Multi-Goal Navigation of a Mobile Robot Using Hierarchical Reinforcement Learning

Marco António Gomes Silva

Integrated Masters in Computer and Telematics Engineering
Supervisor: Prof. Dr. Filipe Silva
July 2021

Universidade de Aveiro

Covered Topics

1. **Introduction and Problem**
2. Proposed approach and Objectives
3. Low-Level Exploratory Behaviour
4. High-Level Implementation
5. Multi-Goal and Dynamic Behaviour
6. Final Remarks and Future Work

Problem

- Endowing a mobile robot with abilities so that it can safely navigate among multiple goals in maze-like environments.

Mobile robot navigation is complex:

- Where am I?
- Where am I going?
- How should I get there?

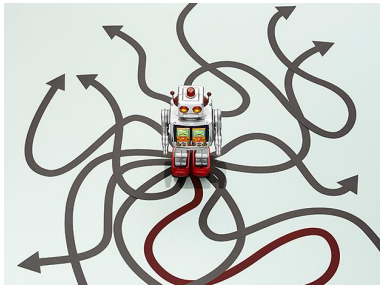
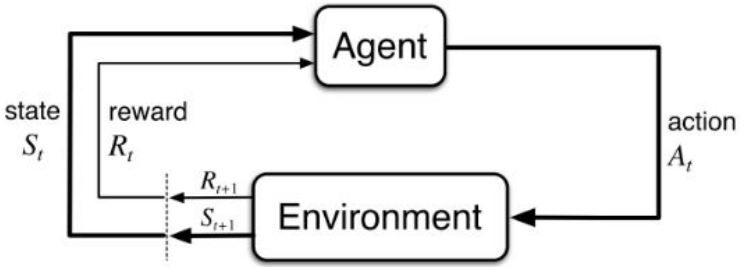


Photo by Dan Saelinger

Reinforcement learning challenges:

- Curse of dimensionality.
- Reward definition.
- Generalization and abstraction.



Covered Topics

1. Introduction and Problem
2. **Proposed approach and Objectives**
3. Low-Level Exploratory Behaviour
4. High-Level Implementation
5. Multi-Goal and Dynamic Behaviour
6. Final Remarks and Future Work

Proposed Approach

Hierarchical Reinforcement Learning (Yannis Flet-Berliac, 2019)*

The main task is decomposed into a hierarchy of sub-tasks where policies at the top of the hierarchy call upon policies from lower levels.

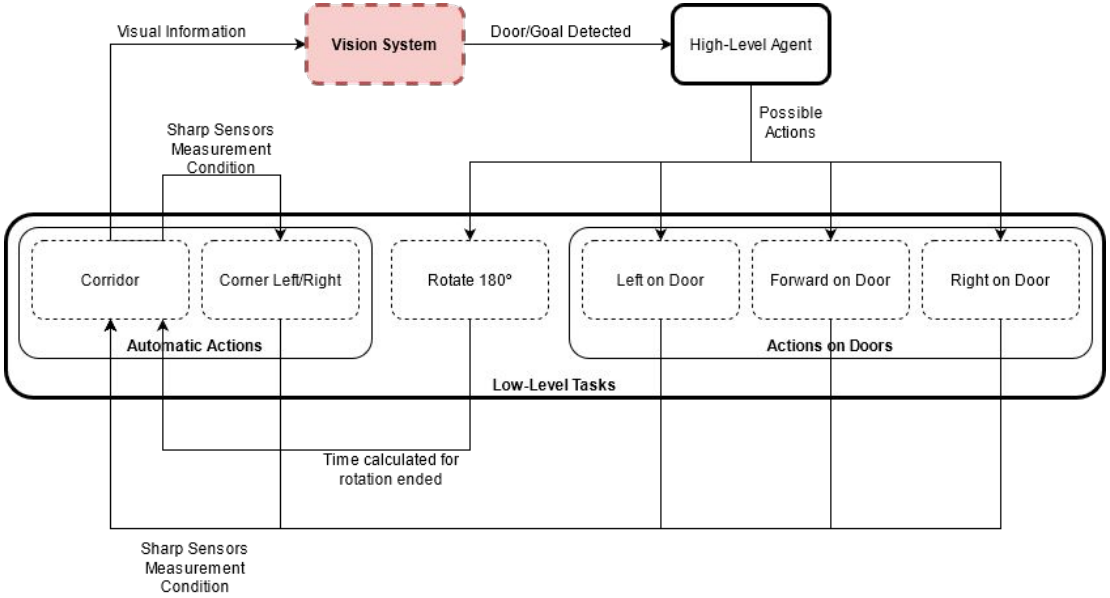
Advantages over classic Reinforcement Learning:

- Transforms one complex problem into multiple simpler problems.
- Reduces computational requirements and complexity.
- Minimizes the state-space to just the necessary in each sub-task.
- Allows the reutilization of sub-tasks.
- Shortens the necessary learning time.

*Yannis Flet-Berliac. The promise of hierarchical reinforcement learning. The Gradient, 2019.

Proposed Approach

Hierarchical Structure



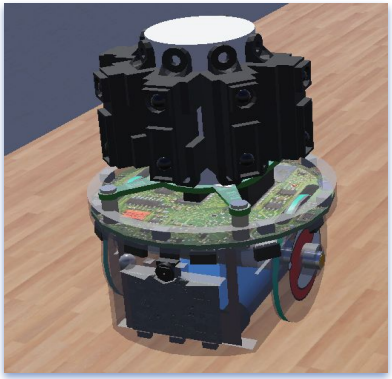
- **Low-Level**: Composes the actions necessary to explore the environment
- **High-Level**: Topological navigation between doors
- **Vision System**: Detect doors and locations in the environment (Not implemented and emulated in simulation)

Objectives

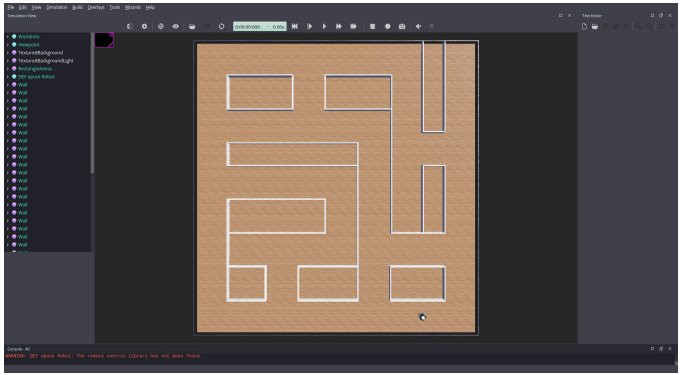
Three main objectives to achieve:

1. Develop elementary functionalities to endow the robot with a safe exploratory behaviour in maze-like environments composed of corridors, 90-degree corners and T-shaped 3-door junctions.
2. Conceive and implement a hierarchical structure to solve the problem of navigating in a maze-like environment resorting to its topological representation.
3. Evaluate the proposed approach in two scenarios: autonomous navigation among multiple goals and in dynamic environments.

