Markov Decision Process

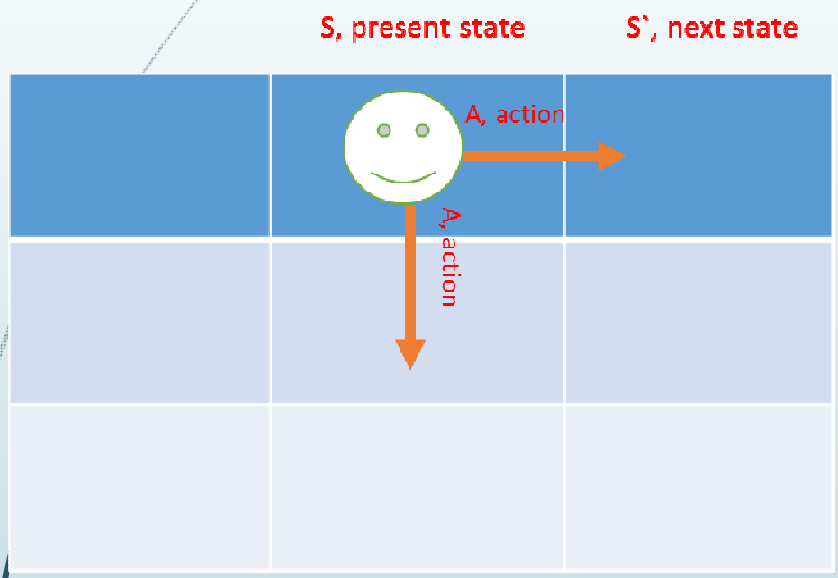
Dr Pradipta Biswas, PhD (Cantab)
Assistant Professor
Indian Institute of Science
https://cambum.net/

Contents

State-Space Model

- Initial State
- Operators: maps a state into a next state
  - alternative: successors of state
- Goal Predicate: test to see if goal achieved
- Optional:
  - cost of operators
  - cost of solution

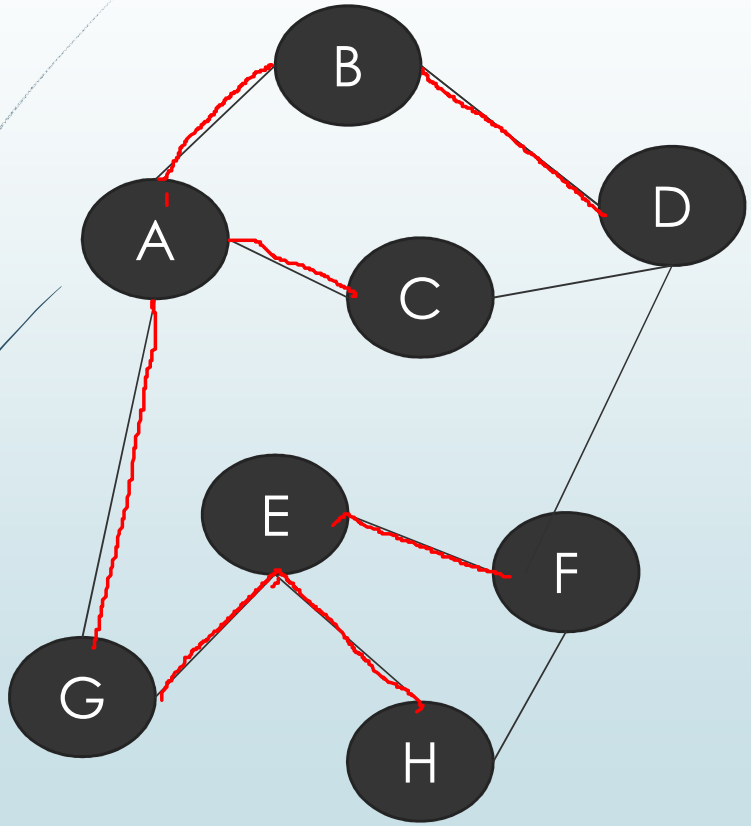
AI Memory Card

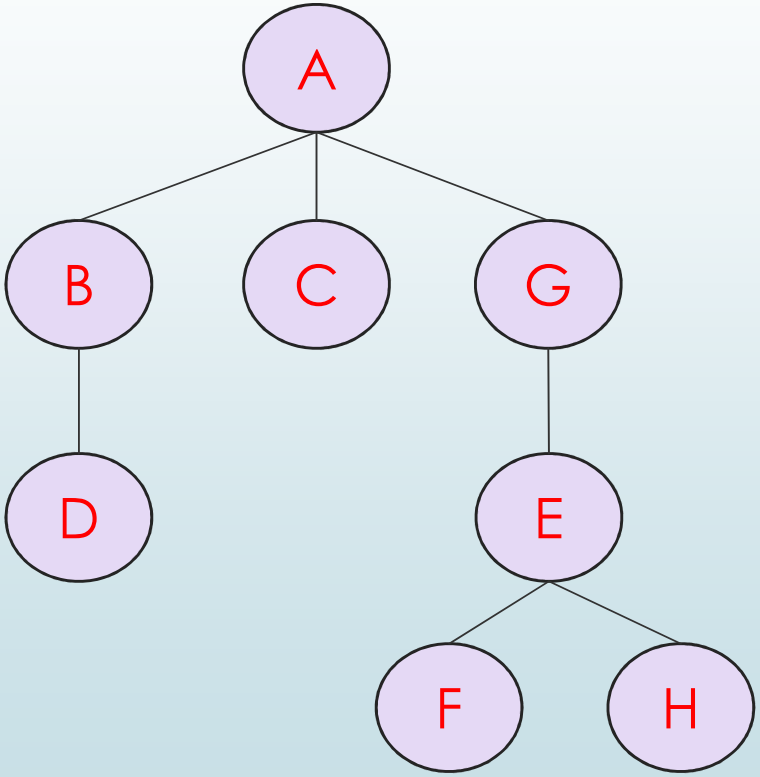


T(S,A,S`) is known as State Transition Matrix

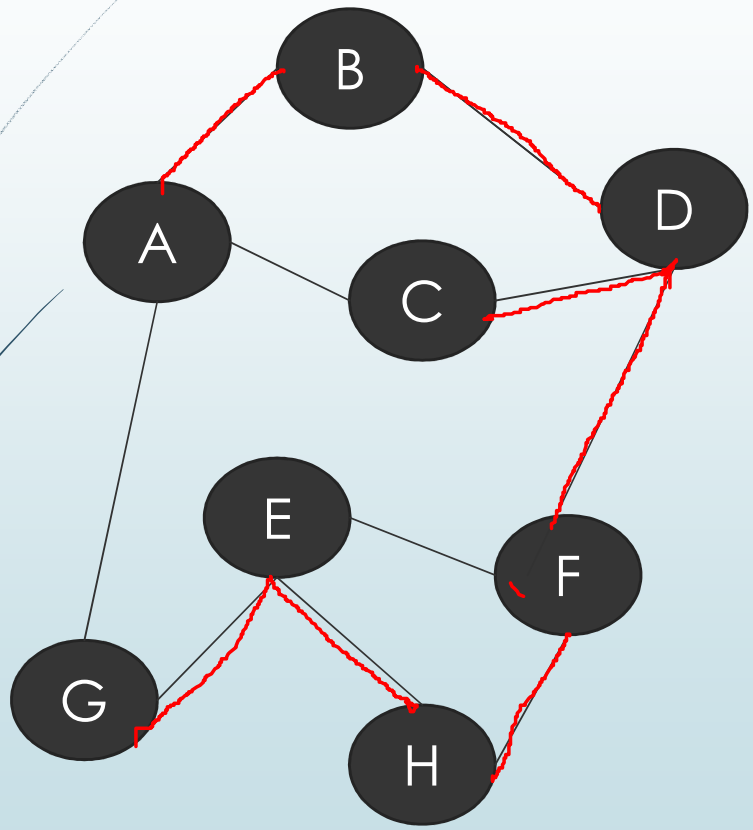
- T(S,A,S`) = Deterministic actual values => Uninformed state space search like DFS, BFS …
- T(S,A,S`) = Deterministic estimated values => Informed state space search like A*, MA*…
- T(S,A,S`) = Probabilistic values => Markov Decision Process (MDP)
- T(S,A,S`) = Unknown => Reinforcement Learning

