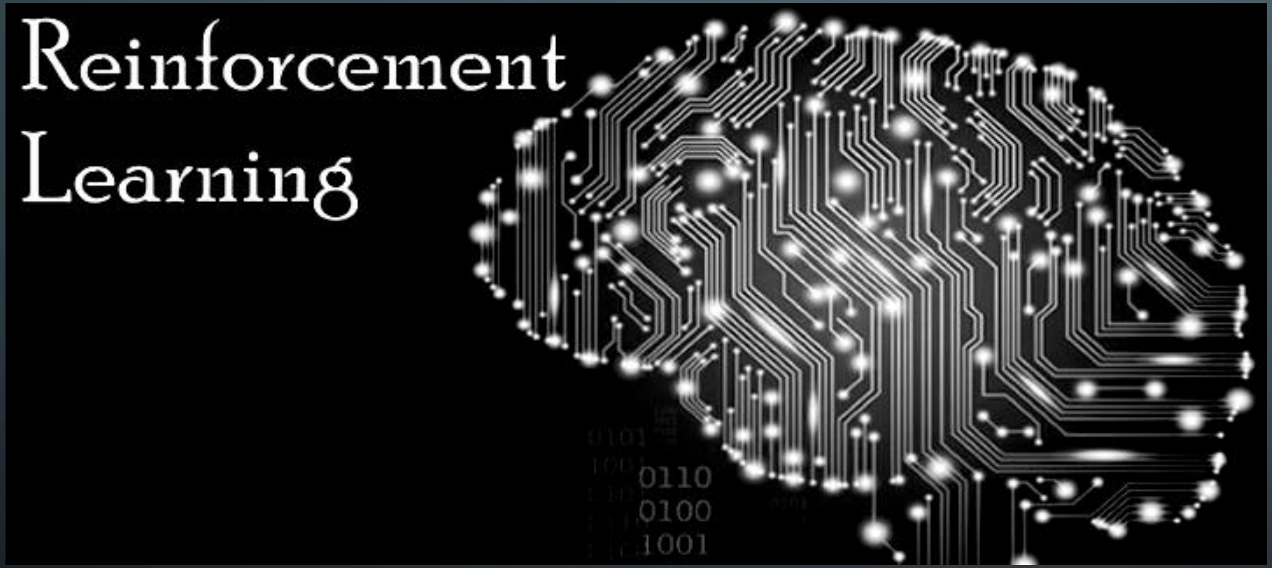
AUTONOMOUS MOBILE ROBOT NAVIGATION USING REINFORCEMENT LEARNING



Done by:
ANIIRUDH R (120012007, B.Tech Mechatronics)

Guided by:
Prof. Dr. RAMKUMAR K (Professor, SEEE)

INTRODUCTION

- Autonomous mobile robots are the robots that have autonomous decision making capabilities in order to navigate in state-space.
- However, they face lots of challenges like lack of knowledge of it’s position, imprecise sensor data and inability to replicate performance in new environment.
- To make the robot know it’s position we have SLAM techniques. To address imprecise sensor data, we have Kalman filtering but performance replication is still a major issue.
- This is where AI/ML based models greatly contribute to robotics.