Robot and Simulation Environment



E-puck with 6 extra sensors



Webots Robot Simulator by Cyberbotics Lda.

- Mandatory for reward shaping and lower-level tasks training (corridor) and transformation (corners and doors).
- Accurately simulates physical properties of objects such as velocity, friction and inertia.

Covered Topics

1. Introduction and Problem
2. Proposed approach and Objectives
3. **Low-Level Exploratory Behaviour**
4. High-Level Implementation
5. Multi-Goal and Dynamic Behaviour
6. Final Remarks and Future Work

Low-Level Exploratory Behaviour

State-action Discretization

- Six sensor's discretization levels with six sensors:
  - Total of 6^6 = 46 656 states.
  - Almost 43 000 states are left unfilled (≈92%)
- Nine possible actions in each state.

Distance (cm)	Level
$0 \leq d < 6$	1
$6 \leq d < 10$	2
$10 \leq d < 15$	3
$15 \leq d < 20$	4
$20 \leq d < 25$	5
$d \geq 25$	6

Sensor discretization levels

Action	Speed (Left, Right) (rad/s)
Front (F)	(2,2)
Light Left (LL)	(1.5,2)
Light Right (LR)	(2,1.5)
Mid Left (ML)	(1,2)
Mid Right (MR)	(2,1)
Left (L)	(0,2)
Right (R)	(2,0)
Hard Left (HL)	(-2,2)
Hard Right (HR)	(2,-2)

Actions discretization

Low-Level Tasks

Move on corridor

- Simplest task of all.
- The robot must move along a corridor aligned with the center.

Corner left/right

- Turn left and right are symmetric.
- Optimal path is hard to define for these situations.

Doors

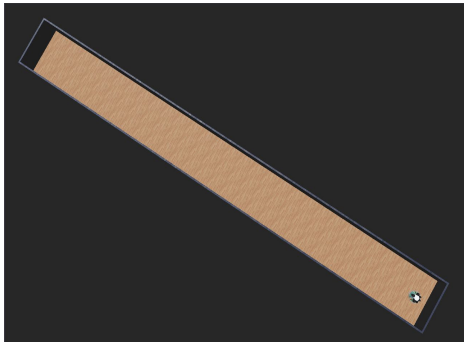
- Three possible actions: turn left, go forward or turn right.
- Optimal path is hard to define here as well.

Reward Shaping - Corridor

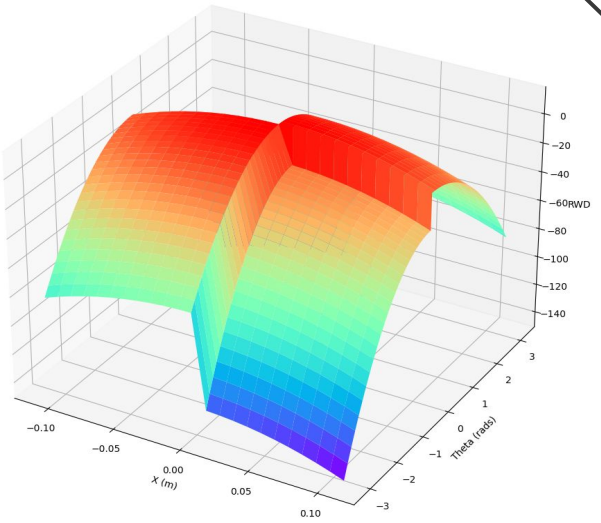
- Intermediate rewards guide the learning process to a good solution (close to the desired behaviour).
- Depending on the task, this reward shaping will consider different features such as the robot orientation or its distance to the walls.
- Performed offline.
- A reward is calculated for each achievable state in the table.
- 5 versions until obtaining a good reward function (one that performed the task).

Move on corridor

- Approximately 17500h in the simulator (±215 real hours).
- Placed in random positions during training to improve convergence.
- R-Learning algorithm used in learning (S Mahadevan, 1996)*



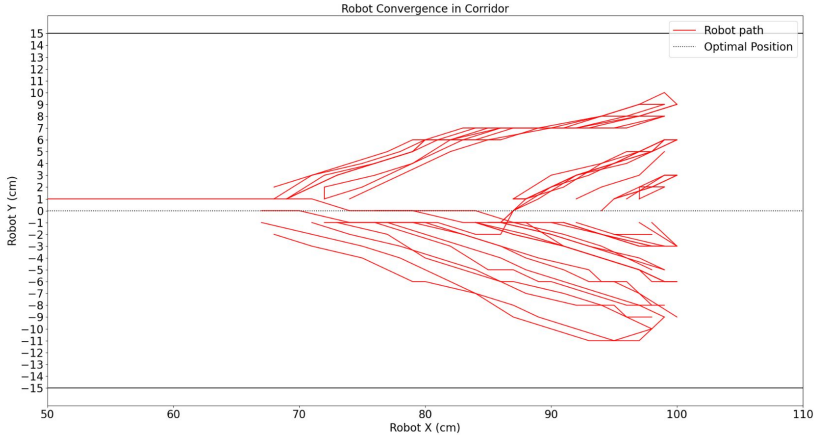
Maze



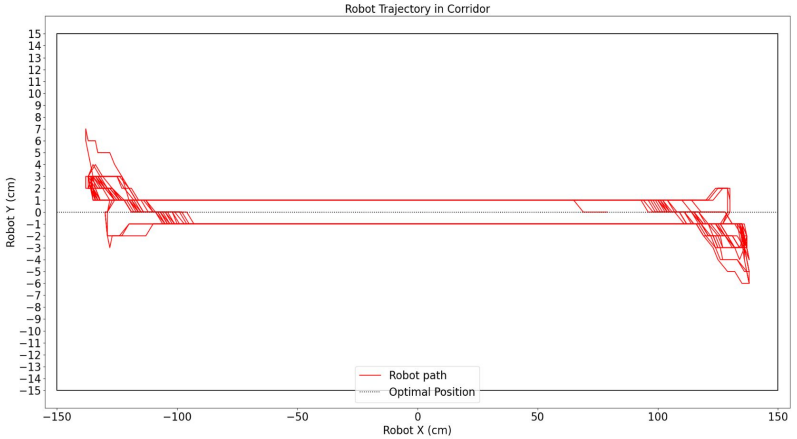
Reward

*S Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. Machine Learning, 22(1-3):159–195, 1996

