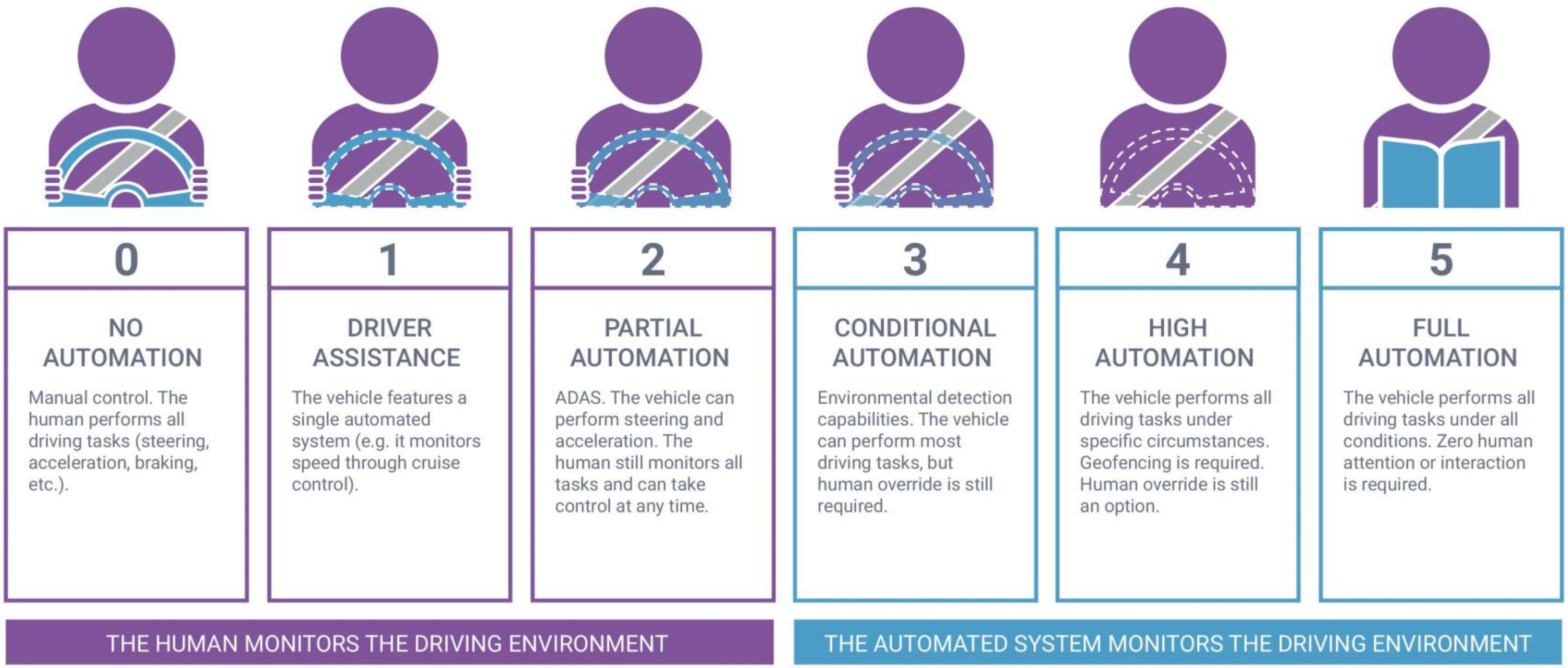
MODULE 4: HUMAN FACTORS IN AUTONOMOUS DRIVING