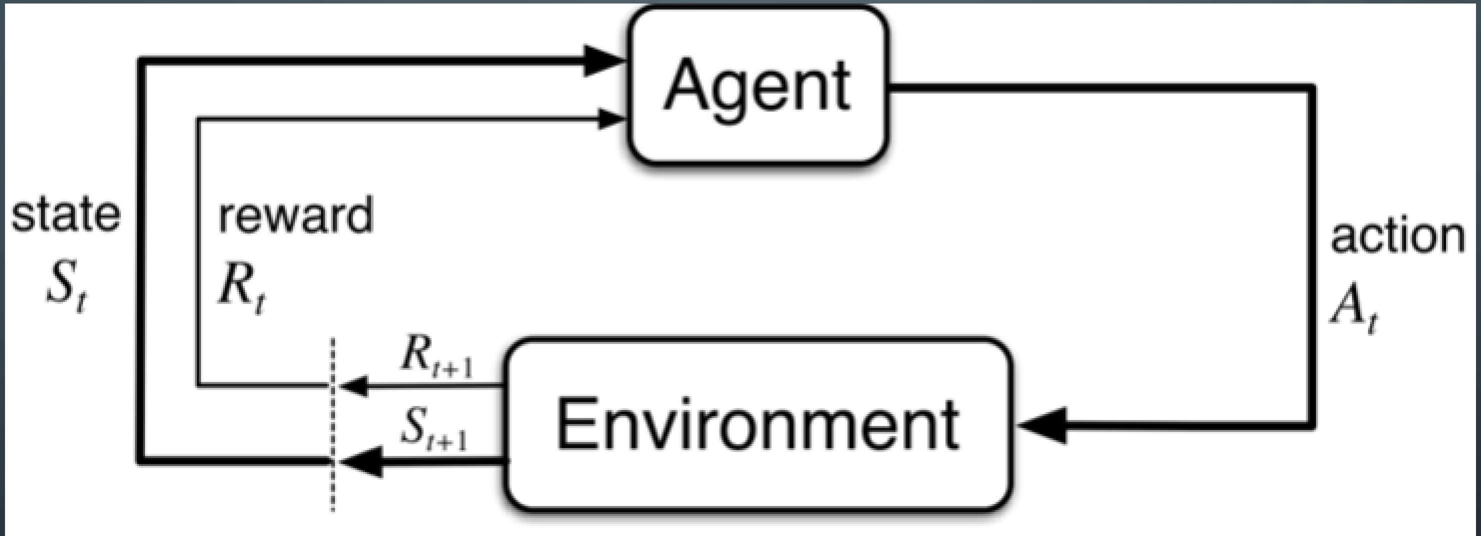
WHAT IS LEARNING AND WHY DO WE NEED IT?

- AI is an umbrella that consists of various techniques like ML, CV, NLP, Evolutionary algorithms, etc…
- Machine learning is defined as the ability of a machine to learn, without explicitly being programmed.
- Machine learning can be classified into 3 topics: Supervised Learning, Unsupervised learning and Reinforcement learning.
- In conventional control algorithms there is no scope for improvement. The control law is explicitly stated and can't adapt to new environments

REINFORCEMENT LEARNING

- Instead of explicit coding of control laws, coding the control law via RL method makes the robot to “learn from experience”.
- This is analogous to how a baby learns to walk be falling down and learning.
- A mathematical reward is assigned for every desired task (distance traversed without collision) and a punishment is assigned for every undesirable task (collision with objects)
- The robot “learns” by trying maximize its reward and minimizing punishment.
- RL is an example of a Markov Decision Process.

RL BLOCK DIAGRAM



- State is given by the sensors, while action (forward, reverse) is performed by actuator.

RL TYPES

- RL has 3 types:Value based(Q learnig), Policy based(Actor Critic) & model based.
- Value based algorithms are optimized to get maximum value (Q value).

$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \mid S_t = s \right]$$

Expected Reward discounted Given that state

- Policy based iterations are optimized to get the best possible policy (control law) based on probability distribution of next state.

Stochastic policy: $\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]$

- Model based approaches require knowledge of existing environment and try to optimize it.
- Exploration vs Exploitaion.

WORK DONE IN THE LAST MONTH

- Identifying the problem statement and planning a sequential set of actions in order to converge to a solution.
- Thorough literature survey, understanding previous research papers and going through the Reinforcement learning textbook by Sutton and Barto.
- Understanding the mathematics and working of Q-Learning algorithm.
- Converting Q-Learning Pseudocode to MATLAB code.
- Debugging the code and evaluating its performance in MATLAB.
- Visualizing the performance and result of the Q-learning algorithm in MATLAB by implementing a 2D simulation framework.