Move on corridor - Results



Convergence to center of corridor



Trajectory in corridor

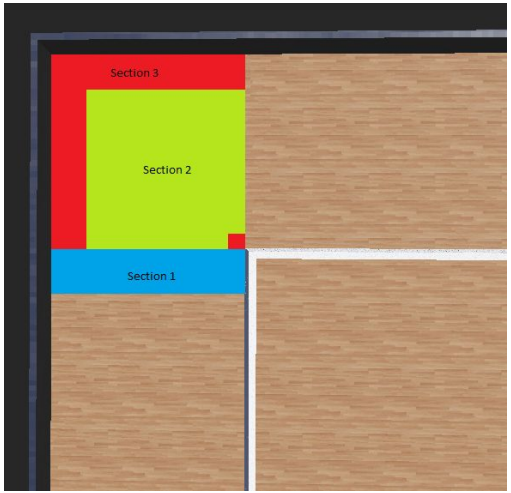
Corner left/right

First approach:

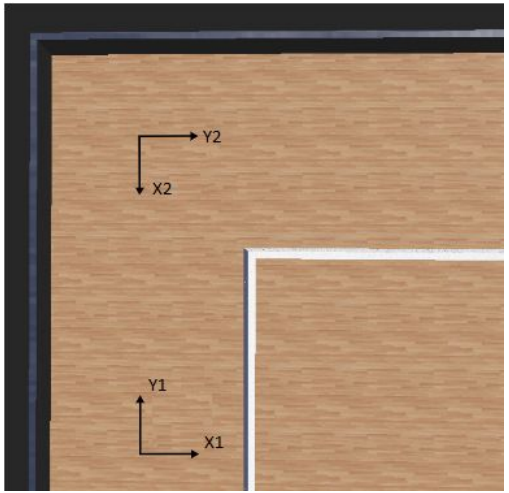
- Divided in three sections with different rewards.
- Robot trained multiple times with different rewards.

Second approach:

- Re-used corridor learning to perform corner to right.
- Applied symmetry to get opposite corner.



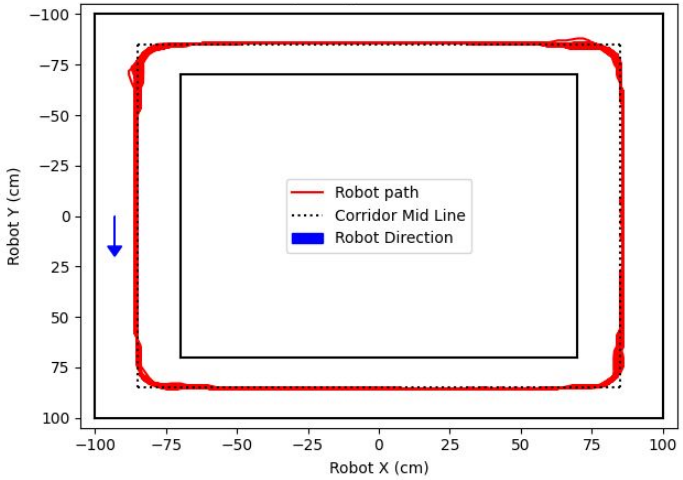
Corner Sections



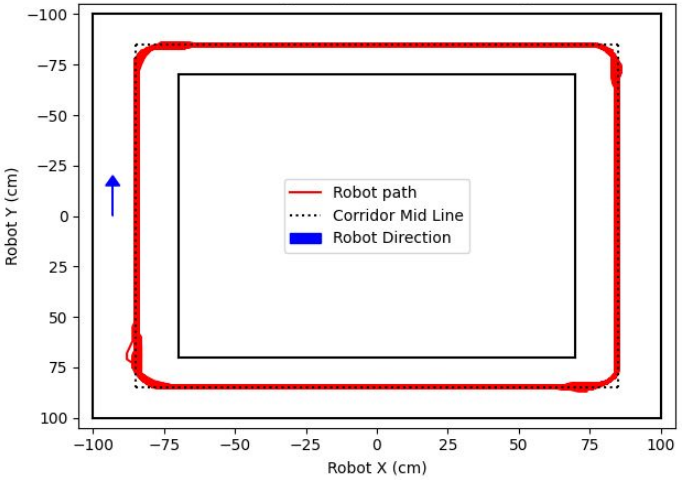
Corridor to corner axis transformation

Corner left/right - Results

- 69 laps complete in each direction.
- 1 lap = 4 corners
  - 4X69 = 276 corners complete

