Reinforcement Learning for Mobile Robot Position Control

Harsh Goyal and Rajneesh Singh

Research advisor: Dr Vineeth B.S.

Indian Institute of Space Science and Technology
Thiruvananthapuram, Kerala, India 695547

April 29, 2024

Outline

Introduction

- **Motivation**
  - Robotics has been very popular for the last decade.
  - Robot has to navigate in different environments by controlling its own position.
  - Traditional control systems are proving to be inadequate for the complex tasks.
  - Led to thinking of alternative algorithms.
  - Reinforcement Learning allows the agent to learn an optimal policy by interacting directly with the environment.
- **Problem**
  - Position control of wheeled mobile robots.
  - Move a robot from its current position to a given destination point avoiding various obstacles in an unknown environment using Reinforcement Learning.

Introduction

S.No.	Paper	Summary
1	Reinforcement Learning for Position Control Problem of a Mobile Robot[1]	Exploring reinforcement learning techniques for the position control of a mobile robot
2	Deep reinforcement learning based mobile robot navigation: A review[2]	used Deep Reinforcement Learning to solve the problem for the navigation of mobile robots

Table 1: List of Papers and Summaries

[1] Gonzalo Farias et al. "Reinforcement Learning for Position Control Problem of a Mobile Robot". In: *IEEE Access* 8 (2020), pp. 152941–152951. DOI: 10.1109/ACCESS.2020.3018026.

[2] Kai Zhu and Tao Zhang. "Deep reinforcement learning based mobile robot navigation: A review". In: *Tsinghua Science and Technology* 26.5 (2021), pp. 674–691. DOI: 10.26599/TST.2021.9010012.

Environment

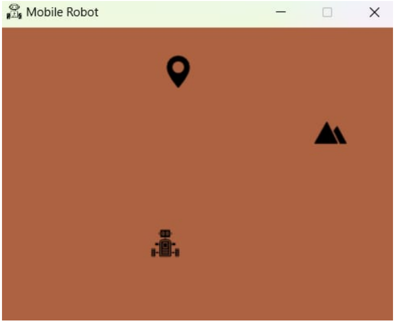


Figure 1: Environment

- The environment includes all possible locations for the robot, its destination, and obstacles.
- Action space consists of four actions: right, down, left, up
- Robot receives a reward of $+2000$ for reaching the destination, incurs a cost of 1000 for hitting obstacles, and is rewarded with the negative distance from the destination for all other cases.
- State: [[destination distance, destination angle], [nearest obstacle distance, nearest obstacle angle]]

Performance Criterion

- Compared the steps taken by the robot to reach the destination.
- Total steps taken by an untrained robot vs our robot which has been trained using Q-learning.
- This criterion suits the project's main goal: designing a reinforcement learning model for a robot to efficiently reach its destination while navigating around obstacles in minimal steps.

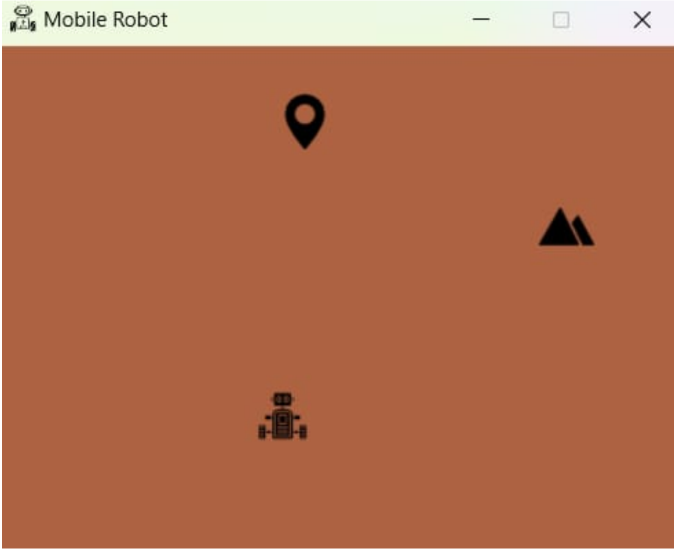
Reinforcement Learning Agent

- Adopted the same model of the agent and the environment as described in the paper.
- Introduced more than one obstacle on the way of the robot.
- Trained our agent for 10 million episodes.
- Q-learning has been used in the RL implementation.
- After the robot has been trained using this policy, we have finally tested the robot using a behaviour policy which focuses on exploitation.

