Implementation of Partially Observed Markov Decision Process in Robotics

Project Presentation for Probability in computer science(CSCI 5434)

Gyanig Kumar, Saksham Khatwani, Uditanshu Tomar
University of Colorado at Boulder

What is POMDP

- A partially observable Markov decision process(POMDP) is an MDP with state uncertainty – meaning we cannot know the true state, only a belief about the true state using observations.
- POMDP is a problem formulation and not an algorithm.
- We use extend solvers like Monte-Carlo Tree Search(MCTS) Path planning algorithms with POMDP to implement Partially Observable Monte-Carlo Planning(POMCP)

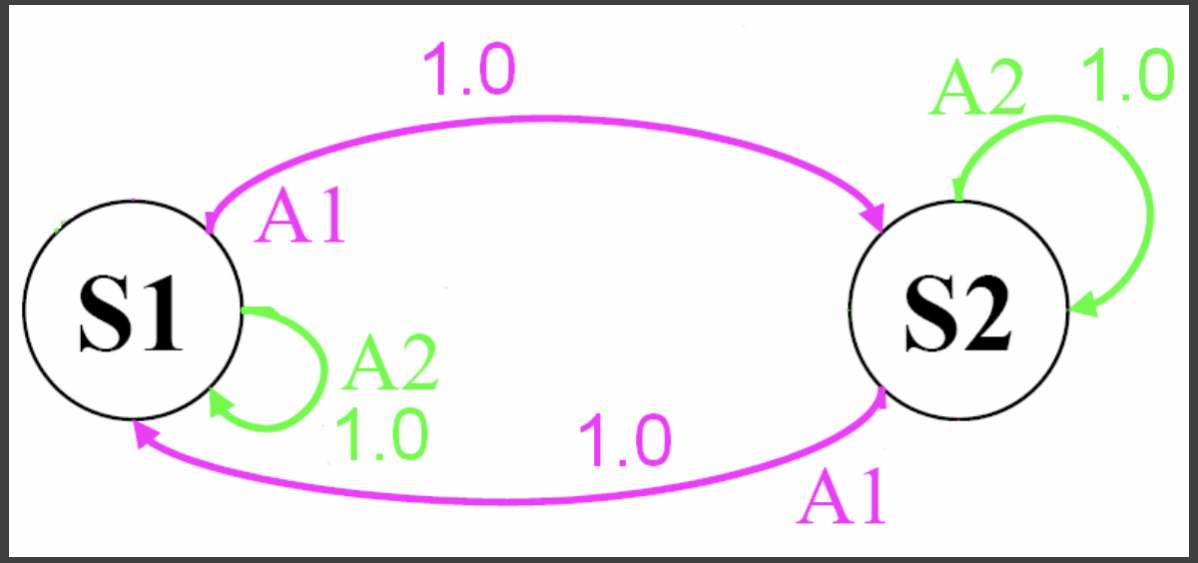
Reference : https://www.cs.cmu.edu/~ggordon/780-fall07/lectures/POMDP_lecture.pdf

What is POMDP

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

Reference : https://www.cs.cmu.edu/~ggordon/780-fall07/lectures/POMDP_lecture.pdf

How Do POMDPs Work ?



POMDP is defined by : States, Action, Transition Model, Observation Model, Rewards, Discount Factor