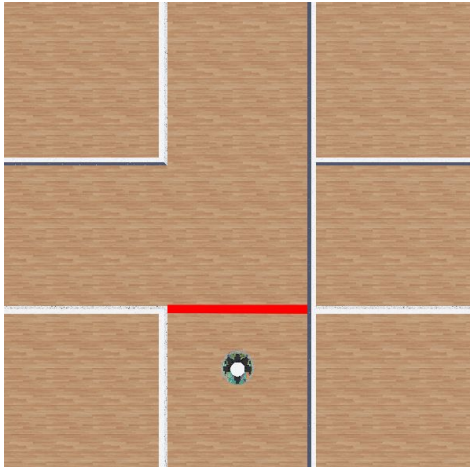


Robot trajectory to the left

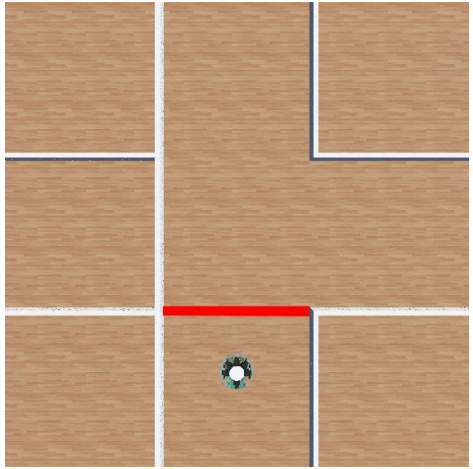


Robot trajectory to the right

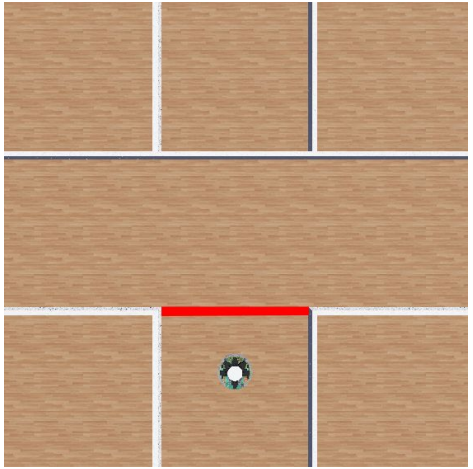
Doors (T-Junctions)



Front-Left Door
(1)

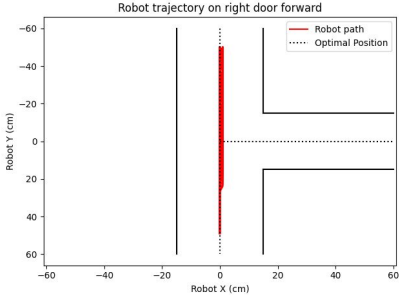


Front-Right Door
(2)

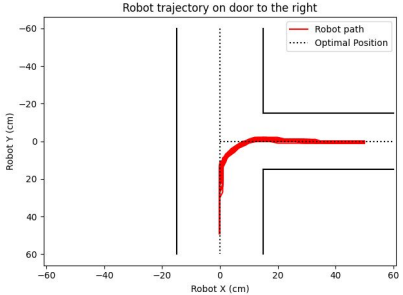


Left-Right Door
(3)

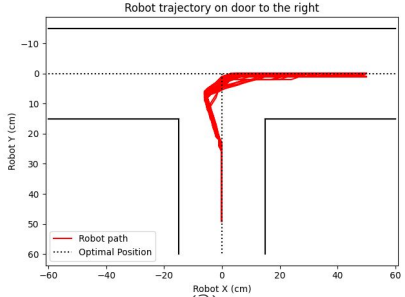
Doors - Results



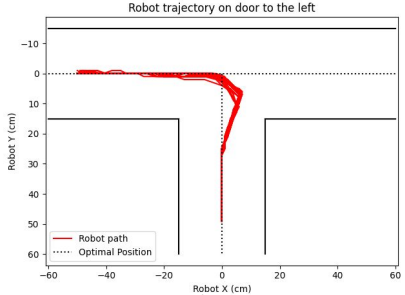
(1)



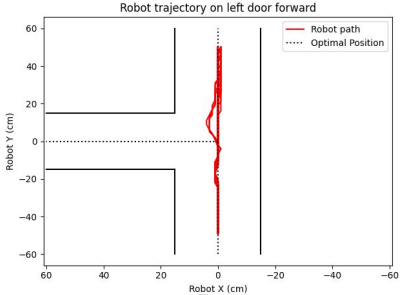
(2)



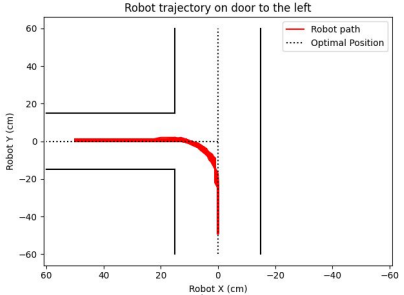
(3)



(4)



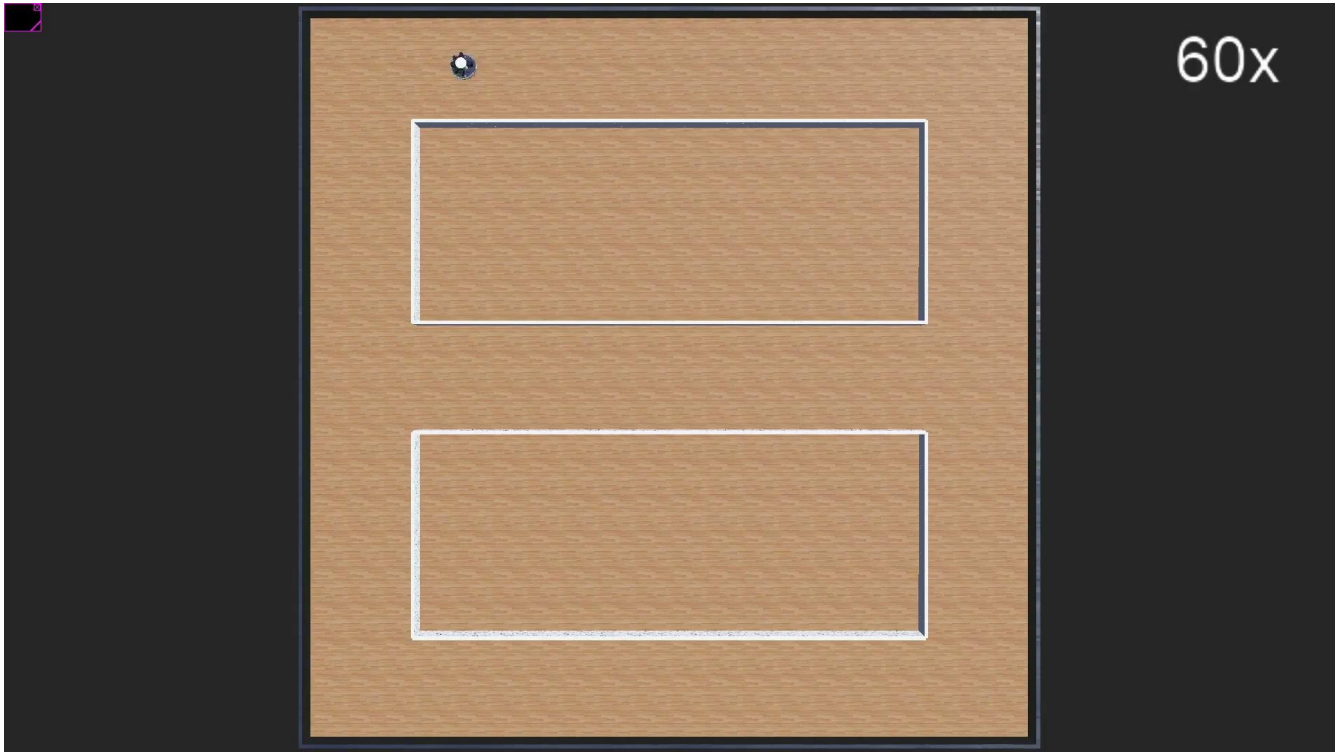
(5)