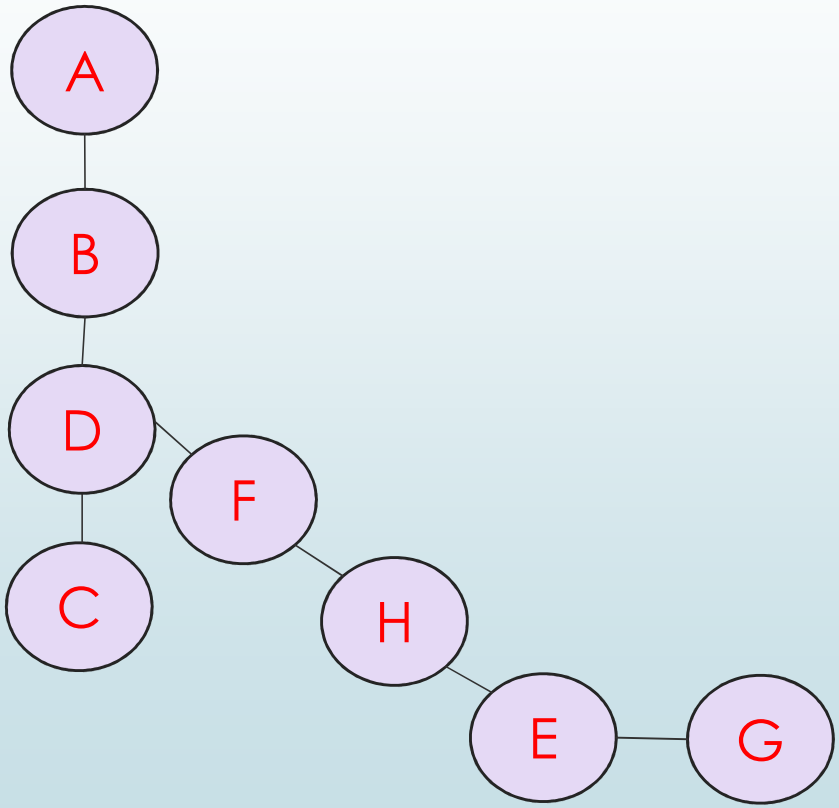
Uninformed Search - BFS



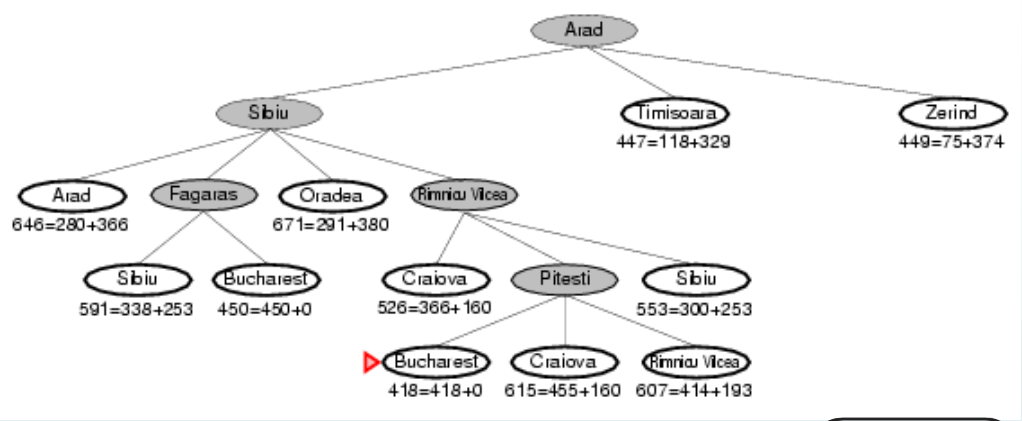


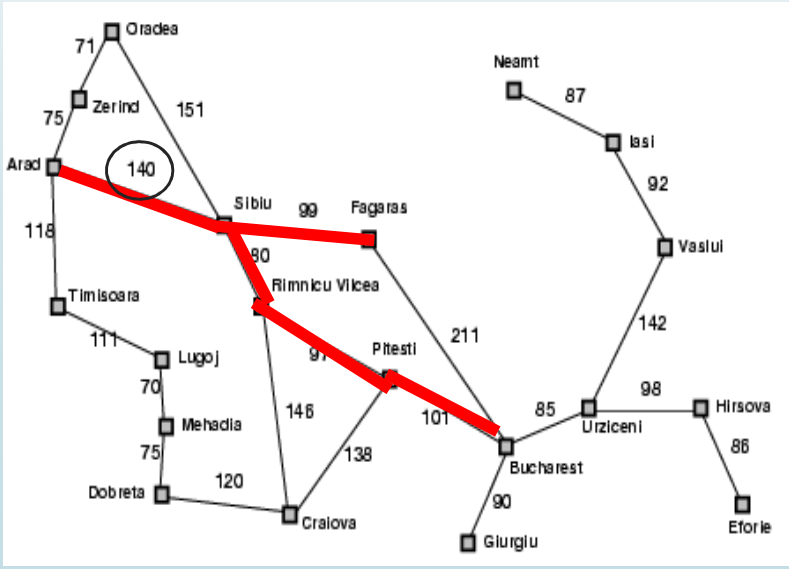
Uninformed Search - DFS





Informed Search -A* search

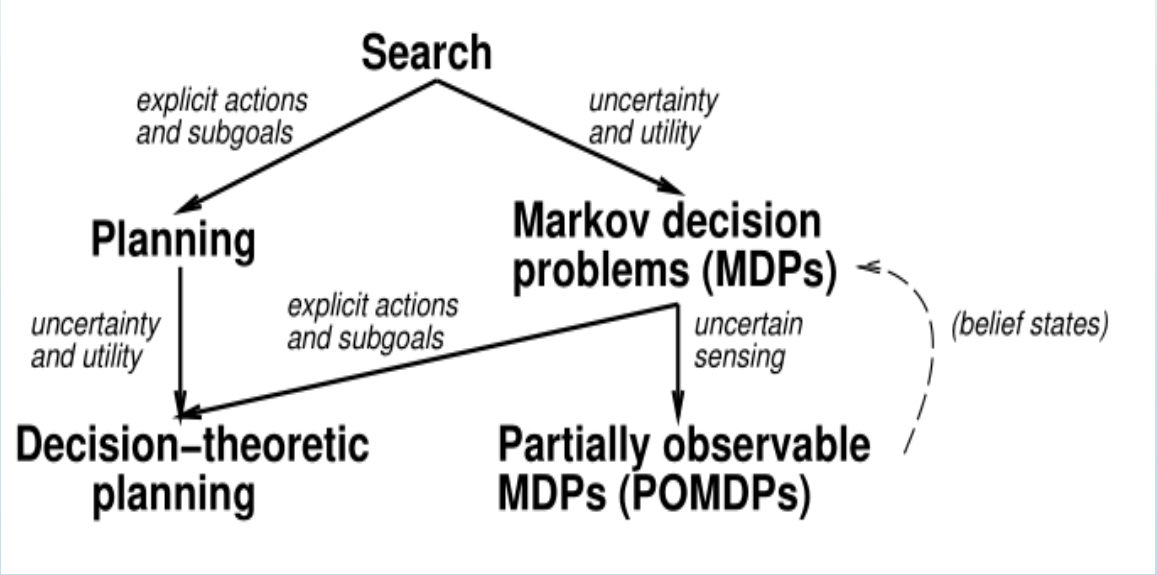




Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

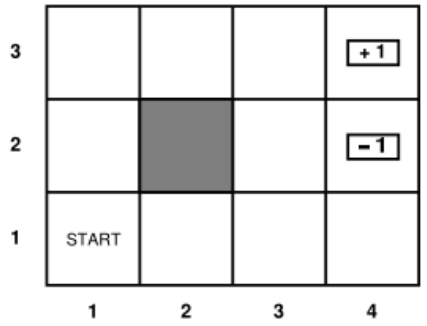
Sequential Decision Problems

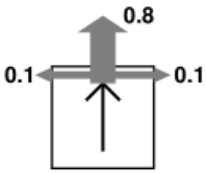
- Sequential decision problems vs. episodic ones
- Fully observable environment
- Stochastic actions



Markov Decision Process (MDP)

1. S_0. Initial state.
2. T(s, a, s'). Transition Model. Yields a 3-D table filled with probabilities. Markovian
3. R(s). Reward associated with a state. Short term reward for state





Model $M_{ij}^a \equiv P(j|i,a)$ = probability that doing a in i leads to j

Each state has a *reward* $R(i)$
= -0.04 (small penalty) for nonterminal states
= ± 1 for terminal states

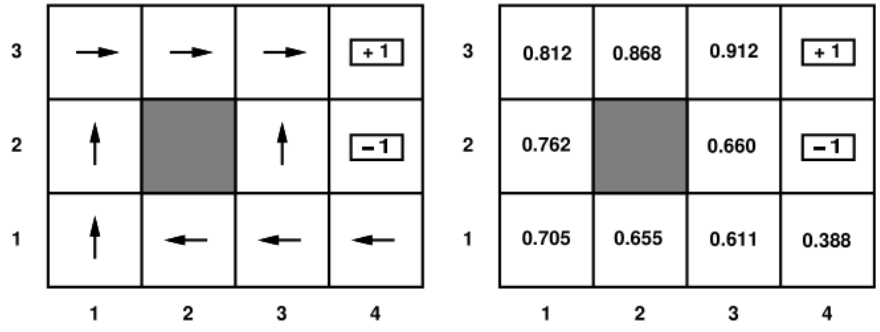
Policy, π

- A complete mapping from states to actions
- π(s). Action recommended in state s. This is the solution to the MDP.
- π*(s). Optimal policy. Yields MEU (maximum expected utility) of environment histories.