Nicholas Ho, PhD
Institute of System Science, NUS

SAE International: Levels of Automation



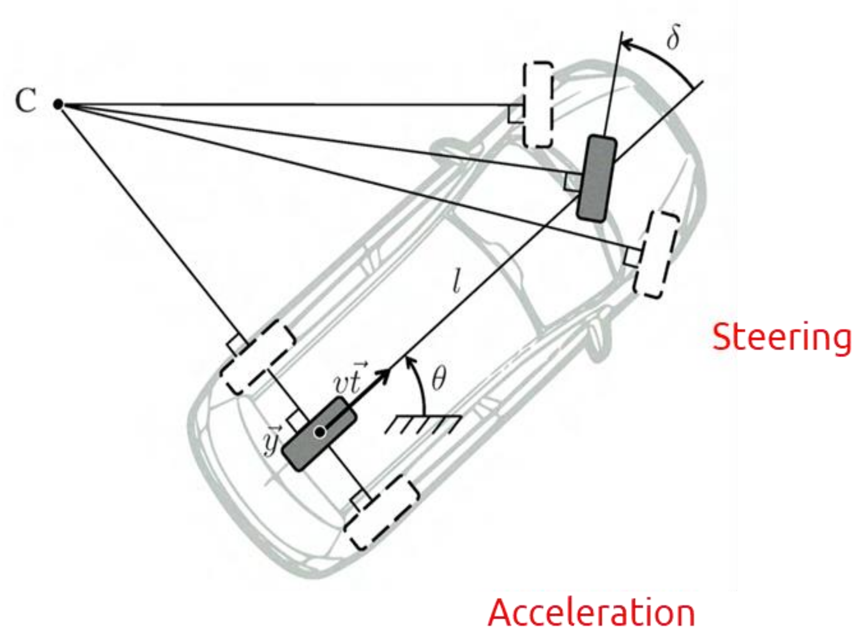
SAE International: Levels of Automation

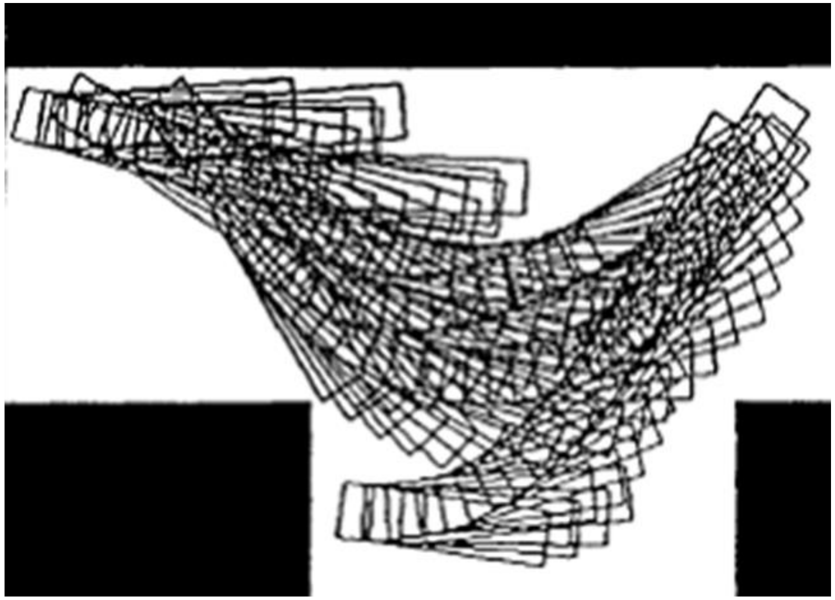
1 Driver Assistance	2 Partial Automation	3 Conditional Automation	4 High Automation	5 Full Automation
Example • City Emergency Braking	**Example** • Adaptive Cruise Control	**Example** • Highway Pilot	**Example** • City & Highway Pilot	**Example** • Robocar
Definition • Steering OR acceleration / deceleration • Human supervision	**Definition** • Steering AND acceleration / deceleration • Human supervision	**Definition** • All driving functions • Human intervention may be needed	**Definition** • All driving functions • Human can but don’t has to intervene	**Definition** • All driving functions • No human intervention possible

LEVEL 1: DRIVER ASSISTANCE

A SINGLE AUTOMATED SYSTEM

Nonholonomic car model

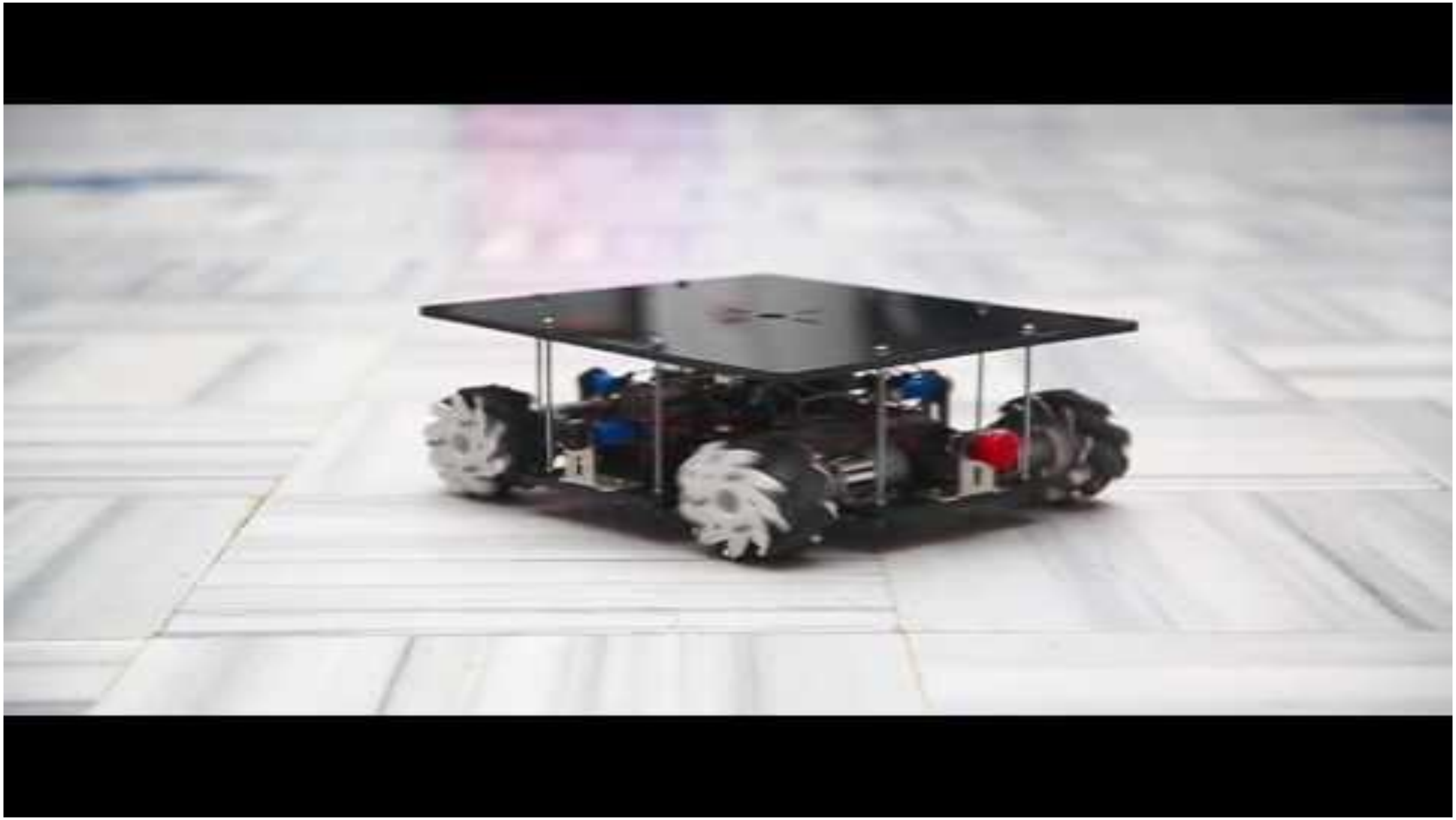




Holonomic robot model

Holonomic system where a robot can move in any direction in the configuration space