(6)

Robot Exploratory behaviour Demo

- **Only 3 tables used:**
  - Go Forward
    - Corridors and doors
  - Turn Left
    - Corners and doors
  - Turn Right
    - Corners and doors
- Transitions between tables using robot sensorial system

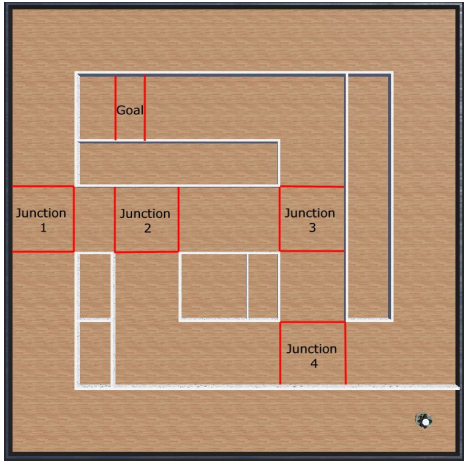


Covered Topics

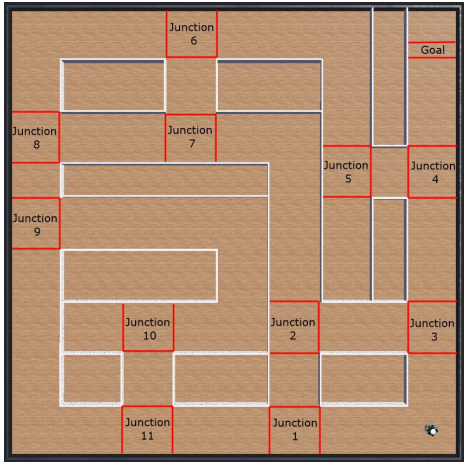
1. Introduction and Problem
2. Proposed approach and Objectives
3. Low-Level Exploratory Behaviour
4. **High-Level Implementation**
5. Multi-Goal and Dynamic Behaviour
6. Final Remarks and Future Work

High-Level

- Each door in T-junction is a node in the topological representation of the environment.
- Decides the best action when the robot reaches a door in a T-Junction.
- Evaluated in two mazes:
  - **Maze 1**: 4 T-Junctions, 12 doors/states.
  - **Maze 2**: 11 T-Junctions, 33 doors/states.



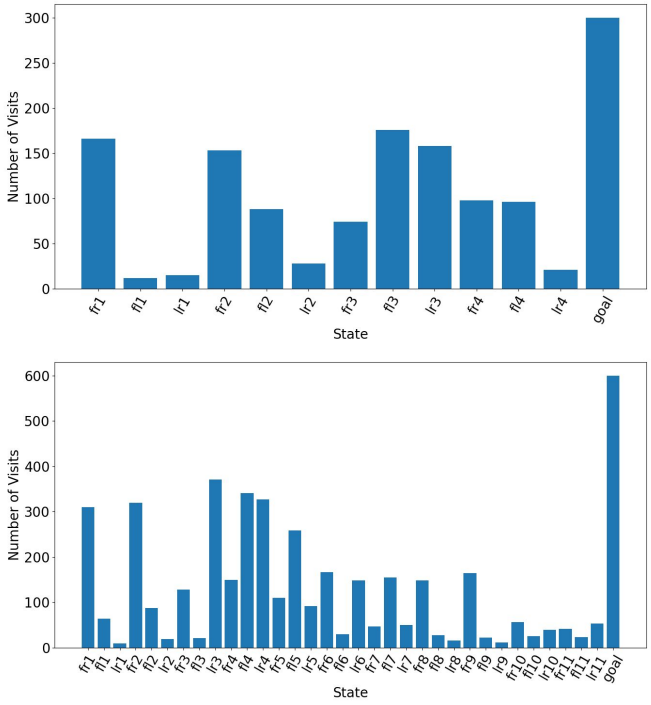
Maze 1



Maze 2

High-Level Tasks - Implementation

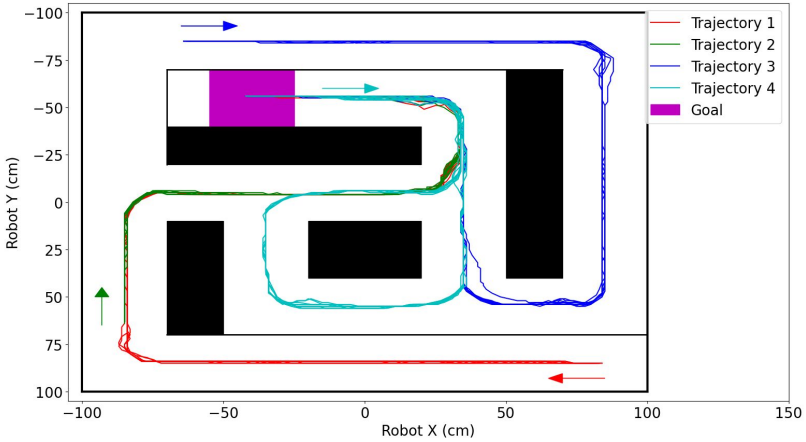
- Q-Learning algorithm used in the learning (C. H. Watkins, 1989)*
  - Maze 1: 300 episodes.
  - Maze 2: 600 episodes.
  - Four starting position.
- Reward based on time between high-level states.
- All states are visited, whereby the whole maze is explored.