In search problems, aim is to find an optimal *sequence*

In MDPs, aim is to find an optimal *policy*
i.e., best action for every possible state
(because can't predict where one will end up)

Optimal policy and state values for the given $R(i)$:



Utility Function
(Long term reward for state)

- Additive. $U_h([s_0, s_1, \ldots]) = R(s_0) + R(s_1) + \ldots$
- However, there are problems because infinite state sequences can lead to +infinity or –infinity.
  - Set R_{max}, $\gamma < 1$
  - Guarantee that agent will reach goal
  - Use average reward/step as basis for comparison
- Discounted. $U_h([s_0, s_1, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots \quad 0 <= \gamma <= 1$

Value Iteration

- MEU principle
  $\pi^*(s) = \text{argmax}_a \sum_{s'} T(s, a, s') * U(s')$
- Bellman Equation
  $U(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s') * U(s')$
- Bellman Update
  $U_{i+1}(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s') * U_i(s')$
- Convergence guaranteed!
- Unique solution guaranteed!

Policy Iteration

1. The policy iteration algorithm works by picking a policy, then calculating the utility of each state given that policy. It then updates the policy at each state using the utilities of the successor states

2. Policy Evaluation. Given a policy π_i, calculate the utility of each state.

3. Policy Improvement. Use a one step look ahead to calculate a better policy, π_{i+1}

Bellman Update

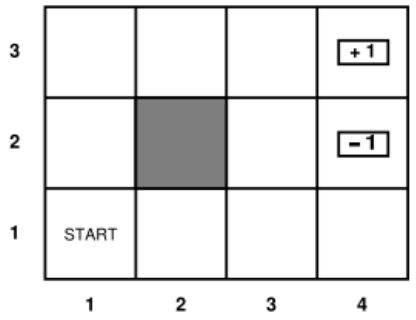
- Old method. $U_{i+1}(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s') * U_i(s')$
- New method. $U_{i+1}(s) = R(s) + \gamma \sum_{s'} T(s, \pi_i(s), s') * U_i(s')$
- Run k updates for an estimation of the utilities. This is called *modified policy iteration*
- N linear equations in N unknowns

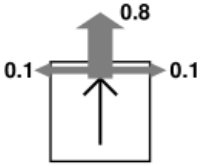
Linear Equations in Policy Iteration

$$U(i) = R(i) + \sum_j M_{ij}^{P(i)} U_t(j)$$

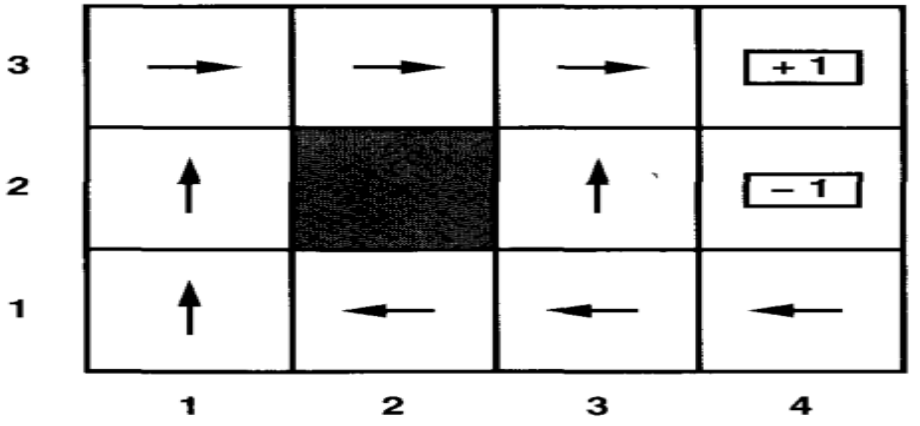
$$u_{(1,1)} = 0.8u_{(1,2)} + 0.1u_{(1,1)} + 0.1u_{(2,1)}$$

