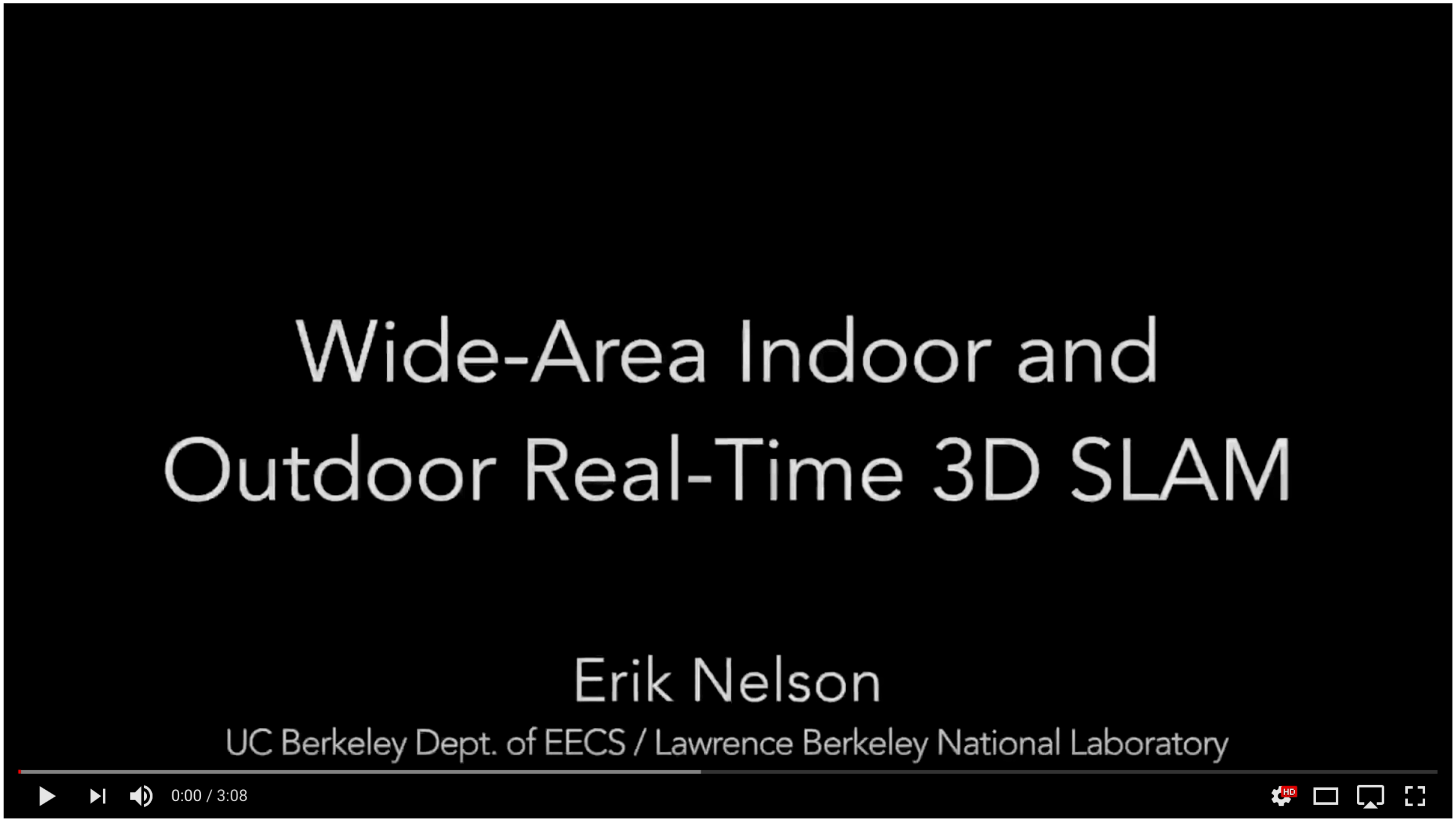
    $$z^i_{1:t} \equiv \{z^i_1, z^i_2, ..., z^i_t\}$$
    $$u^i_{1:t} \equiv \{u^i_1, u^i_2, ..., u^i_t\}$$

  - Simultaneous localize (find sequence $x^i_{1:t} \equiv \{x^i_1, x^i_2, ..., x^i_t\}$)

  and a map of the agents' environment

$$p(m, x_{1:t} | z_{1:t}, u_{1:t}, x_0)$$

Example Video of SLAM



Wide-Area Indoor and Outdoor Real-Time 3D SLAM

Applications

- What is SLAM important for?
  - Navigation – this can be robots, cars, drones, etc
  - Mapping and reconnaissance
  - Autonomous driving
    - Rideshare
    - Delivery

- What we will cover in class
  - Introduction to SLAM
  - High-level review of two major approaches (EKF and particle SLAM)
  - Multi-robot algorithms

Next Time

- Recitation on hidden state problems
- Introduction to SLAM
- Final project presentation schedule is online
- Final project presentations begin on the 19th!