Q- LEARNING ALGORITHM PSEUDOCODE

Estimate the start state

While state != terminal (exit of the maze)

Choose an action based on e-greedy method (exploration and exploitation approach)

Execute the action (forward, right or left by enabling both motors, left motor or right motor, respectively)

Measure output voltage of the proximity sensors

Update the state vector and its type (health/subhealth)

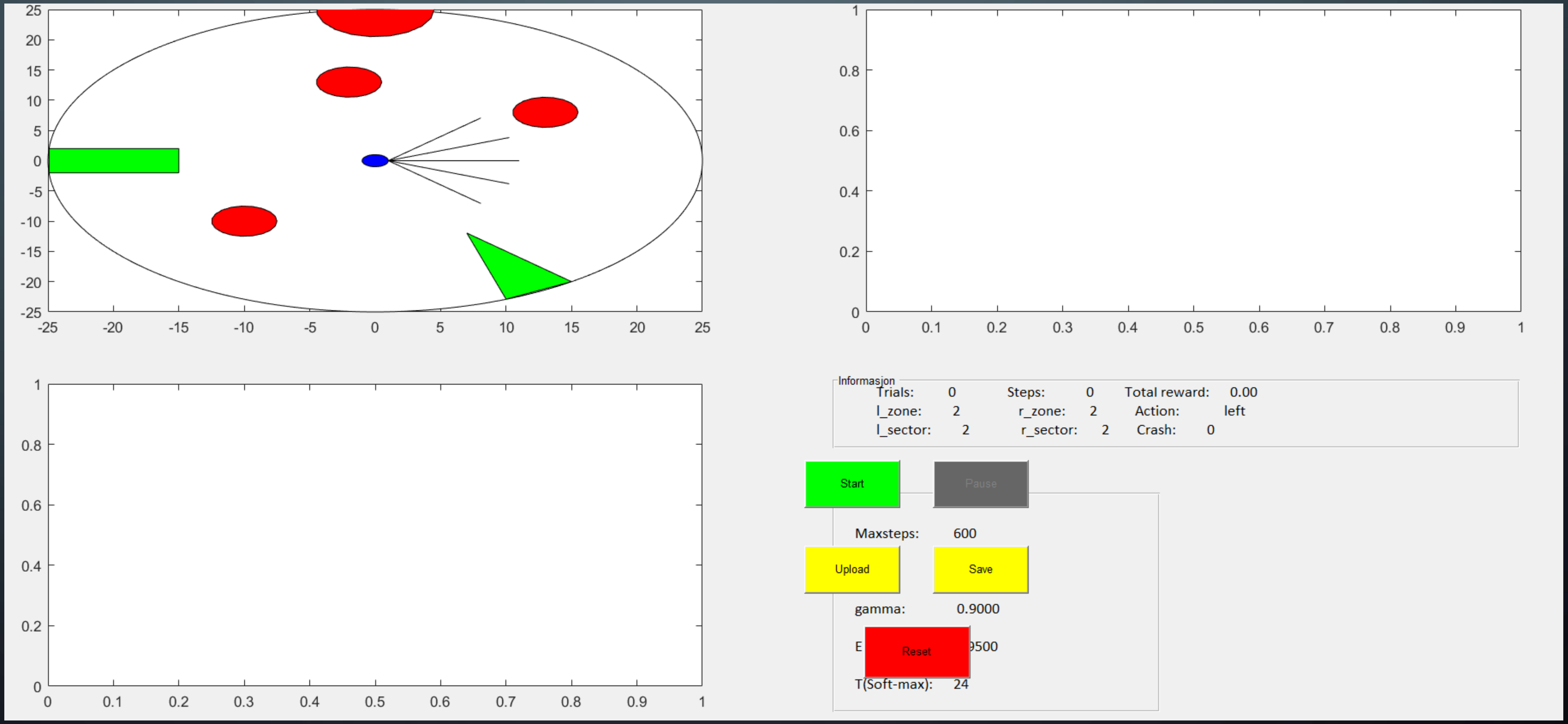
Check if there is an emergency condition (any of the sensors has a distance shorter than the minimum allowed for subhealth state)