$$u_{(1,2)} = 0.8u_{(1,3)} + 0.2u_{(1,2)}$$





Model $M_{ij}^a \equiv P(j|i,a)$ = probability that doing a in i leads to j

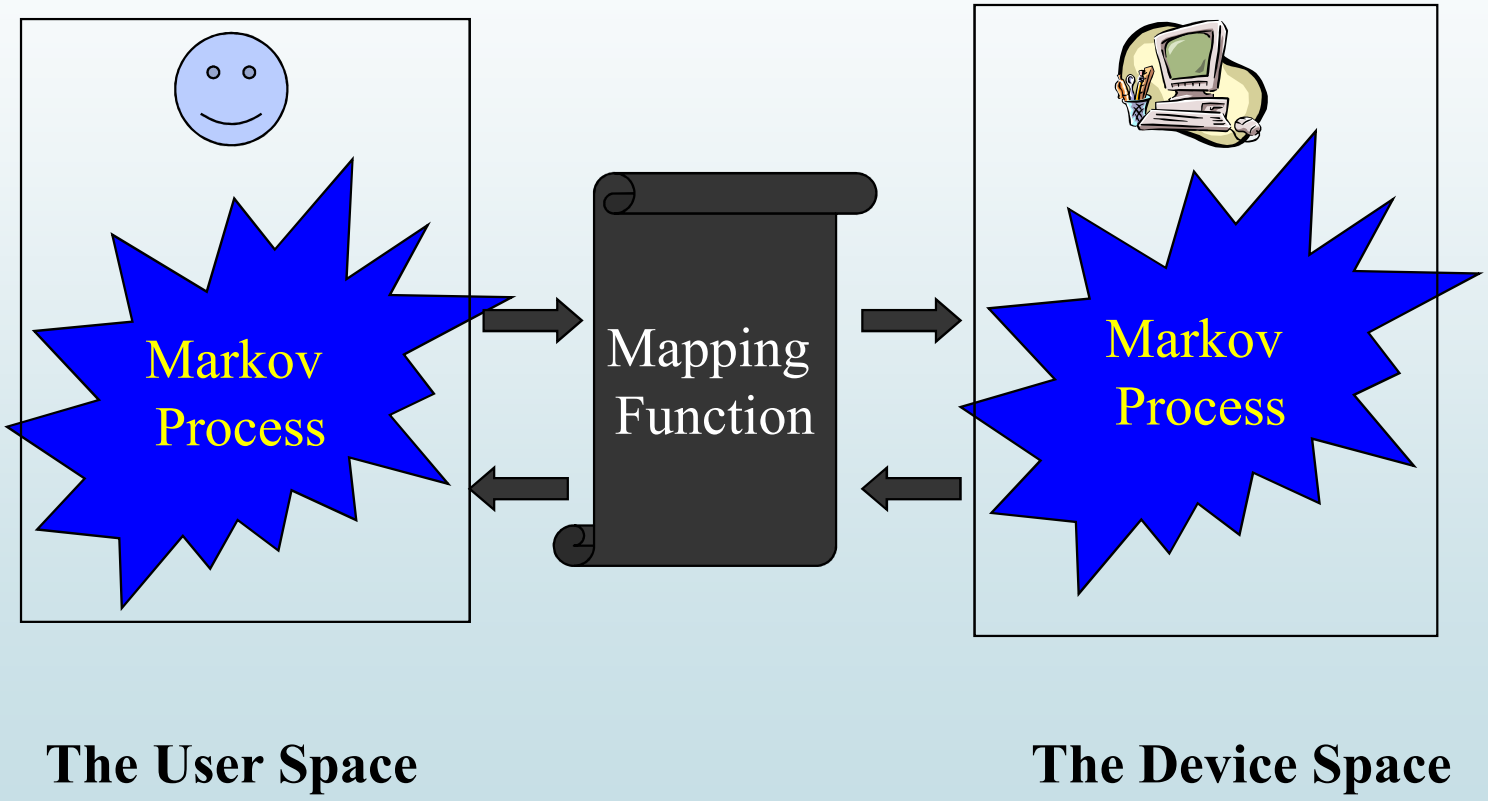


Case Studies

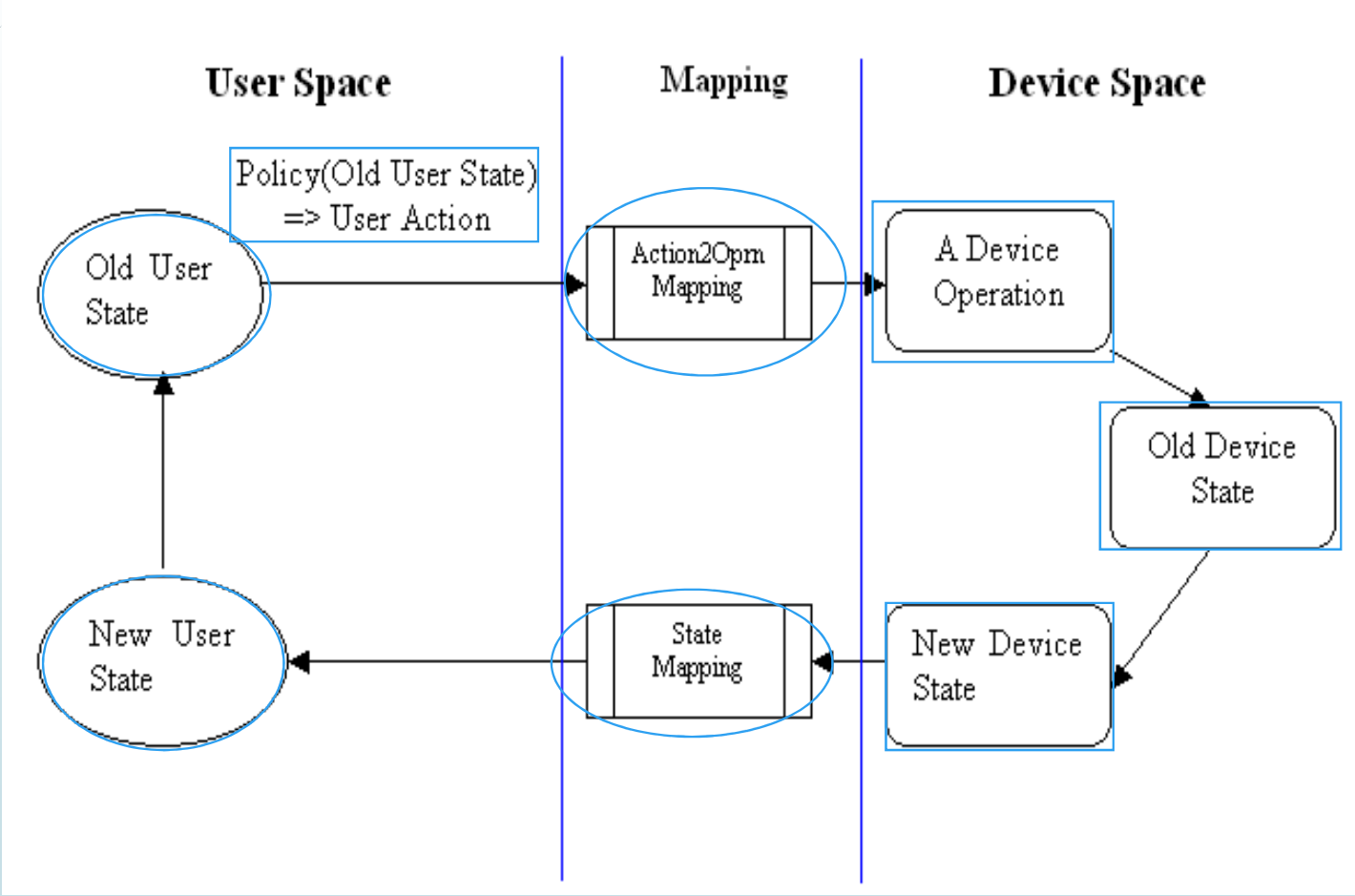
The Cognitive Model

P. Biswas and P. Robinson, Automatic Evaluation of Assistive Interfaces, Proceedings of the ACM International Conference on **Intelligent User Interfaces (IUI)** 2008, pp. 247-256

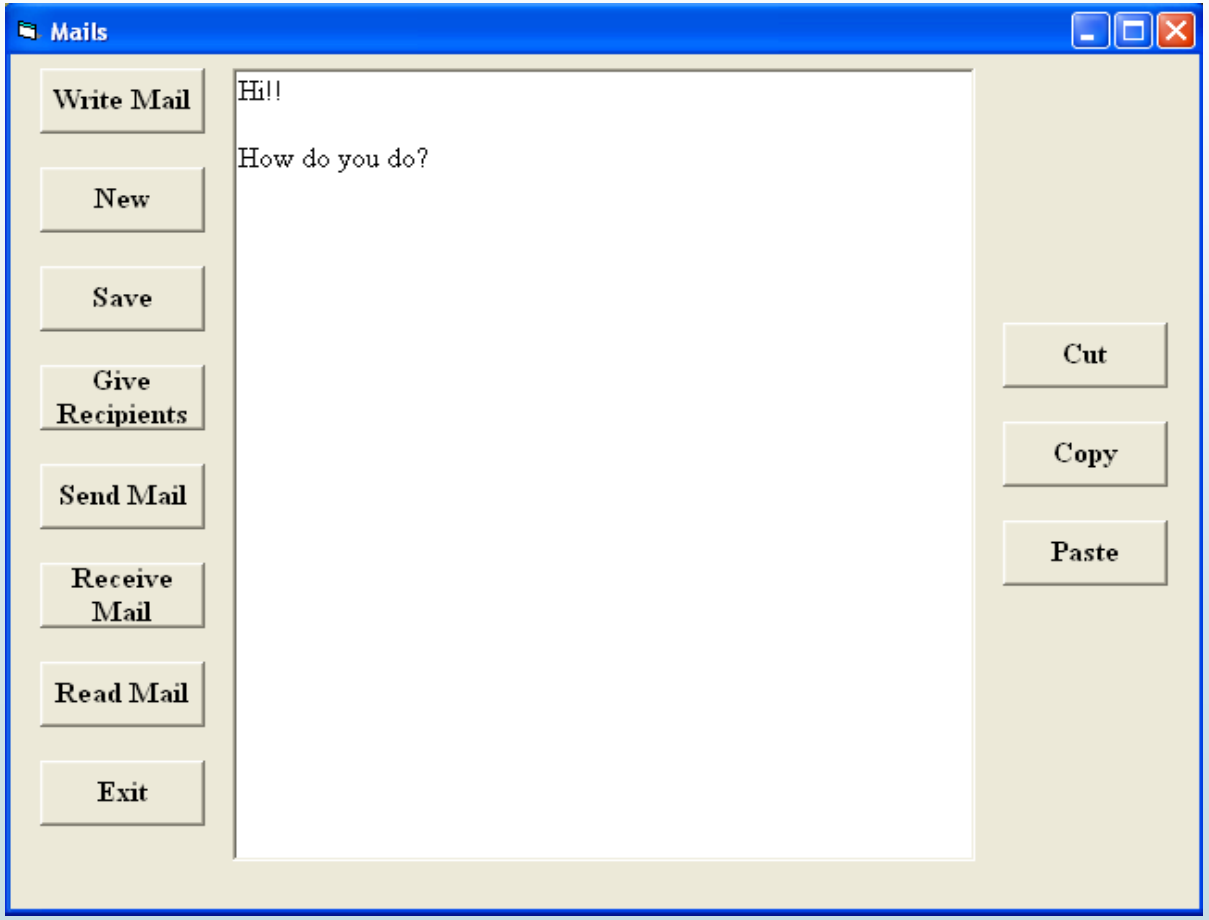
Assumptions for the Cognitive Model



Performance



Sample Application

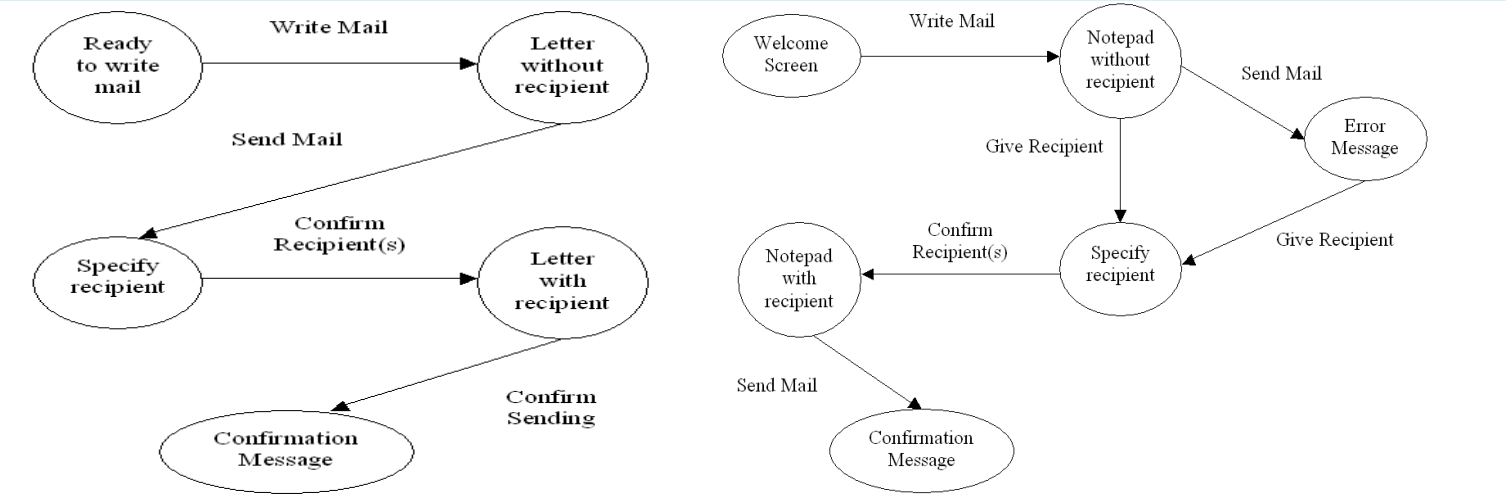


State Spaces

User Space	Device Space
States	
Ready to write mail	Welcome Screen
Letter without recipient	Notepad without recipient
Specify recipient	Specify recipient
Letter with recipient	Notepad with recipient
Confirmation Message	Confirmation Message
Actions	
Write Mail	Write Mail
Send Mail	Send Mail
Confirm Recipient(s)	Confirm Recipient(s)