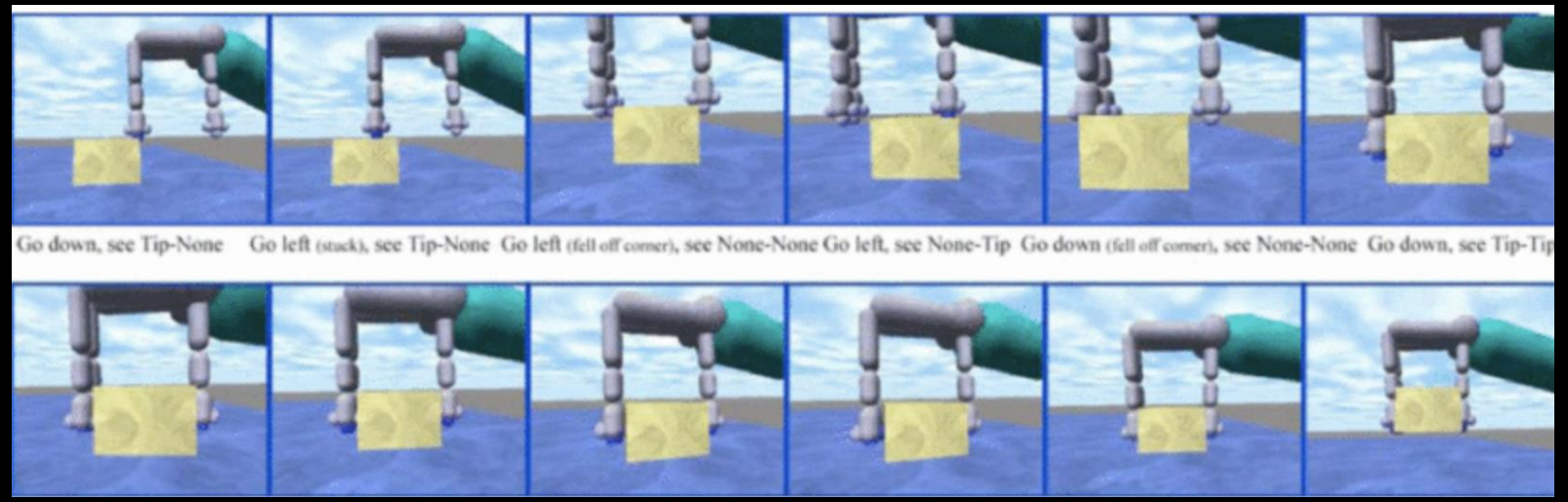
Reference : https://www.cs.cmu.edu/~ggordon/780-fall07/lectures/POMDP_lecture.pdf

Human Intent Understanding

- The task is to understand how does a human input (like keyboard control) affect the robot's behaviors.
- To be able use human input, we need a dynamic understanding of our state space, which provides us context on human input and control over robot movements
- We explore this task using Partially Observed Markov Decision Process

Methodologies

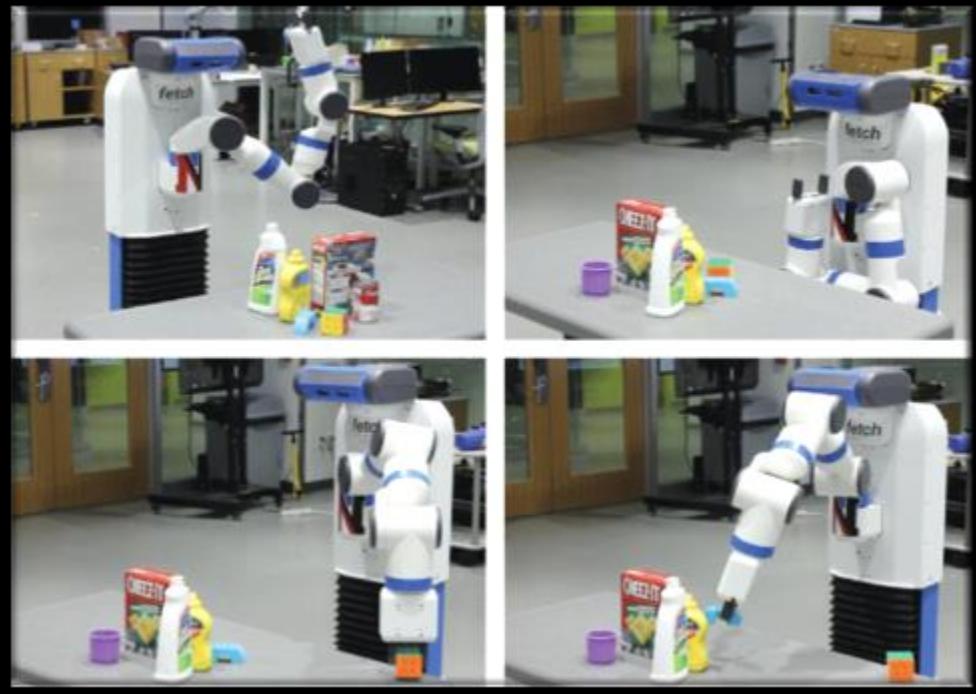
Point based value iteration (PBVI)



- K. Hsiao, L. P. Kaelbling and T. Lozano-Perez, "Grasping POMDPs"