NAMLA Autonomous Omni-Directional Wheeled (i.e. Mecanum Wheeled) Mobile Robot



Source: https://www.youtube.com/watch?v=xL8deJDusns

Acceleration Assistance: Adaptive Cruise Control

Source: https://www.youtube.com/watch?v=GInSPWZRFRM

Steering Assistance: Lane Keeping Assist

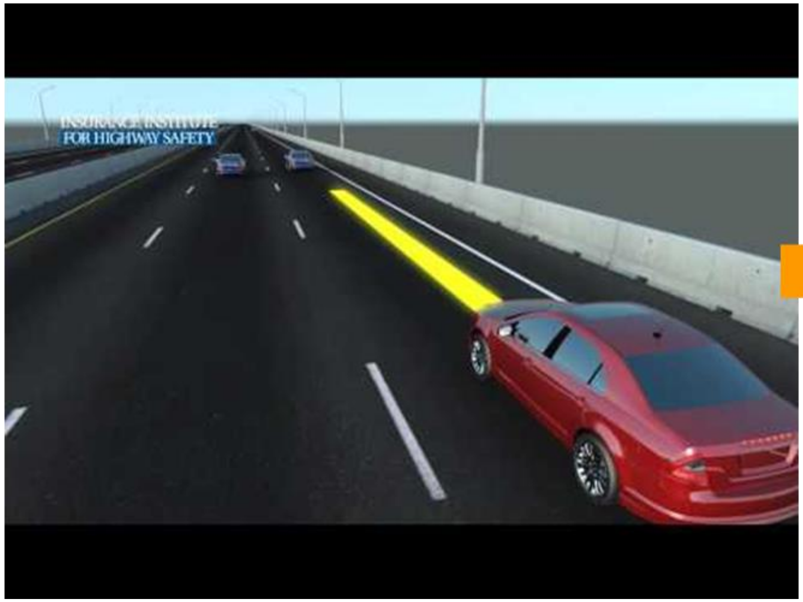
Source: https://www.youtube.com/watch?v=OQkdvi55woA

LEVEL 2: PARTIAL AUTOMATION

- STEERING + ACCELERATION
- HUMAN MONITORS ALL TASKS

Combination of Acceleration and Steering

Adaptive Cruise Control



Lane Keeping Assist

SAE Level 2 Fatal Accident

Source: https://www.youtube.com/watch?v=CgLE_ZLLaxw

5 fatal accident (4 in US, 1 in CN)

All are driver fatalities

LEVEL 3: CONDITIONAL AUTOMATION

- MOST DRIVING TASKS
- HUMAN MONITORS ALL TASK

Smart Summon

Source: https://www.youtube.com/watch?v=nlCQG2rg4sw

Smart Summon (Cont)

Source: https://www.youtube.com/watch?v=3o2sI37xwOc

SAE Level 3 Fatal Accident

Source: https://www.youtube.com/watch?v=ou8sqWr6mF0

1 fatal accident in 2018

Pedestrian fatality

LEVEL 4: HIGH AUTOMATION

- ALL DRIVING TASKS UNDER SPECIFIC CIRCUMSTANCES
- GEOFENCING IS REQUIRED

LEVEL 5: FULL AUTOMATION

- ALL DRIVING TASKS UNDER ALL CONDITIONS

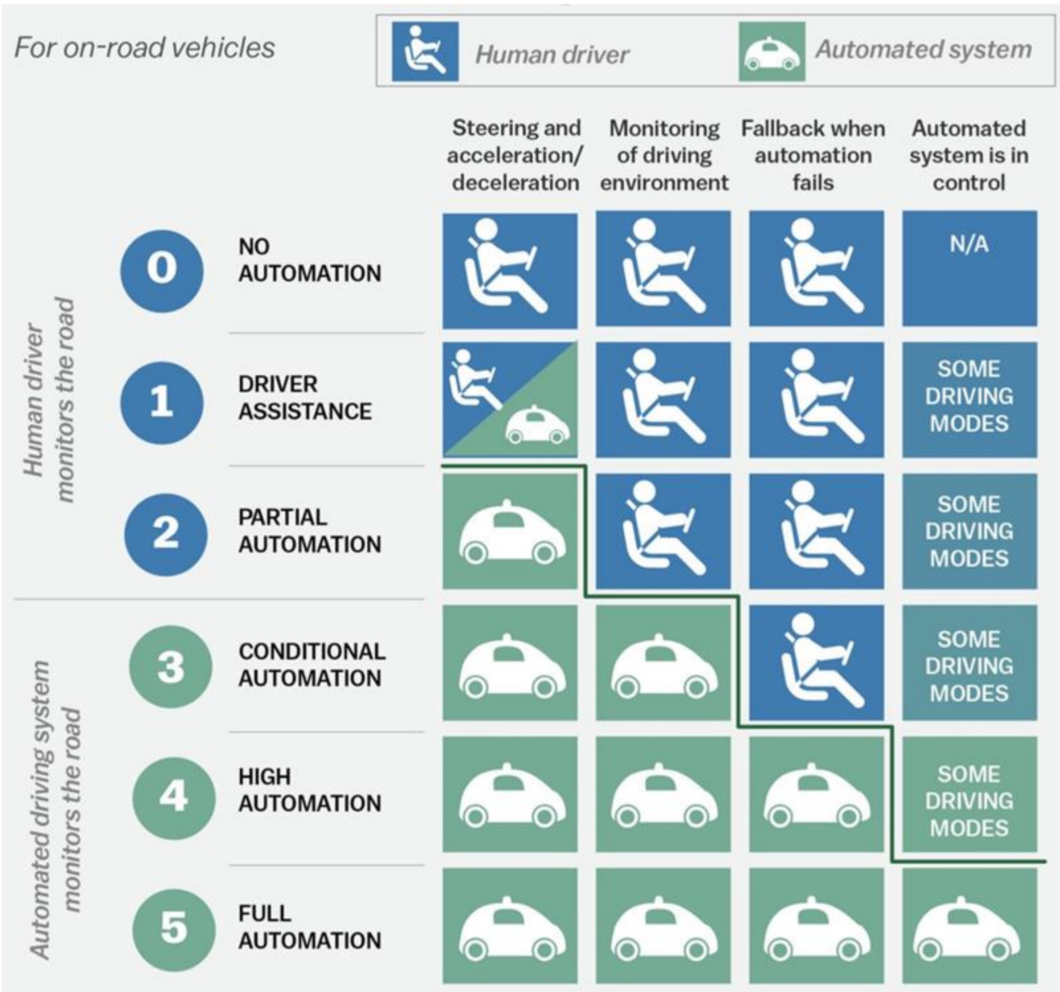
*A geofence is a virtual perimeter for a real-world geographic area; A predefined set of boundaries