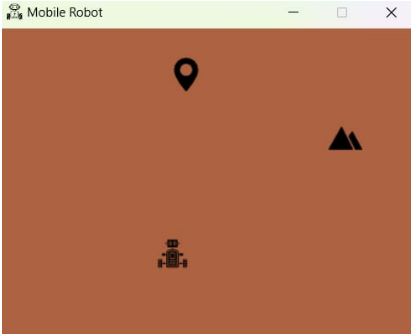
$$Q^*(s_t, a_t) = r(s_t, a_t) + \gamma \max_a Q^*(s_{t+1}, a) \tag{1}$$

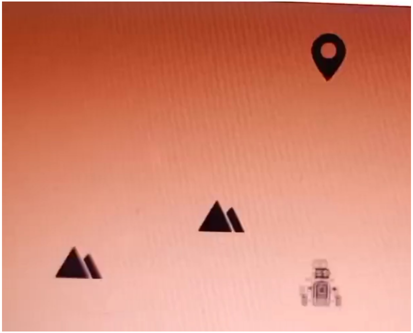
$$\begin{aligned} Q^{(i+1)}(s_t, a_t) = {} & (1-\alpha) Q^{(i)}(s_t, a_t) \\ & + \alpha \left(r(s_{t+1}, a_t) + \gamma \max_a Q^{(i)}(s_{t+1}, a) \right) \end{aligned}$$

Untrained Robot



Trained Robot





Extensions

1. **Introduction**
   - Various approaches exist for solving the position control problem of mobile robots.
   - Machine learning techniques such as neural networks offer efficient solutions.
2. **Neural Networks for Position Control**
   - Utilizing neural networks with 8 neurons for efficient task execution.
   - PyTorch and TensorFlow can be employed for implementation.
3. **Other Considerations**
   - Exploring different methodologies including Temporal Difference and Function Approximation.
   - Supervised learning showcases promise, especially when the environment os obstacle free.

Extensions

1. **Temporal Difference (TD)**
   - Application of Q-learning with TD(0) and offline policy improvement.
   - Computational time constraints observed with TD algorithms.
2. **Function Approximation**
   - Effective for large state spaces, focusing on basis function selection.
   - Gradient descent aids in optimization for mobile robot tasks.
3. **Non-RL Based Control Algorithms**
   - Non RL control laws outperforming RL in certain scenarios.
   - Highlighting the challenge of learning from experience in non-RL methods.

Performance Performance analysis and comparison

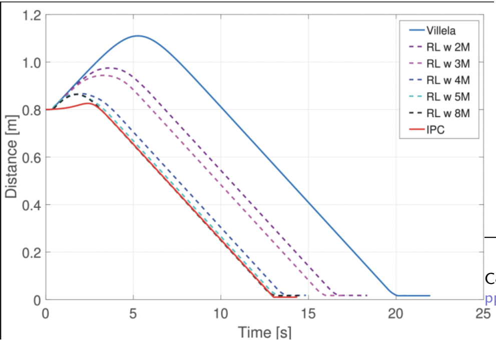


Figure 2: Distance to the target vs. time for each control algorithm (Villela, IPC, and RLw)

- Figure from[a]
- IPC algorithm performs the best when no obstacle has been introduced into the environment

[a] Gonzalo Farias et al. "Reinforcement Learning for Position Control Problem of a Mobile Robot". In: *IEEE Access* 8 (2020), pp. 152941–152951. doi: 10.1109/ACCESS.2020.3018026.