State Spaces

User Space	Device Space
States	
Ready to write mail	Welcome Screen
Letter without recipient	Notepad without recipient
Specify recipient	Specify recipient
Letter with recipient	Notepad with recipient
Confirmation Message	Confirmation Message
Actions	
Write Mail	Write Mail
Send Mail	Send Mail
Confirm Recipient(s)	Confirm Recipient(s)



Execution of the Model

	Device Space	User Space
Iteration 1		
State	Welcome Screen	Ready to write mail
Action	Write Mail	WriteMail
State	Notepad without recipient	Letter without recipient
Action	SendMail	SendMail
State	ErrorMsg	
New Action Learned	**GiveRecipients**	
State	Specify recipient	Specify recipient
Action	ConfirmRecipient	ConfirmRecipient
State	Notepad with recipient	Letter with recipient
New Action Learned	SendMail	
Action	SendMail	Confirm Sending
State	Confirmation	Confirmation

Example of Learning

	Device Space	User Space
Iteration 2		
State	Welcome Screen	Ready to write mail
Action	Write Mail	WriteMail
State	Notepad without recipient	Letter without recipient
Action	GiveRecipients	GiveRecipients
State	Recipient	Recipient
Action	ConfirmRecipient	ConfirmRecipient
State	Notepad with recipient	Letter with recipient
Action	SendMail	Confirm Sending
State	Confirmation	Confirmation

Features of Cognitive Model

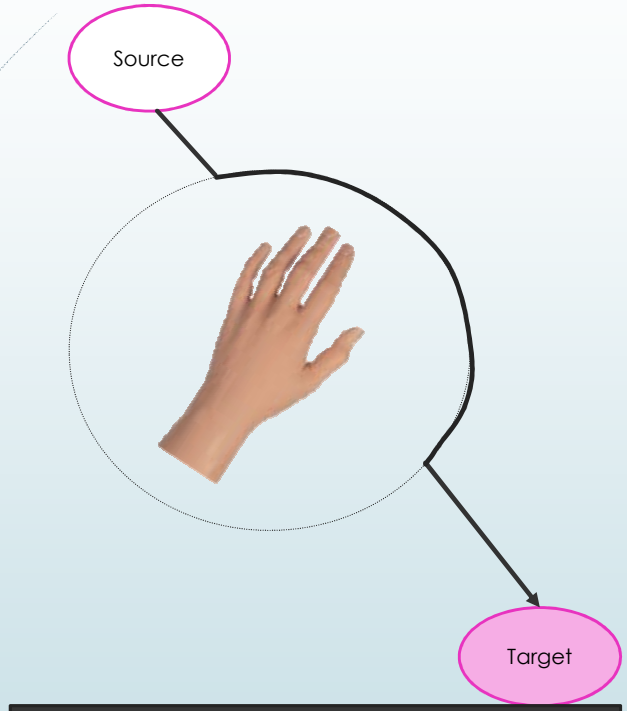
- Learn a new state or a new operation.
- Support the label matching principle.
- Take instruction during execution.
- Model the practice effect.
- Has user-friendly interfaces for development and execution.

Gaze Controlled Safe HRI for Users with SSMI

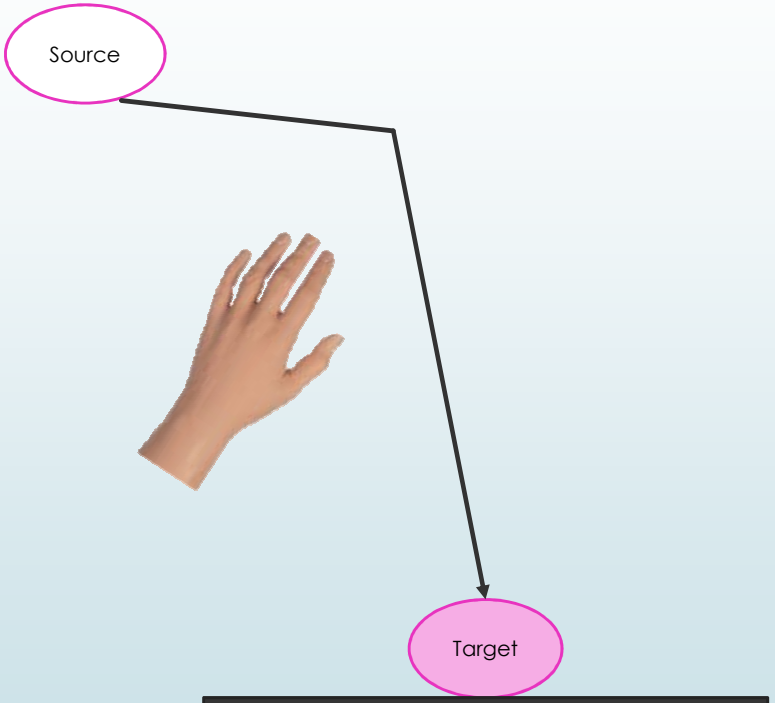
Vinay K Sharma, LRD Murthy, Pradipta Biswas