Minor Accident in SG

The self-driving car was changing lanes in Biopolis Drive at one-north when it knocked into the lorry. The car was travelling at a "low speed" at the time of the accident --- 2016

What if ……

SAE International: Levels of Automation



- It is a complex interaction between human drivers and SAE Levels 2 and 3 autonomous vehicles
- Any automated system that removes the human from the driving task, yet requires the human to monitor and supervise the system and regain control when necessary, could be unsafe

‘Ironies of automation’

- **Irony:** combination of circumstances, the result of which is the direct opposite of what might be expected
- *“The mere fact that you can automate does not mean that you should”*
- Humans may **misuse**, **disuse** and **abuse** automation technology
- Humans tend to be **poor supervisors** of automation

‘Ironies of automation’

Source: https://www.youtube.com/watch?v=kzBNdssqrEA

Challenges

- **Driver inattention and distraction**
- **Situational awareness**
- **Overreliance and trust**
- **Skill degradation**
- **Motion sickness**

The Future???

- Re-engaging the driver
- The user interface and the communication of automation limitations
- Automation misuse and the need to monitor the driver
- The personalization of automation
- Acceptance

The Future???

THE STRAITS TIMES WORLD

Tesla crash victim had lauded 'full self-driving' in videos on TikTok

Since 2016, at least three Tesla vehicles operating on Autopilot have been in fatal crashes. PHOTO: REUTERS

The Future???

Source: https://www.youtube.com/watch?v=HI23Yiy-EAE

From 16:40

HUMAN FACTORS IN LEVEL 5 AUTOMATION

What if ……

traffic agents: pedestrians, bicycles, cars, buses, etc..

Uncertain future motions

Problem Statement

predict:

future positions of N agents for t_{pred} steps

given:

- history positions
- types (pedestrian, bicycle, car, etc.)
- positions of obstacles

Challenges of predicting their motions

Diverse dynamics, geometry, behaviors (heterogeneous)

Challenges of predicting their motions

Diverse dynamics, geometry, behaviors (heterogeneous)

Intensive interactions

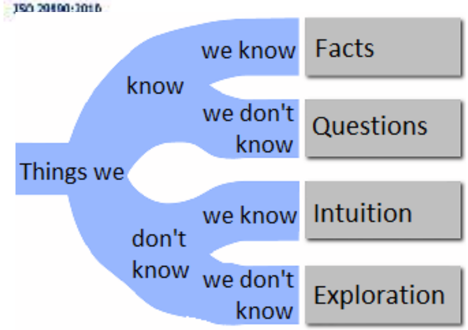
Challenges of predicting their motions

Diverse dynamics, geometry, behaviors (heterogeneous)

Intensive interactions

Complex road conditions

How real-world agents move?



intention kinematics

Traffic agents try to reach their destinations under their kinematic constraints, and in the meantime, they share the responsibility for collision avoidance with each other, while having limited attention capabilities

responsibility geometry

attention

We know these are unknown (uncertain)

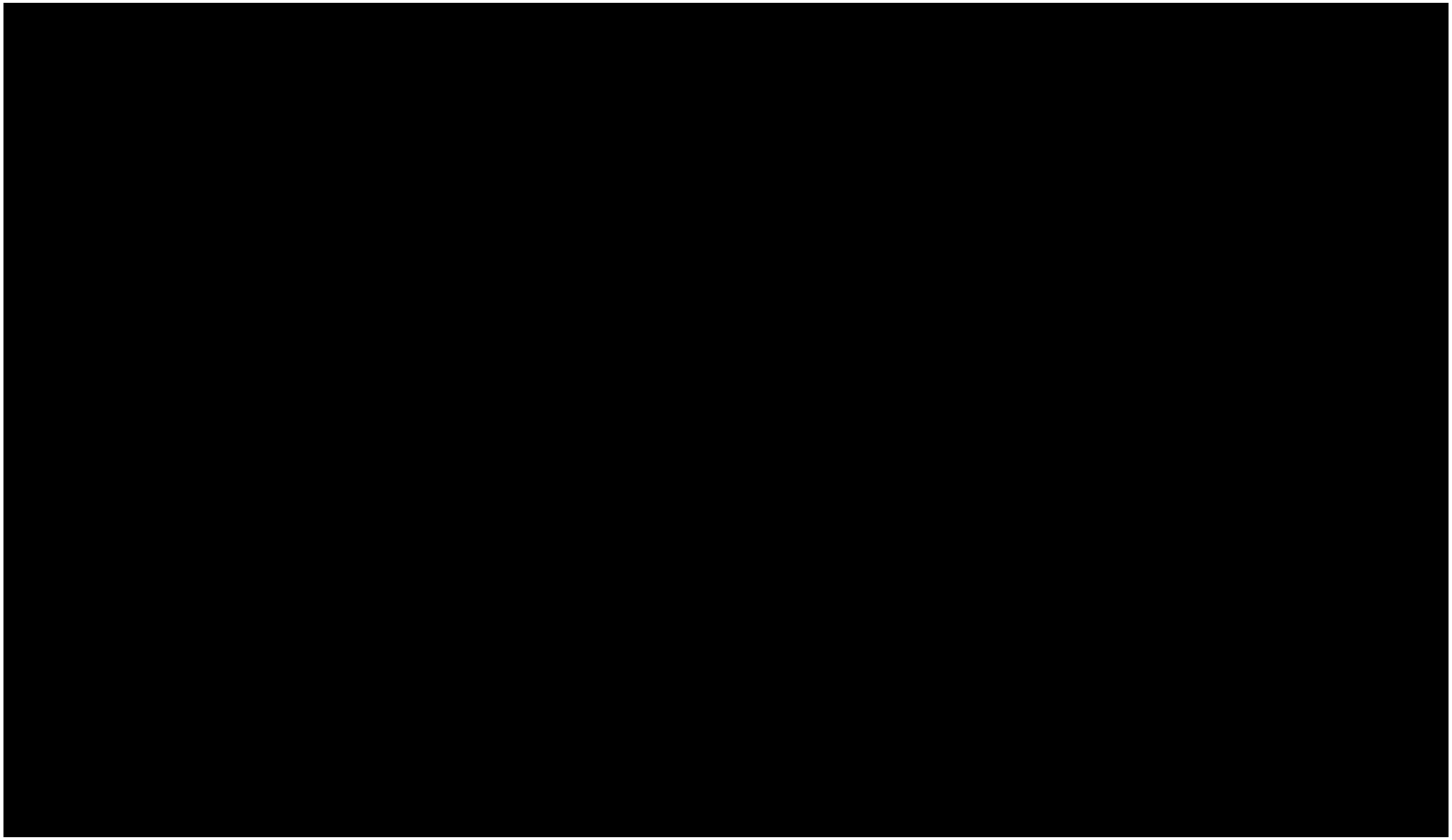
We know these are known (quite certain)