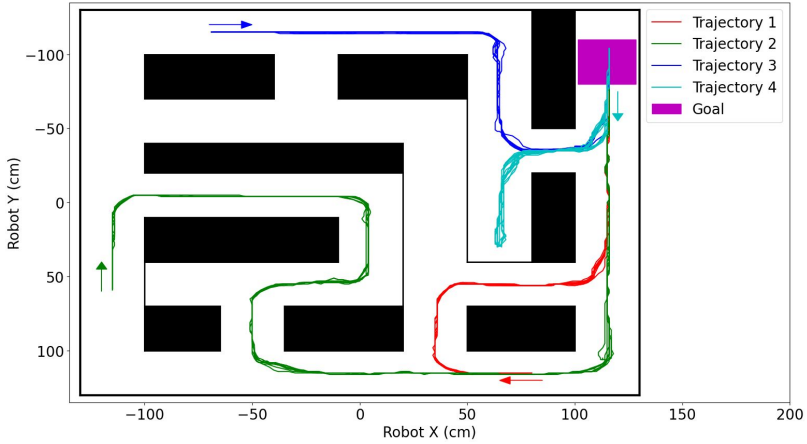
Visited States Maze 1 (top) and Maze 2 (bottom)

*C. J. C. H. Watkins. Learning from Delayed Rewards. PhD thesis, King's College, Oxford, 1989

High-Level Tasks - Results



10 superpositioned runs for each starting position in Maze 1



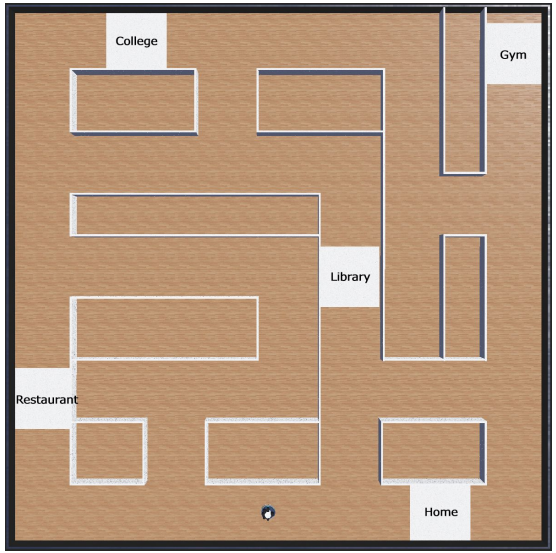
10 superpositioned runs for each starting position in Maze 2

Covered Topics

1. Introduction and Problem
2. Proposed approach and Objectives
3. Low-Level Exploratory Behaviour
4. High-Level Implementation
5. **Multi-Goal and Dynamic Behaviour**
6. Final Remarks and Future Work

Multi-Goal

- Five locations are defined in Maze 2 (white squares) and the robot must be able to navigate between them.

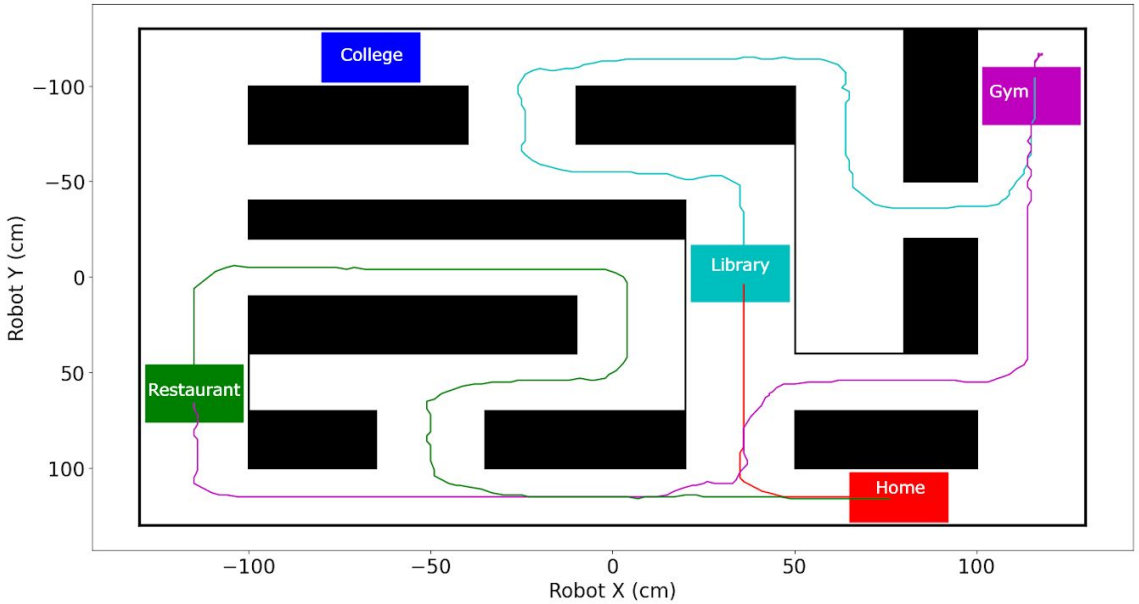


Learning algorithm:

1. Define multiple starting positions for the robot (other goals preferably) to explore the whole maze.
2. Perform the learning as in a single-goal problem.
3. While exploring, save the states and decisions until reaching a goal.
4. When reaches a goal save that information as a path between the two goals.
5. In the end, a topological graph connects all the goals in the environment.