I^{3}D Lab, IISc

Avoiding Obstacles

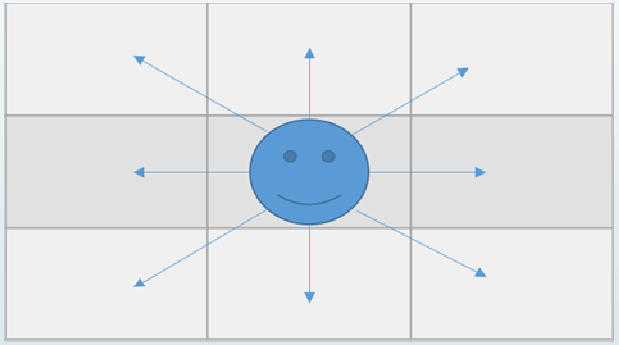


Navigation path with naïve geometry based algorithm

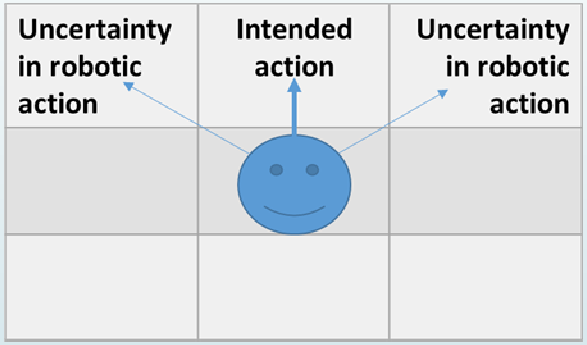


Navigation path with intelligent algorithm

Uncertainty Modelling

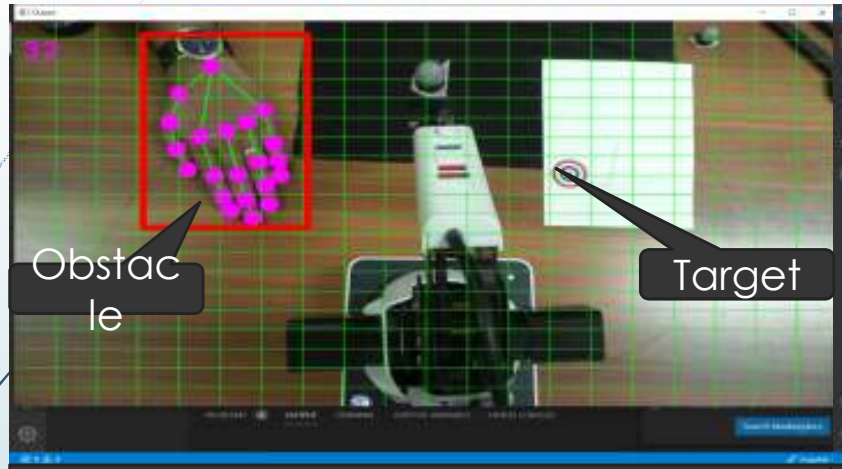


Eight possible actions of robotic agent

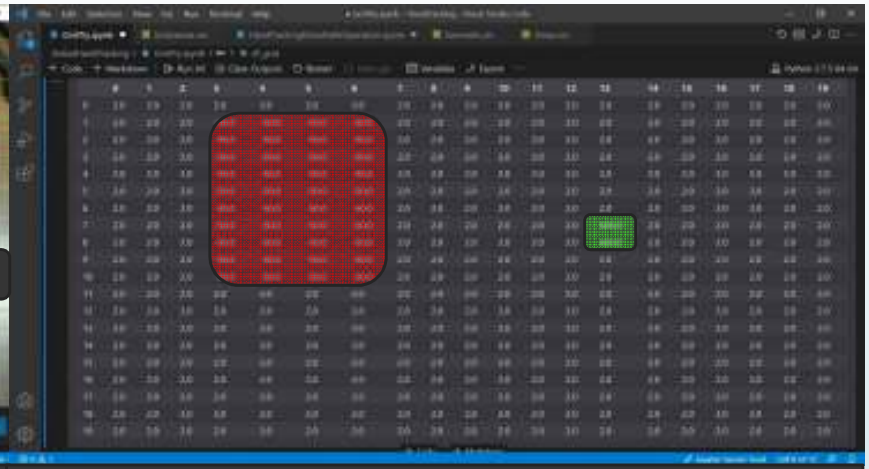


Modelling uncertainty in robotic action

Proposed State Space Model & MDP



a. Screenshot of actual set up



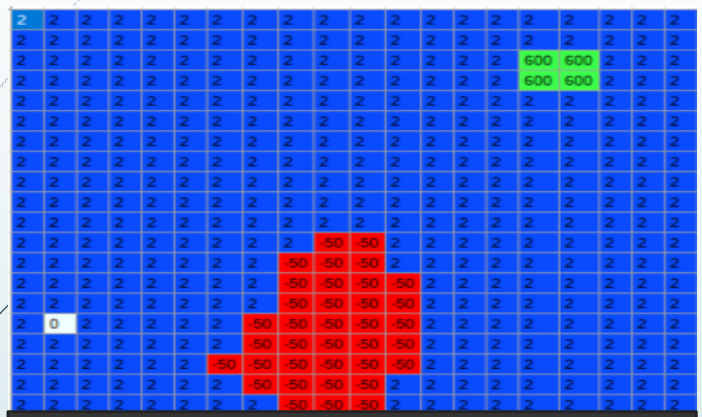
b. State space model with initial reward

c. Navigation path with no uncertainty in robotic movement

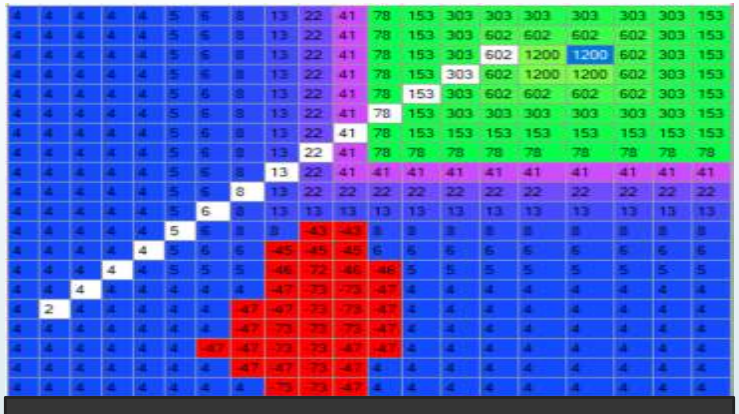
d. Navigation path with 0.05 uncertainty in robotic movement to move on neighbouring cells

e. Navigation path with 0.3 uncertainty in robotic movement to move on neighbouring cells

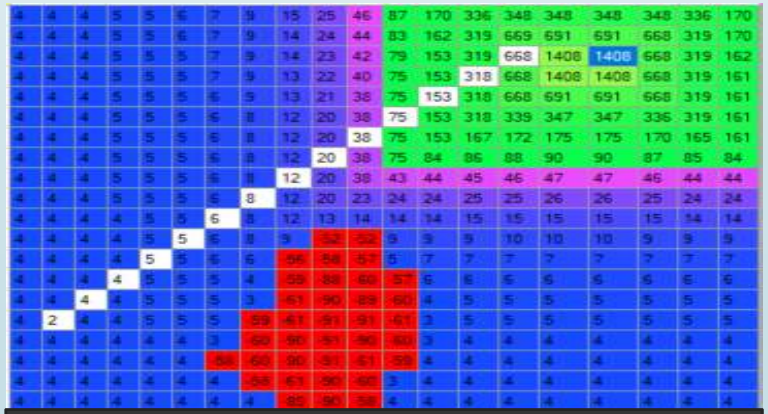
Navigation Under Uncertainty



a. Source, target and hand region



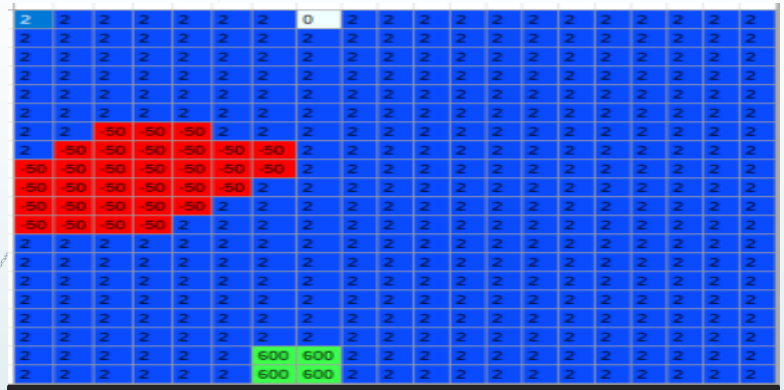
b. Navigation path with no uncertainty in robotic movement



c. Navigation path with 0.05 uncertainty in robotic movement to move on neighbouring cells

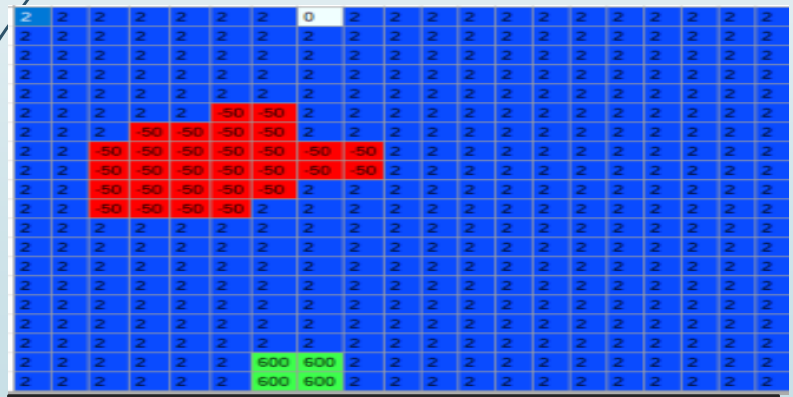
d. Navigation path with 0.3 uncertainty in robotic movement to move on neighbouring cells