Example unknown known: driving is reactive?!

To overtake, the car needs to:

- Understand the stopped car's intention of not moving
- Understand the road context information such as left/right lane exist and is of same direction

Planning vs Reactive



Source: https://www.youtube.com/watch?v=SVPVTwxNRLc

MODULE 3 & 4 WORKSHOP: WORKSHOP DAY 2

WORKSHOP: HANDS-ON CONSTRUCTION OF MDP/HRIMDP MODELS OF AUTONOMOUS DRIVING

Personal Healthcare Robot

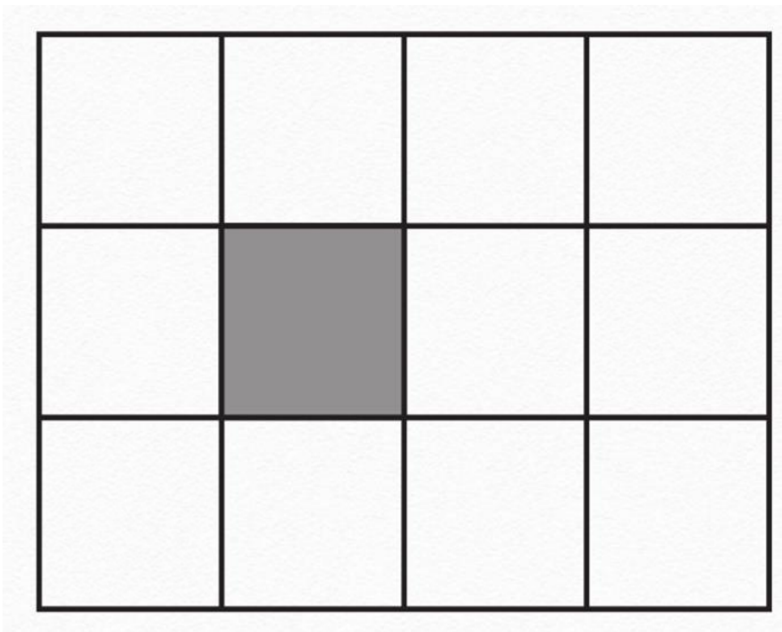
- Personal assistant robots gain their popularity, E.g., Google Home, Amazon Echo, Xiao Mi robot vacuum, ASUS Zenbo, etc
- They are perfect candidates to become the future personal healthcare givers that monitor the elderly users at the nursing home or in the hospital. In this workshop project, we formulate this interaction between the personal healthcare robot and the human user

Learning Objectives of Workshop Day 2

1. Basic programming that creates trajectories of human and robot movements within a virtual world
2. Basic programming that enables you to visualize these interactions between the human and robot (include movements and collision avoidance)
3. Implementation of a Human-Robot-Interaction-focused MDP method (named as HRIMDP) within a 5D gridworld
4. Understand the benefits of the HRIMDP method vs conventional MDP methods

Part A: Introduction to Grid World

Grid World



The perfect-sized environment to learn about Reinforcement Learning for the start