Methodologies

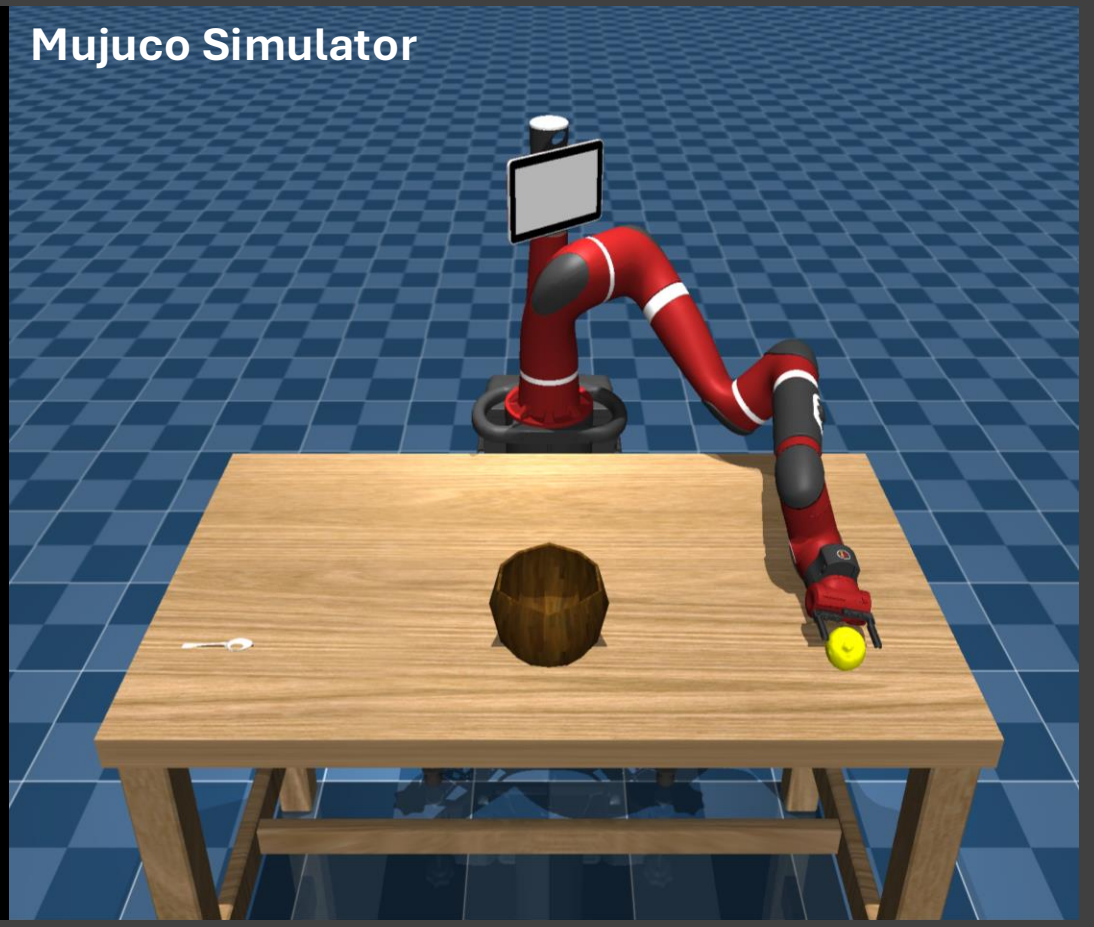
Monte Carlo Tree Search



- Y. Xiao, S. Katt, A. ten Pas, S. Chen, and C. Amato, “Online planning for target object search in clutter under partial observability,”

Our Experiment

- State : Position of robotic grasper in x, y, and z-axes.
- Observation : User input guiding the robot.
- Action : Movement of the robot in the 3 axes.
- Goals : Objects at the table with varying heights.
- Objective : Enable the robotic grasper to reach the top of a selected object.