Remembering Past Positions



a. Source, target and hand region

b. Navigation path

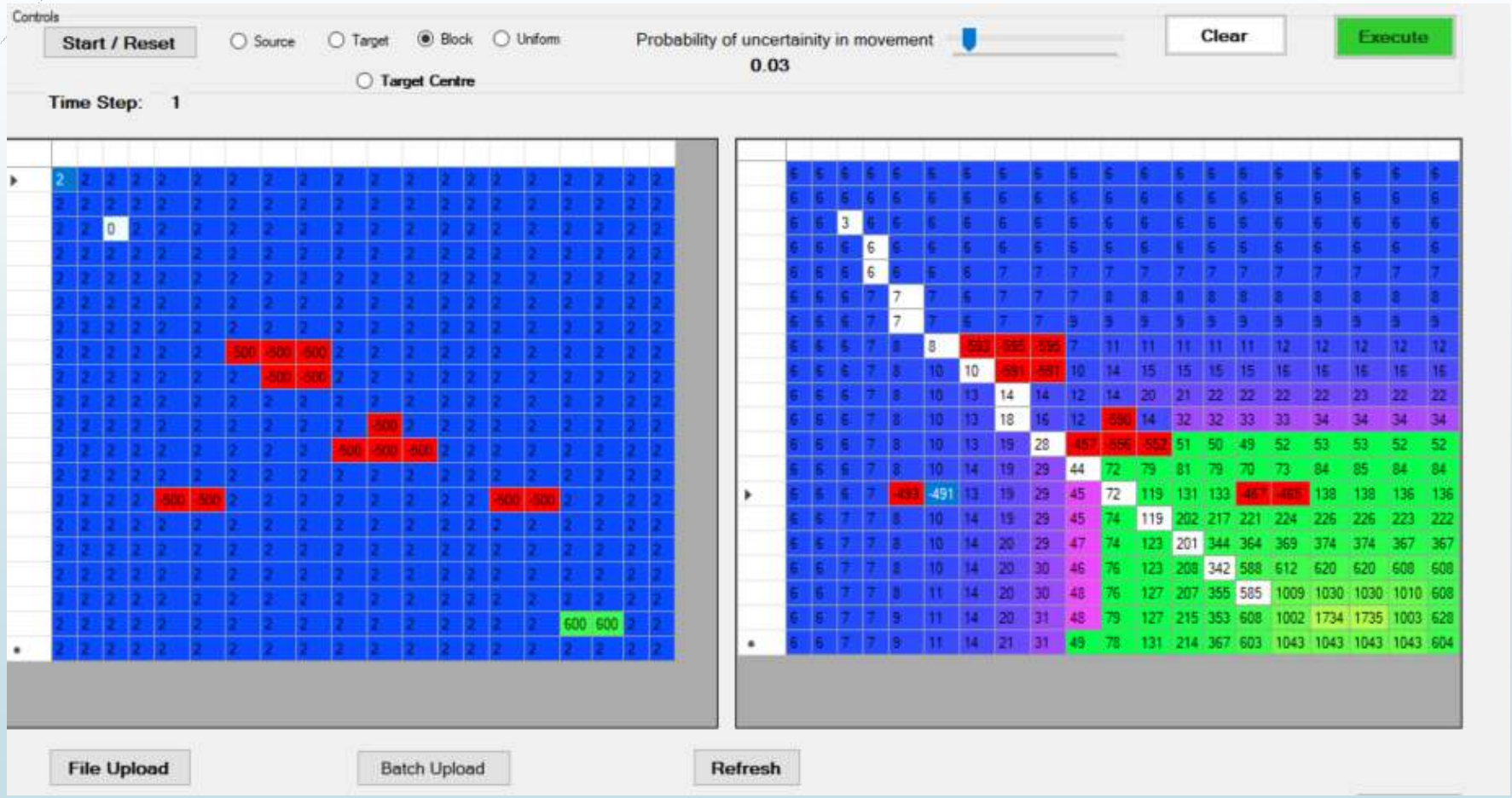


c. Hand position moved to right

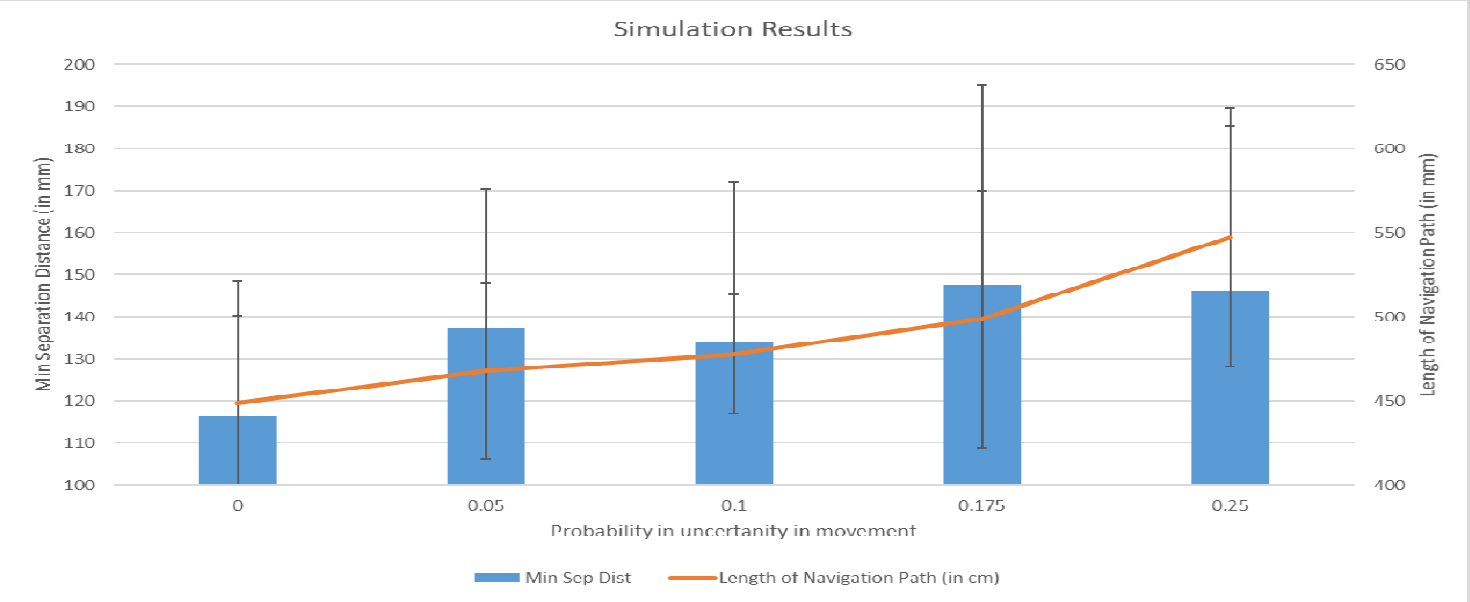
d. Navigation path remembering previous position