robot

human

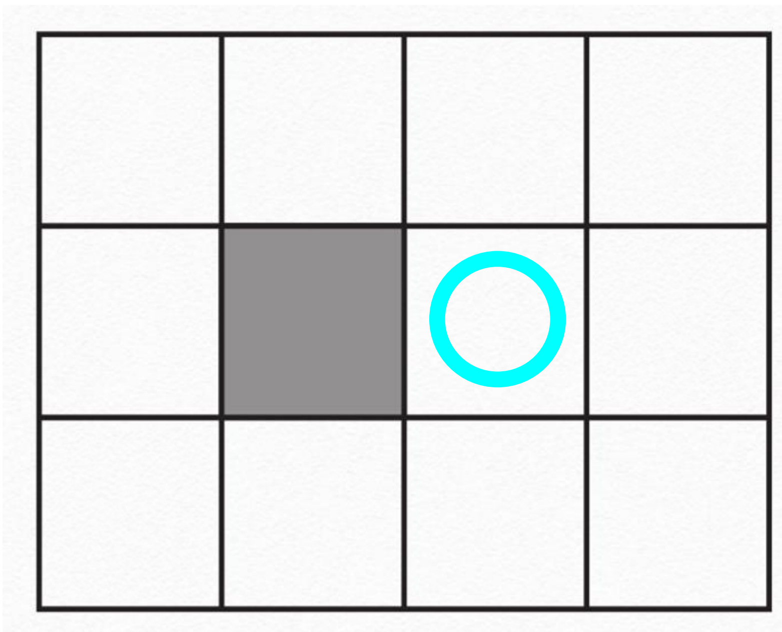
world1.csv

0,0,0,0

0,1,0,0

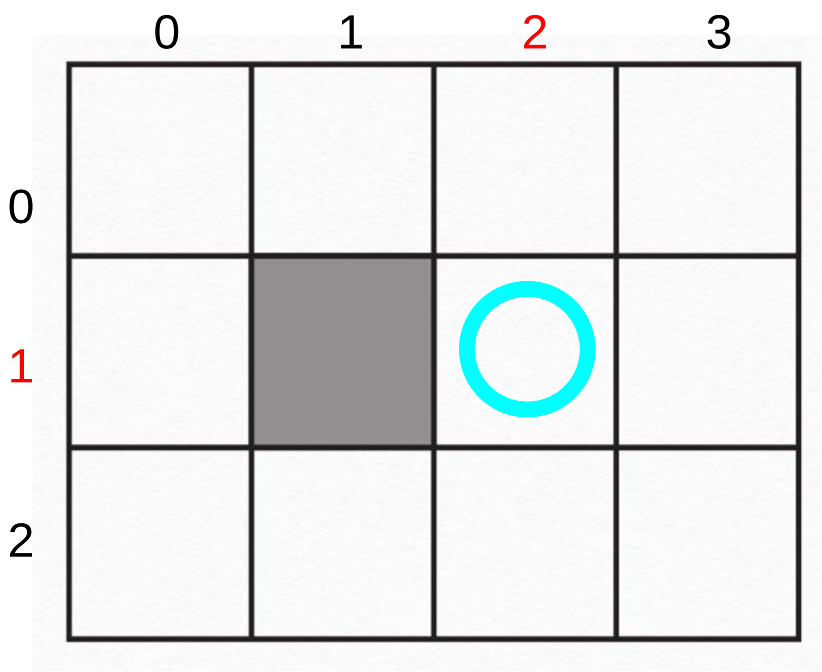
0,0,0,0

Robot Actions



- S: Stay put
- U: Go up
- D: Go down
- L: Go left (but in this case, stay put due to collision)
- R: Go right

Robot Execution



robot_a1.csv

U,L,R,R,S,U,L,U,L

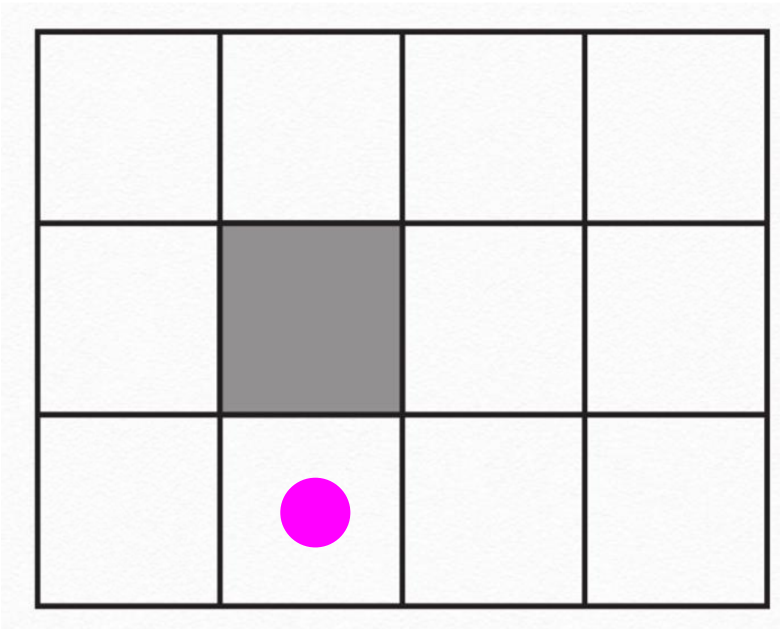
> python3 robot_execute.py

world1.csv	Gridworld
robot_a1.csv	Robot actions
1 2	Robot initial state
robot_s1.csv	Robot trajectory

Executing the sequence of actions results in a robot trajectory:

[[1,2], [0,2], [0,1], [0,2], [0,3], [0,3], [0,3], [0,2], [0,2], [0,1]]

Human Actions



- S: Stay put
- U: Go up (but in this case, stay put due to collision)
- D: Go down (but in this case, stay put due to collision)
- L: Go left
- R: Go right
- T: Toggle request status

Human Request

-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	10	-1	-1	-1
-1	-1	10	10	10	-1	-1
-1	-1	-1	10	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1

1

-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-5	-1	-1	-1
-1	-1	-5	-10	-5	-1	-1
-1	-5	-10	-10	-10	-5	-1
-1	-1	-5	-10	-5	-1	-1
-1	-1	-1	-5	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1

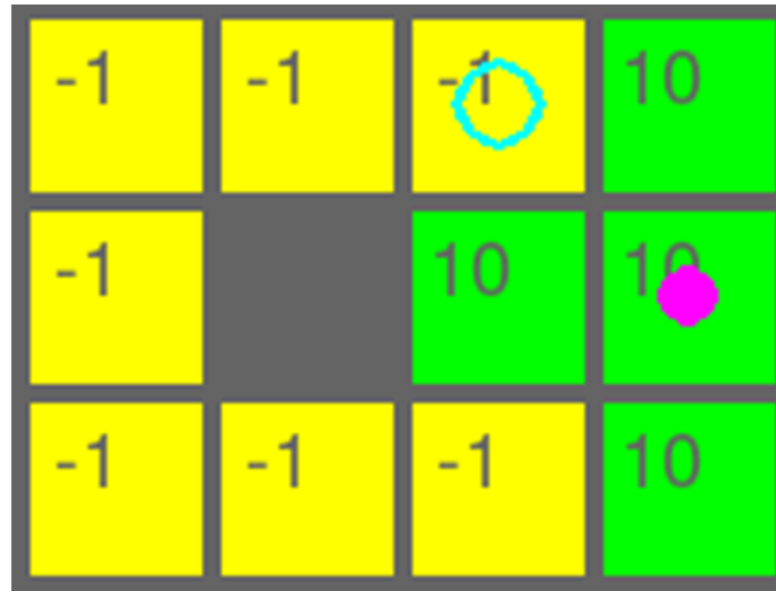
0

Why the rest of the cells is put at -1 reward instead of 0?