Obtained rewards based on prior state and executed action

Update the action value function Q

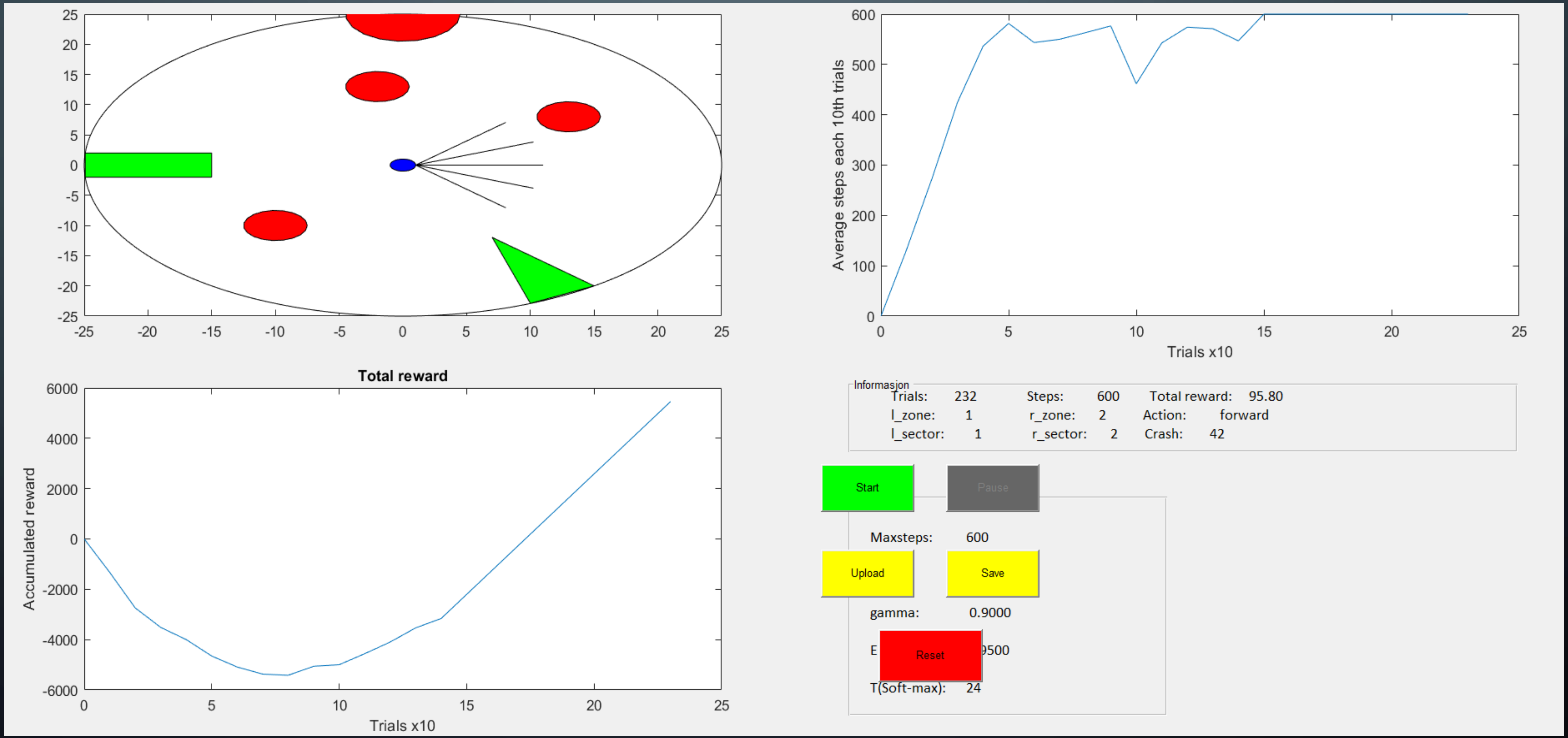
2D SIMULATION ANALYSIS

BEFORE:



2D SIMULATION ANALYSIS

AFTER:

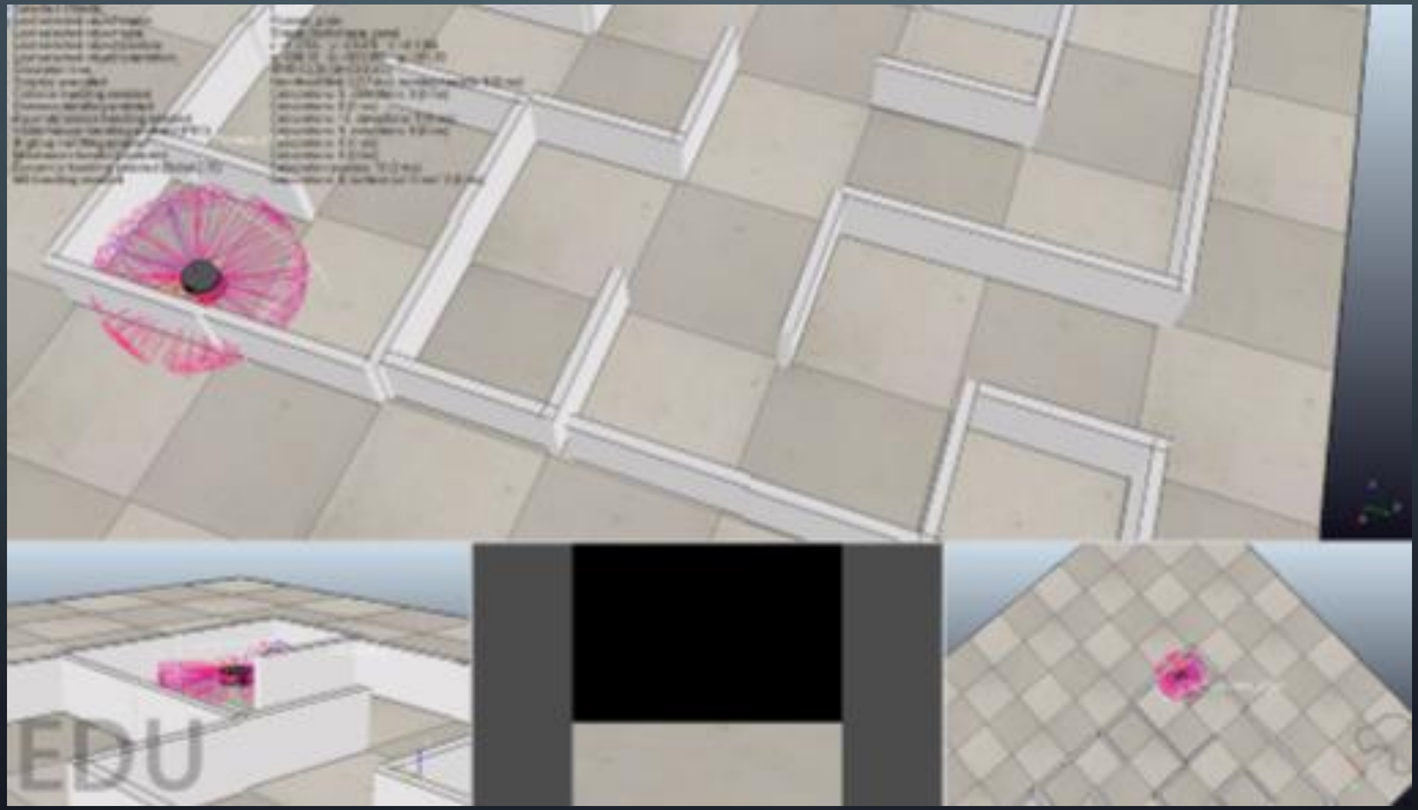


PLANNED WORK OVER THE NEXT 2 MONTHS

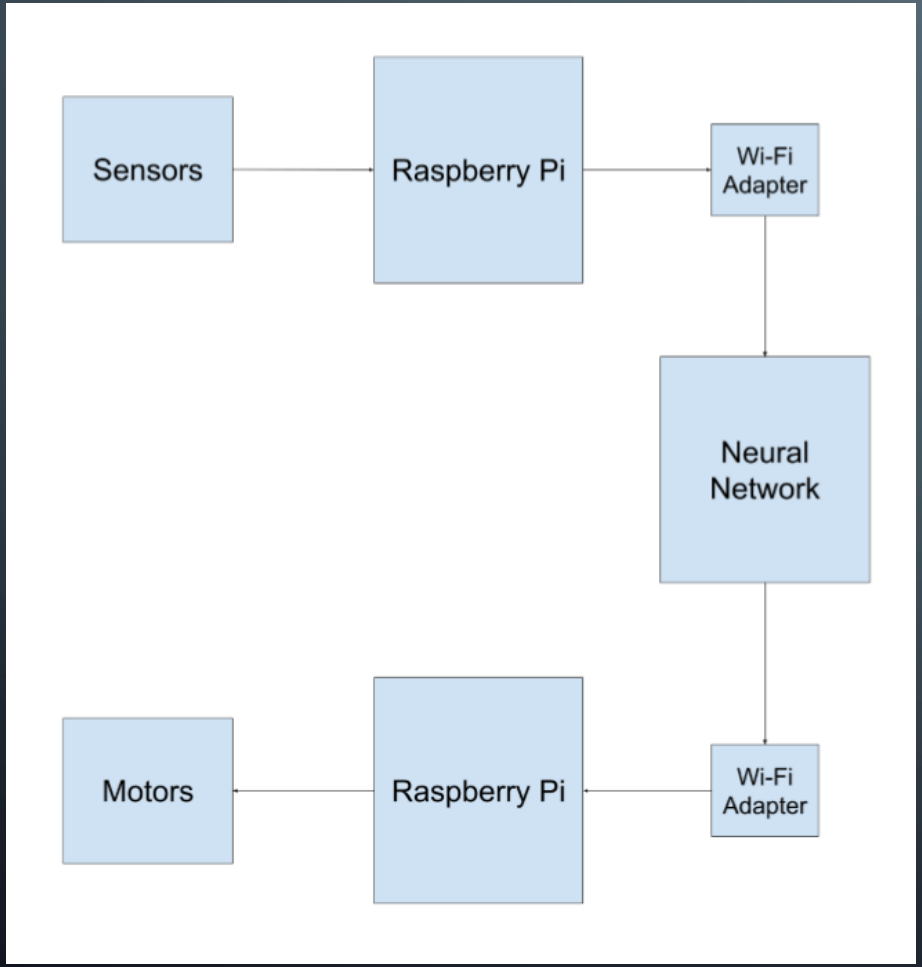
- 2D simulations are used to analyze and visualize the performance of the algorithm. We need 3D simulations in order to replicate the real-world performance of the algorithm.
- 3D Simulation using Virtual Robot Experimentation Platform (V-REP) by interfacing MATLAB.
- Testing the “learning ability” by simulating in a new environment.
- Explore different algorithms like Actor Critic, SARSA & compare performance.
- After 3D simulation, implementing this algorithm in a physical robot to measure real-world performance by using Rpi/Arduino for motor control and interfacing it with MATLAB and V-REP.
- Finally, documenting all the work done in the form of a research paper.

V-REP ENVIRONMENT

- Allows interfacing of MATLAB, RPi and Arduino and replicates real-world performance. Artificial environments can be created in it and tested.



BLOCK DIAGRAM FOR PHYSICAL IMPLEMENTATION



- RL is computationally intensive, so an on board Microcontroller cant compute the algorithm.
- So, we'll be interfacing RPi/Arduino with MATLAB.
- MATLAB in the laptop will run the RL algorithm and send control signals to RPi/Arduino, which will act exclusively as a motor controller.

Thank you