Performance Performance analysis and comparison

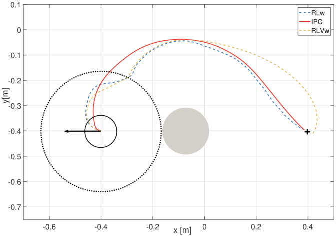


Figure 3: Performance of the algorithms when an obstacle has been introduced

- Figure from[a]
- IRL algorithm performs even better than the IPC algorithm

[a] Gonzalo Farias et al. "Reinforcement Learning for Position Control Problem of a Mobile Robot". In: *IEEE Access* 8 (2020), pp. 152941–152951. DOI: 10.1109/ACCESS.2020.3018026.

Performance Performance analysis and comparison

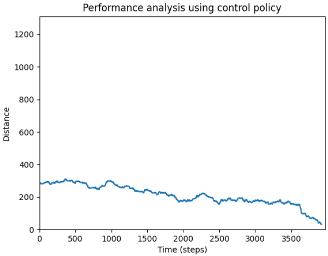


Figure 4: Performance of the algorithms when an obstacle has been introduced

- Performance of our agent
- robot first tackles the obstacle so its distance from the destination slightly increases and then it reaches the destination in such a way that the distance in each step is minimised.

Performance Performance analysis and comparison

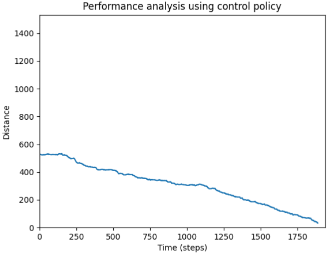


Figure 5: Performance analysis when there is no obstacle

- Performance of our agent
- robot reaches the destination in such a way that the distance in each step is minimised

Performance Performance analysis and comparison

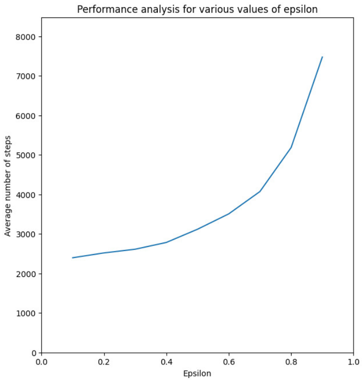
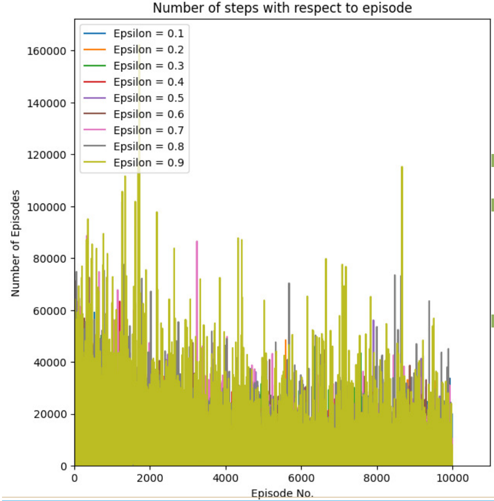


Figure 6: Performance analysis for various values of epsilon

- Performance of our agent
- as the value of epsilon increases the average number of steps also increases
- agent takes more steps to reach its destination point as epsilon increases

Performance Performance analysis and comparison



- Performance of our agent
- as the episode no. increases the steps required by a mobile robot to reach the destination decreases
- maximum for epsilon = 0.9 and minimum for epsilon = 0.1

Challenges, ideas and Future Works

- system model refinement
- enhanced reward design
- transfer learning and generalization
- computational complexity and efficiency
- Future work may include applying RL to various control tasks for mobile robots, like navigating around bigger obstacles, following specific paths accurately, coordinating groups of robots in formations, and reaching agreements between multiple robots. RL based algorithms are also useful in space applications such as rovers to explore unknown terrain of any planet.

Questions?

rajneesh.sc21b111@ug.iist.ac.in

Thank you.