Belief Update

Transition Probabilities: state (s to s') under action (a) at time (t)

Reward Function: Expected reward for action

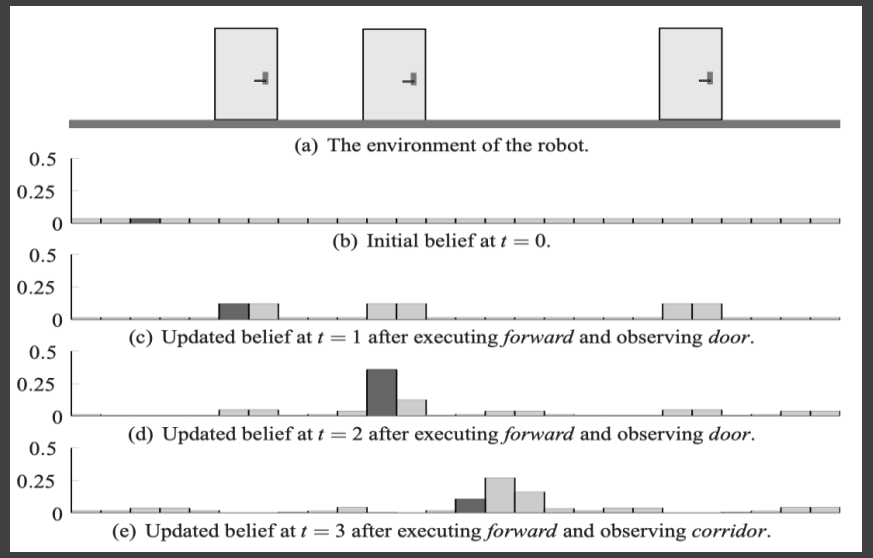
Initial State Distribution: Uniform dist. at t=0

History: Sequence of actions and observations

Policy: Maps history to action probabilities dist.

Belief: Probability distribution over states given history

$$P(s'|s,a) = \Pr(s_{t+1} = s' \mid s_t = s,\ a_t = a)$$
$$R(s,a) = \mathbb{E}[r_{t+1} \mid s_t = s,\ a_t = a]$$
$$P(s_0) = \Pr(s_{t=0} = s)$$
$$h_t = (a_1, o_1, \ldots, a_t, o_t)$$
$$\pi(a|h) = \Pr(a_{t+1} = a \mid h_t = h)$$
$$b(s|h) = \Pr(s_t = s \mid h_t = h)$$



Metrics

Belief Distribution

- The core state estimate POMDP lives on.
- A categorical probability vector **b = b1,b2,…,b‖S‖** where $\mathbf{b_k = P(s_k \mid history)}$

$$b_{t+1}(s') = \eta\, O(o_{t+1} \mid s', a_t) \sum_{s \in S} T(s' \mid s, a_t)\, b_t(s)$$

(Bayes-filter update: predict → correct)

- Encodes the agent's **current belief** about every hidden state

Metrics

Belief Entropy

- How much uncertainty is left in the robot's internal belief.
- *Shannon entropy* of the current belief distribution b_t
- Formula :

$$H(b_t) = -\sum_{s \in S} b_t(s) \log b_t(s)$$

- Measures the *spread* of probability mass.

Metrics

Cross-Entropy

- Is the belief putting probability mass on the **right** state?
- General form :

$$\mathrm{CE}(q,\, b_t) \;=\; -\sum_{s \in S} q(s)\, \log_2 b_t(s)$$

- Penalises beliefs that assign *low probability* to the true state
- Captures cases where entropy is low *but concentrated on the wrong state*.