$$U^*(s, t) = U^*(s, t) + \alpha\, U^*(s, t-1),\ 0 < \alpha < 1$$

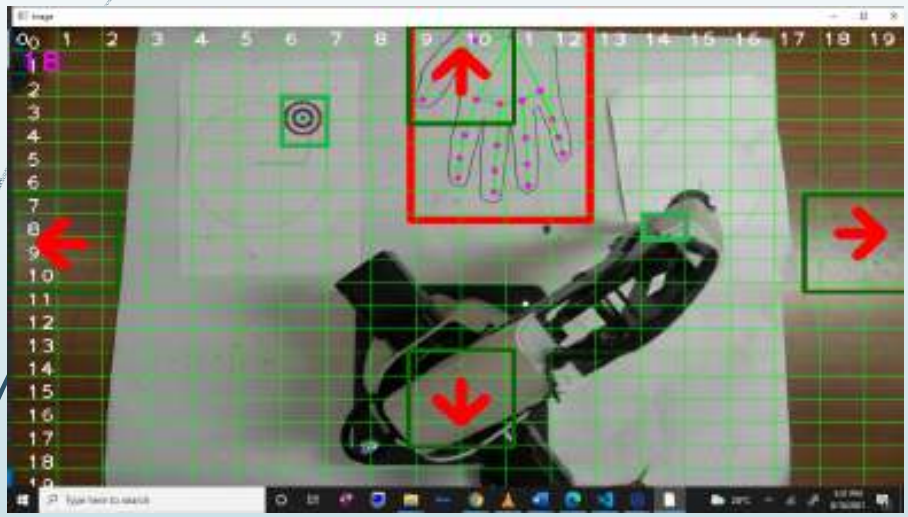
Multiple Obstacles

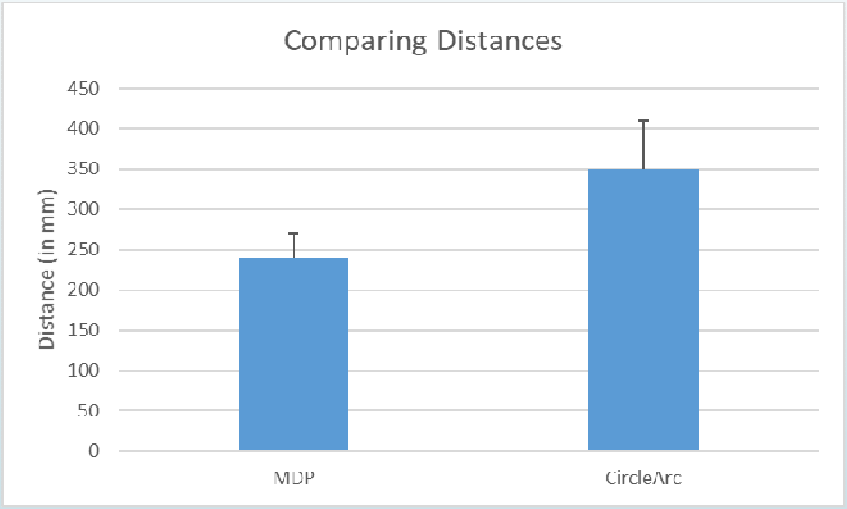


Simulation Result

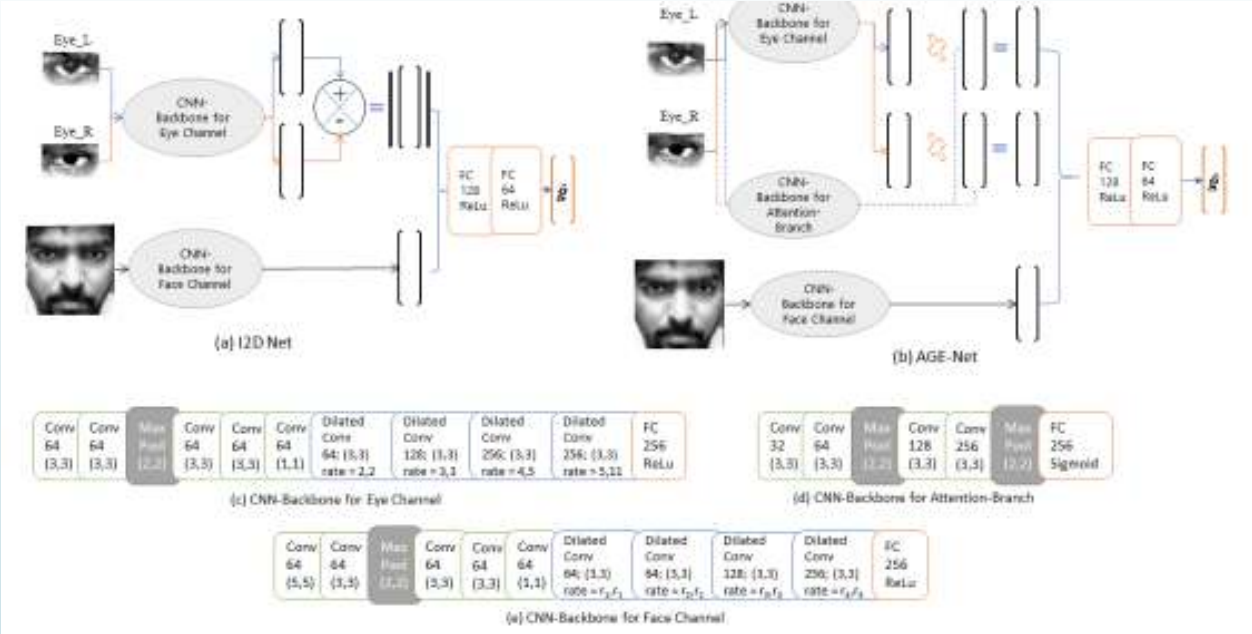


Simulation Result



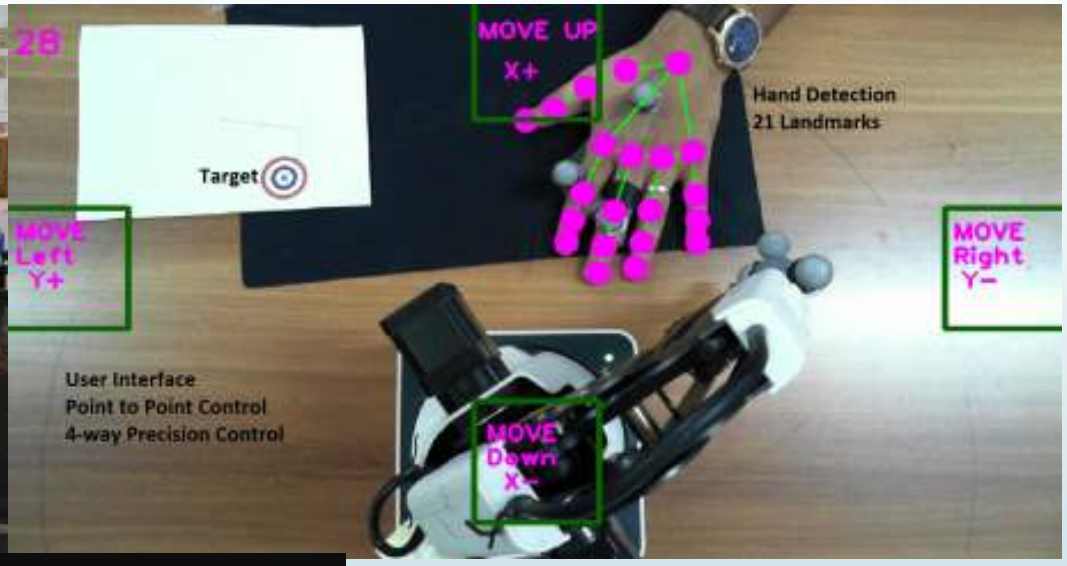


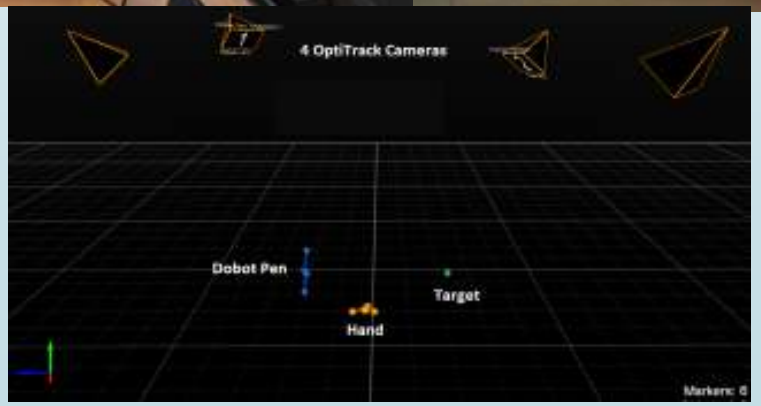
Eye Gaze Tracking



LRD Murthy and P. Biswas, Appearance-based Gaze Estimation using Attention and Difference Mechanism, 3rd International Workshop on Gaze Estimation and Prediction in the Wild (GAZE 2021) at CVPR 2021

Experiment Set Up

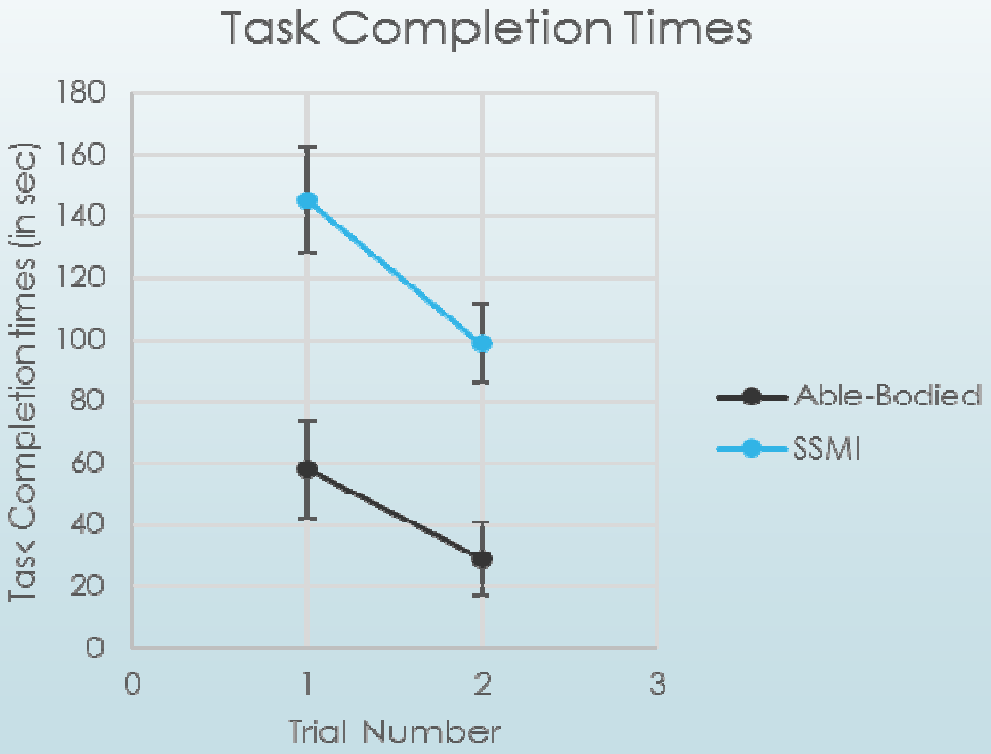




Gaze Controlled Safe HRI for Users with SSMI

User Study

- 12 Users (5 users with SSMI and 7 able-bodied users)
- Dobot Magician System
- Two reachability tasks for randomly positioned target
- Significant main effect of *Trial Number*
  - $F(1,11)=8.648,\ p<0.05,\ \eta^2=0.44$
- Significant main effect of *Participant Type*
  - $F(1,\ 11)=106,16,\ p < 0.05,\ \eta^2=0.906$



Take Away Points

- Modelling Uncertainty using Markov Decision Process
- Bellman Ford Update
- Value and Policy Iteration Algorithms
- Case Studies
  - Cognitive Model using two parallel MDP
  - Navigation of a Robotic Manipulator