HRI Reward Structure

human_state [0]

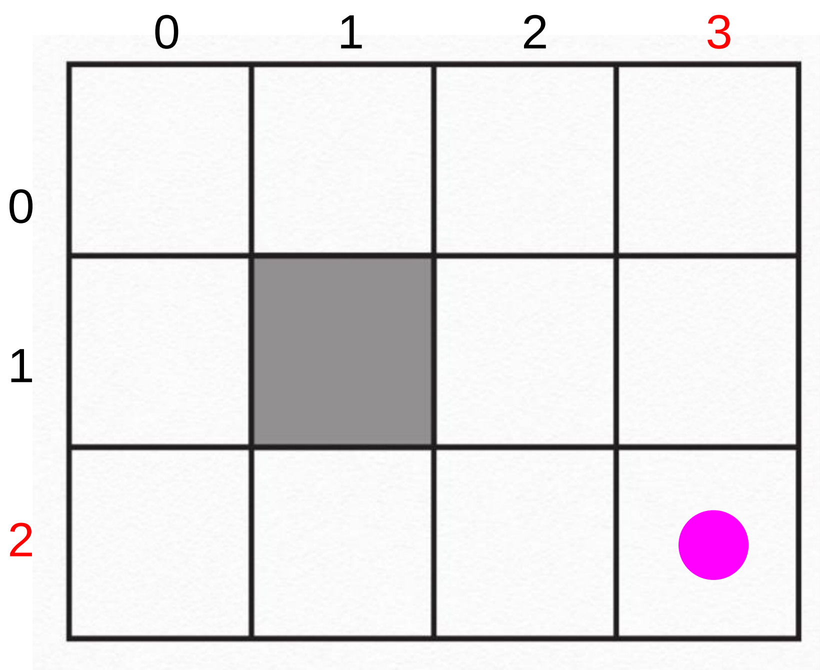
"Push" Robot away



human_state [1]

"Pull" Robot nearer

Human Execution



human_a1.csv

U,L,R,T,D,T,L

> python3 human_execute.py
world1.csv Gridworld
human_a1.csv Human actions
2 3 0 Human initial state
human_s1.csv Human trajectory

Executing the sequence of actions results in a human trajectory:

[[2,3],0],[[1,3],0],[[1,2],0],[[1,3],0],
[[1,3],1],[[2,3],1],[[2,3],0],[[2,2],0]

Setup & Preparation

1. In terminal, change directory to the day2a folder

   First and last reminder!

2. Run the skeleton code with the following command. This step is to ensure all the dependencies and packages are properly installed. E.g. python3, pip3, pygame, etc

   python3 visualizer.py world1.csv human_s1.csv robot_s1.csv

3. Install all missing dependencies and packages

   Refer to “Setup and Preparation” README file

Complete Run (Try it out!)

1. Generate a valid trajectory of a human based on the init state and a sequence of actions

python3 human_execute.py world1.csv human_a1.csv 2 3 0 human_ss1.csv

2. Generate a valid trajectory of a robot based on the init state and a sequence of actions

python3 robot_execute.py world1.csv robot_a1.csv 0 1 robot_ss1.csv

3. Visualize

python3 visualizer.py world1.csv human_ss1.csv robot_ss1.csv

***Note that you need to install all dependencies beforehand (i.e. `python3, pip3, pygame`) beforehand**

Clarification

1. **The nature of the robot's movements will be dependent on the human's request status**

2. **Rewards are not collected and factored in yet**

Human is visualized as a dot, and the robot is visualized as a circle

-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	10	-1	-1	-1
-1	-1	10	10	10	-1	-1
-1	-1	-1	10	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1

[[x,y], 1]

-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-5	-1	-1	-1
-1	-1	-5	-10	-5	-1	-1
-1	-5	-10	-10	-10	-5	-1
-1	-1	-5	-10	-5	-1	-1
-1	-1	-1	-5	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1

[[x,y], 0]