Results

Hyper-parameters

Discount factor : 0.7

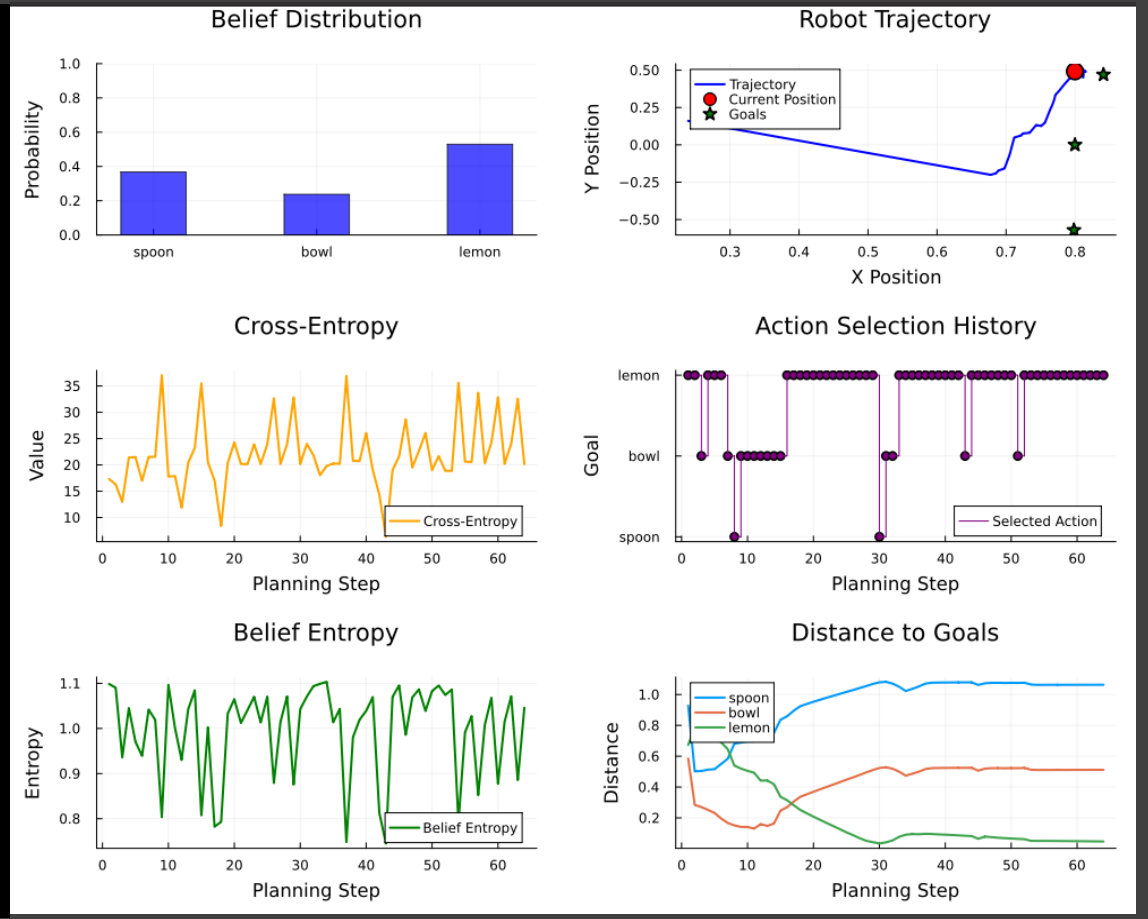
Max depth : 100

Goal threshold : 2cm

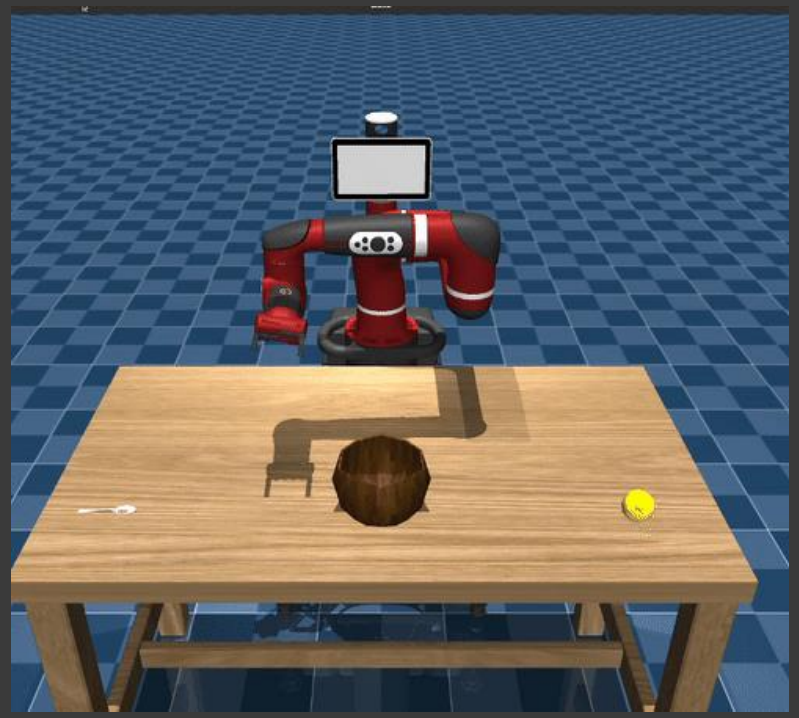
Default initial goal : Bowl

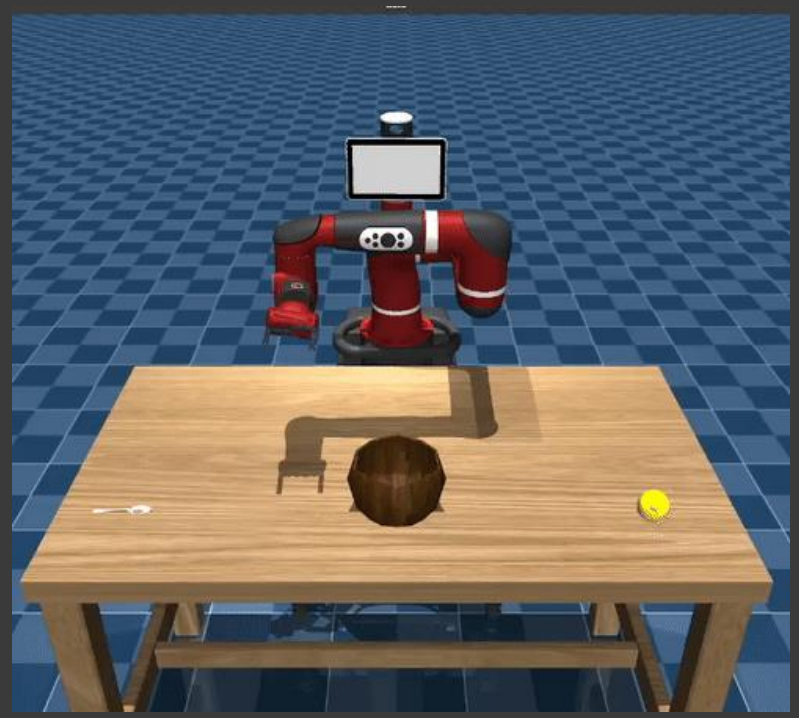
Number of objects : 3

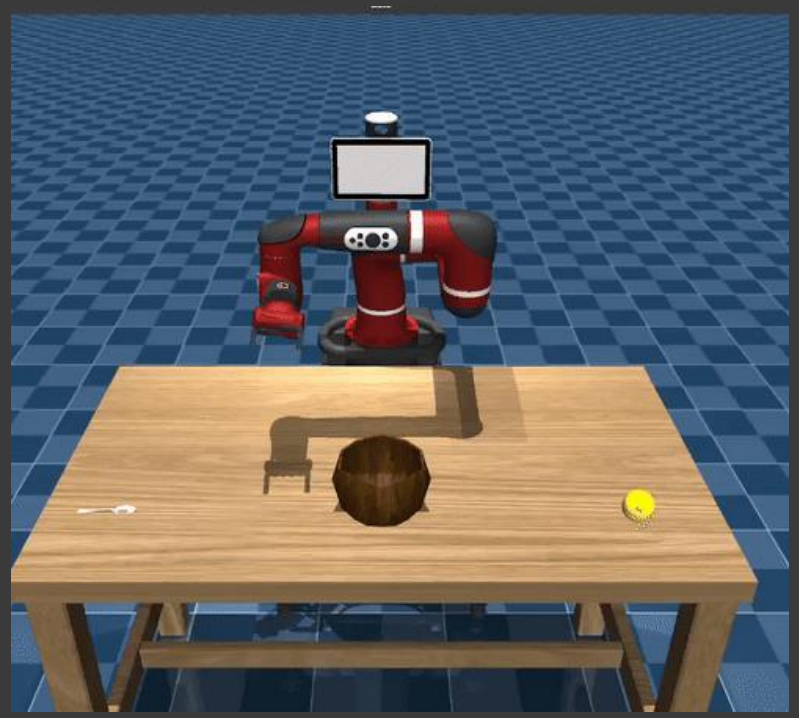
Temperature : 0.01



Simulation Results







Conclusion

- Successfully implemented POMDP in a simulator.
- Verified the belief distribution aligns with our goals.
- Incorporated human input to enhance observation updates.

Future Work

1. Test with solvers other than POMCP.
2. Evaluate belief updates using human input (consistent or legibility-based) and trajectory similarity to enhance human-robot interaction.
3. Expand the number of objects and vary the environment setups.

Questions?