Multi-Goal - First Results

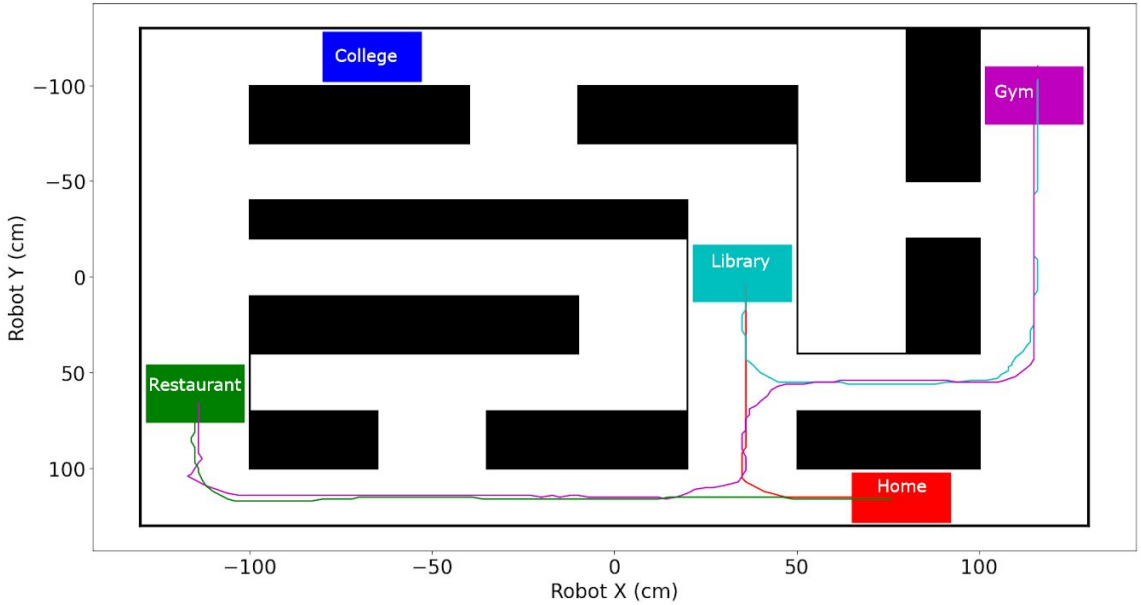
- The robot does not always take the best path.
- From *Restaurant* to *Home* would be better to turn around.
- Extra Lower-Level Task developed.
- Robot becomes able to rotate 180° and go in the reverse direction.



Trajectory between Multiple Goals:
Home -> Library -> Gym -> Restaurant -> Home

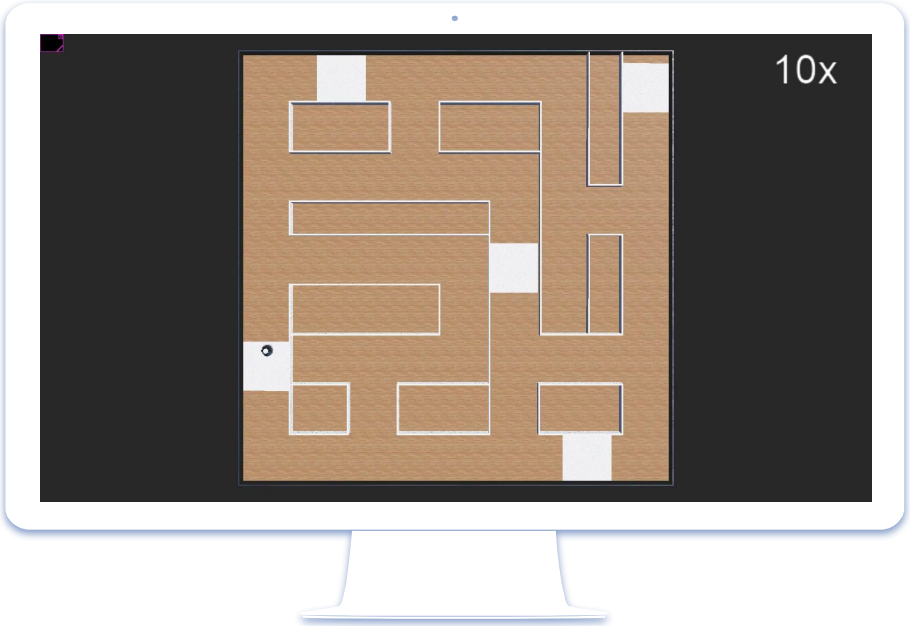
Multi-Goal - Final Results

- **Two paths changed:**
  - From *Restaurant* to *Home.*
  - *From Library to Gym.*