DEMO

playback

simulate system dynamics

visualization

Part B: Introduction to HRIMDP

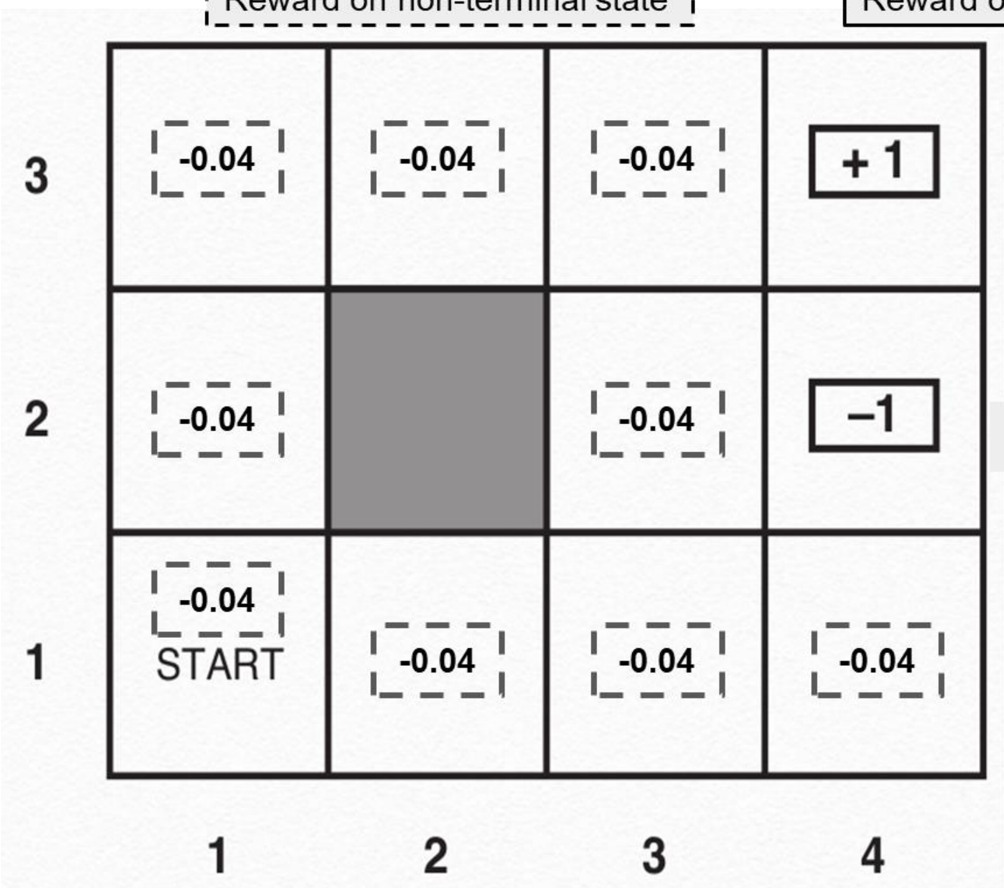
Comparison between GRIDMDP and HRIMDP

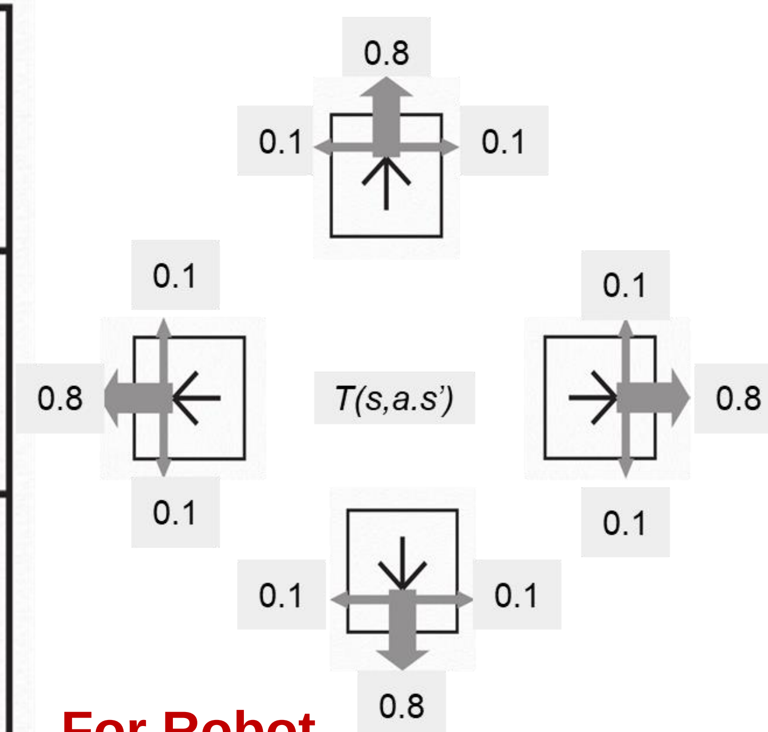
	GRIDMDP	HRIMDP
Has HUMAN	NO	YES
State Space	2D robot workspace $S_r = \{robot_x, robot_y\}$	5D joint state space $S_r \times S_h = \{(robot_x, robot_y, human_x, human_y, human_{request})\}$
Action Space	$A_r = \{'U', 'L', 'R', 'D'\}$	$A_r = \{'S', 'U', 'L', 'R', 'D'\}$
Transition	Uncertain execution	Certain execution But uncertain human state
Reward	Static 2D reward	Dynamic reward in 2D But static in 5D
Terminal States	Yes	No

Reward & Transition Structures used for GRIDMDP

Reward on non-terminal state

Reward on terminal state





For Robot

The **reward** function **R** and **transition** function **T** remain unchanged for this example unless stated.

Things to do

1. Verify installation, pytest is needed

python3 test_mdp.py

2. Try out solving MDP without terminal states

python3 test_mdp_without_terminal_states.py

3. Implement our HRIMDP with 5 dimensional gridworld
    a) Assign **reward[state]**
    b) Complete function **calculate_T**
    c) Implement function **human_execute_one_step**

4. Test implementation with

python3 test_hrimdp.py

Hints to complete test_hrimdp.py

1. Assign reward[state]

Need a set of codes to represent below reward structure

-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	10	-1	-1	-1
-1	-1	10	10	10	-1	-1
-1	-1	-1	10	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1

[[x,y], 1]

-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-5	-1	-1	-1
-1	-1	-5	-10	-5	-1	-1
-1	-5	-10	-10	-10	-5	-1
-1	-1	-5	-10	-5	-1	-1
-1	-1	-1	-5	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1

[[x,y], 0]

Hints to complete test_hrimdp.py

2. Complete function calculate_T

Assume Transition Matrix of Human Movements is:

		Next Status					
		S	U	D	L	R	T
Current Status	S	0.5	0.1	0.1	0.1	0.1	0.1
	U						
	D						
	L						
	R						
	T						

Hints to complete test_hrimdp.py

3. Complete function human_execute_one_step

- Refer to the function before this (i.e. robot_execute_one_step) for reference; both should be similar except for:
- You have to account for the following events:
  a) There is a toggle request from human, upon which, will change the toggle request status. This means if the current status is 'False', this action will change the next status to 'True'; whereas if the current status is 'True', this action will change the next status to 'False'
  b) There is NO toggle request, upon which, the toggle request status remains the same as the previous

DEMO

playback

simulate system dynamics

visualization

Instructions

This is a group project. Each group submits one zip file of all your codes/files (i.e. py) into LumiNUS at the end of the workshop

A123456_A234567_A345678_P2.zip

- Download all files in the directory **/workshops/day2** for reference codes
- Refer to the README file for instructions

THANK YOU

Email: nicholas.ho@nus.edu.sg