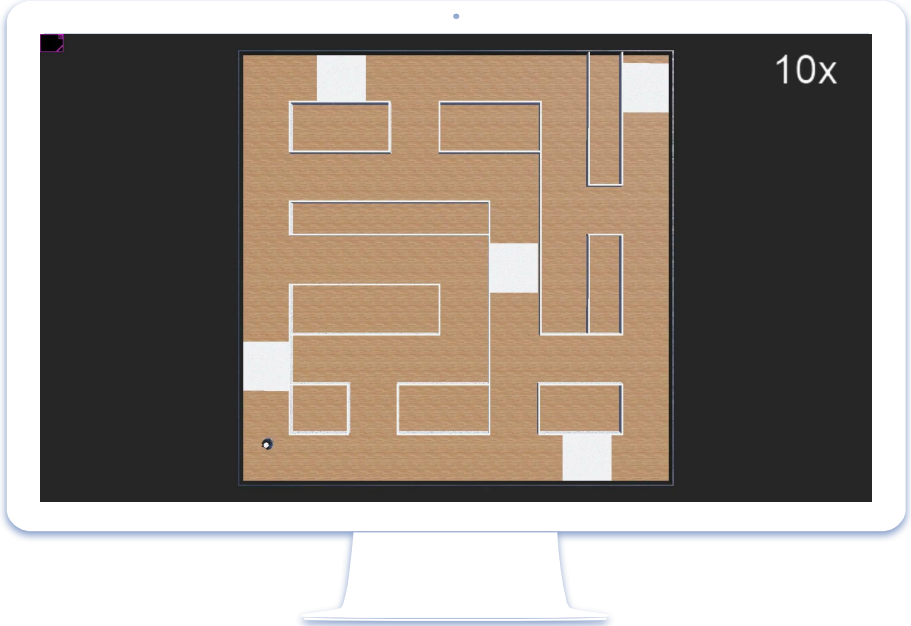


Trajectory between Multiple Goals
Home -> Library -> Gym -> Restaurant -> Home

Multi-Goal Robot Behaviour



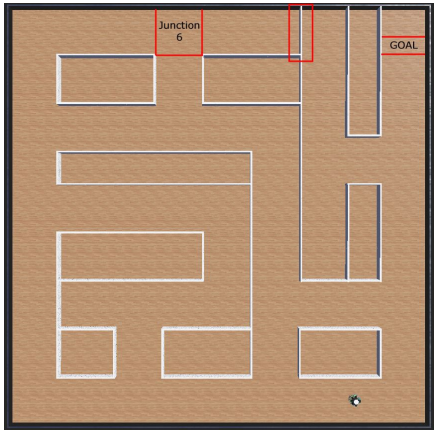
Multi-Goal w/o 180° rotation



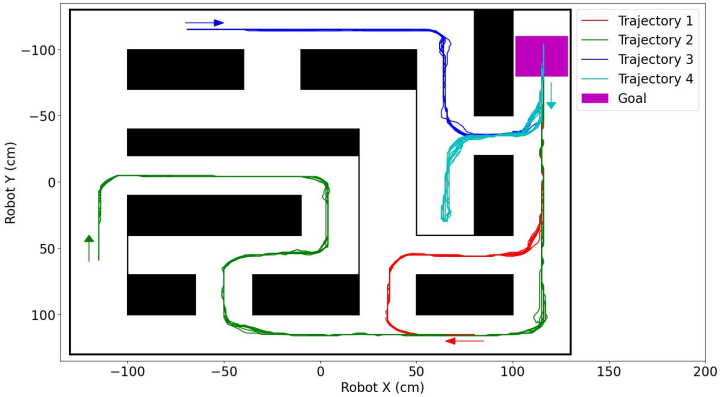
Multi-Goal with 180° rotation

Dynamic Behaviour

- Ability for the robot to adapt the acquired knowledge to changes on the environment after learning.
- A blocked path is detected when robot reaches the same junction it went through before.

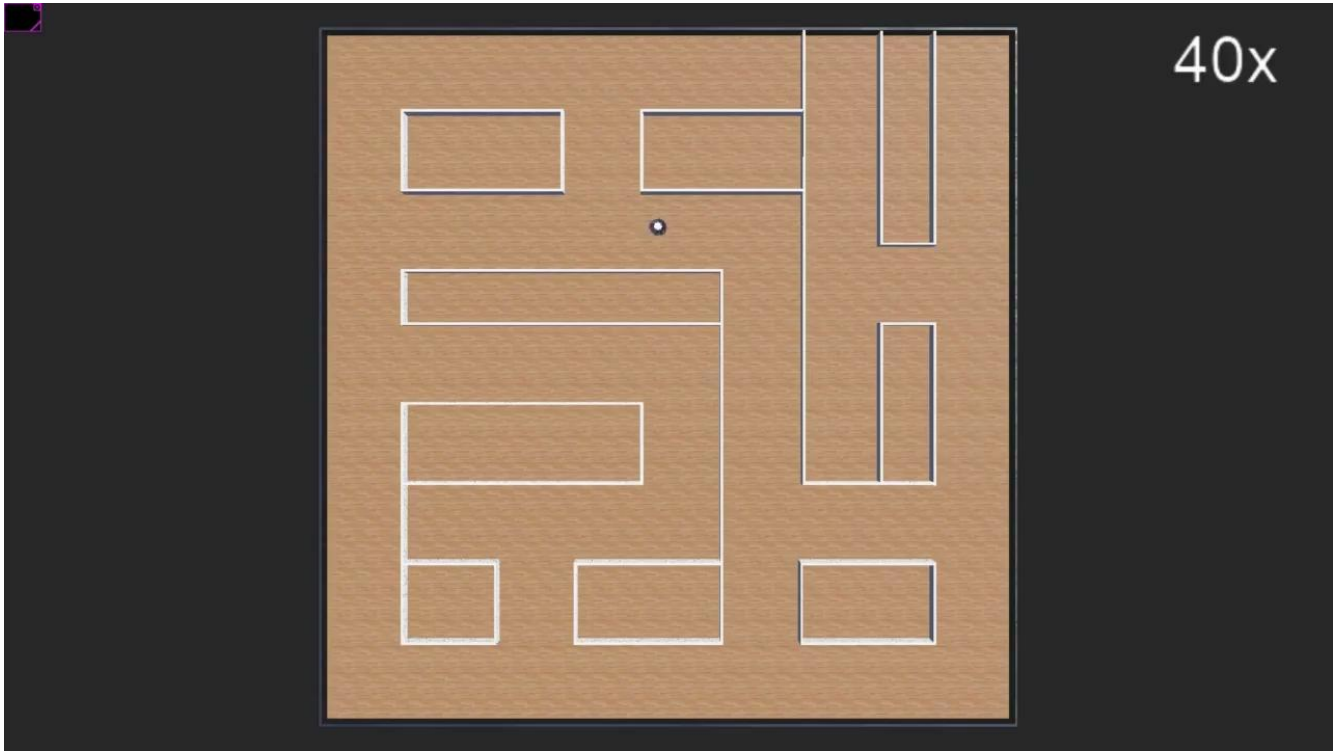


Maze 2 Top Path Blocked



Normal Trajectories in Maze 2

Robot Dynamic Behaviour Demo



Covered Topics

1. Introduction and Problem
2. Proposed approach and Objectives
3. Low-Level Exploratory Behaviour
4. High-Level Implementation
5. Multi-Goal and Dynamic Behaviour
6. **Final Remarks and Future Work**

Final Remarks – Contributions

- The HRL allowed for faster learning of higher-level decisions and to overcome some RL weaknesses. It proved to be very effective and easier to adapt in mobile robot navigation problems.
- The robot can navigate between multiple locations in a maze resorting to the topological representation of the environment and experience memorized during the learning.
- The dynamic behaviour endows the robot with the ability to adapt its trajectories whenever changes occur in the environment.

Final Remarks - Limitations

- A vision system to detect doors and goals is mandatory in a real robot. This feature is emulated during simulation.
- The extensive amount of time required to train the corridor task, as well as, the transformations performed for corners and doors require the use a simulator and only then is the learning transferred to the real robot.
- The environments must all be constituted just by 30cm-wide corridors, 90-degree corners and T-shaped junctions.

Future work

- Improve low-level layer to eliminate particular cases, as well as, apply other learning algorithms to train the tasks more effectively.
- Implementation of a vision system to detect doors and locations in real environments.
- Evaluate and compare the HRL approach to other approaches.
- Evaluate the R-Learning algorithm performance and the impact of the hyperparameters on the learning.
- Generalize the work to be implemented to different robots and/or environments.

THE END! QUESTIONS?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**