F21RO Intelligent Robotics

Group 3 - Coursework Report

Heriot Watt University

School Of Mathematics and Computer Sciences

Final Word Count:

Authors:

- Aarathi Raju – H00354041
- Soham Kakra – H00360439
- Varun Senthil Kumar – H00332328

Supervised By:

Prof. Talal Sheikh

Table of Contents

Introduction

The following report summaries the implementation of F21RO Coursework and discussing the implementation of Behaviour-Based Robotics (BBR) and Evolutionary Robotics (ER) controllers in a T-Maze using an e-puck robot. Both Task 1 and Task 2 in arenas T-Maze A and T-Maze B respectively have been implemented and working appropriately.

The aim was to create the quickest "robot-rat", by making it explore the T-Maze and finding the reward zone. The statistics have been recorded thrice for time spent to complete each scenario and getting the average run time, sorted using a table given in the coursework specification.

Rules:

Listed below were some base rules set which have provided the guidelines to complete this project:

- Using the e-puck robot as the "robot-rat".
- We have not moved any objects in the arena or tried to change the world.
- The dimensions of the arena, areas, and obstacles were not changed.
- The maximum time to perform each task of the coursework is under 5 minutes.

Activities:

In order to completing the project, we were asked to perform a set of activities, such as:

Activity 1: We chose the corresponding project repo from our lab material into the coursework folder.

Activity 2: We chose the appropriate arena environment.

Activity 3: We coded and ran the controller to perform the given tasks.

Activity 4: We chose the corresponding project repo from our lab material into the coursework folder.

This project was made using Webots, a free open source and multi-platform desktop application that is used to simulate robots. It provides a complete development environment to model, program and simulate robots. All the necessary materials, which were imported into the project files were provided on our canvas page.

Task 1

Methods & Implementation Rationale

In this section, we will be discussing about the Behaviour-Based Robotics BBR approach to solving the given problem. This approach uses proximity sensors in order to detect a boundary and turn to a specified direction. To detect the boundary ahead, middle of the sensor value is calculated and then provided a condition based on if the ground sensor was able to detect a black hint square (value of less than 500) or not, it then turns either right or left respectively after reaching the wall.

```
frontwall= (self.prox_sensor[0].getValue()+self.prox_sensor[7].getValue())/2
```

Figure 1: Getting the front sensor values

```
if frontwall > 100:
    if self.obj==True: #If it's seen black the bot will turn right
        self.left_speed = self.max_speed
        self.right_speed = self.min_speed
    else: #If it has not seen black then the bot will turn left
        self.left_speed = self.min_speed
        self.right_speed = self.max_speed
```

Figure 2: turning the e-puck

The initial behaviour this task is the ground searching. When a black hint square is found, the robot starts to learn how to navigate to the reward area. After one run, the system is transited to the learning of a position reaching behaviour. The learning then alternates between these two behaviours in order to maintain continuous learning without external intervention. If obstacles are discovered, the robot terminates the current learning individual in order to avoid the obstacles until no obstacles are discovered (Gu, et al., 2003).

Results & Analysis

This task releases continuous consistent results with the controller always working as expected. This means that the controller will only function properly in the same scenario and given environment as it has been hard-wired for it. Any other environment with a different layout may cause the e-puck robot to fail.

Some of the issues that arise with such approach is that, the robot creates a significant time delay while turning and halting. This could be because of the sensors values consistently changing with every small rotation in the wheels. It takes the robot approximately 2 seconds to turn in every scenario and another 2 seconds to come to a halt in the scenario where a black hint is detected. In total it takes about 9 seconds for the e-puck robot to complete each of the task. Thus making the turn and halting take about 44% of the total run time.

The initial positioning of the e-puck robot also has to be consistent in order for it to complete its turns smoothly, as it only moves towards reward area when the robot is facing forward. By turning the robot in any other way, it will be unable to reach either the objective or the top of the maze without first colliding with something.

Task 2

Methods & Implementation Rationale

In this section, we will be discussing about the Evolutionary Robotics ER approach to solving the given problem. ER provides distinct and sophisticated modelling tools for evolutionary experimentation. Its agent-centred nature, combined with its behaviour-based selection process, enables researchers to model difficult-to-study phenomena using a simplified and global statistics-based process. It gives the tools needed to investigate the evolution of social behaviours such as cooperation and communication (Doncieux, et al., 2015).

There exists a fitness function to outline the adjustments depending on whether the robot is moving forward or backwards, dodging obstacles, or rotating. To prevent rotations on the spot and banging into walls when moving backwards a penalty will be . A spin that is detected by both wheels spinning in opposing directions at any speed results in a +2 fitness score.

Results & Analysis

Discussion & Conclusion

	Time in Minutes			
	First Rn	**Second Run**	**Third Run**	**Average Time**
Task 1 T-Maze A	8.79	8.64	8.51	8.64
Task 1 T-Maze B	8.35	8.35	8.32	8.34
Task 2 T-Maze A				
Task 2 T-Maze B				

In ER, a robot's perception-action loop is determined by an artificially generated control mechanism. BBR originate through interactions between a placed agent and its surroundings. Neural networks are utilized in ER to control robots, and an evolutionary algorithm is used to construct and/or train the best potential NN to accomplish the task. Behaviour models, on the other hand, are represented by signatures written as character strings and compared using the string edit distance (Trujillo, et al., 2008).

Upon analysing the table above, we can conclude that when the e-puck robot runs using BBR method, it was consistent and predictable. This comes with limitations that it can only perform in this set environment and no other alternative. Whereas if the ER method works, it would show that the robot can perform under various different situations but would take indefinite amount of time to achieve the reward area.

References

Gu, D., Hu, H., Reynolds, J. & Tsang, E., 2003. GA-based learning in behaviour based robotics. *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automationcfor the New Millennium,* Volume 3, pp. 1521 - 1526.

Doncieux, S., Bredeche, N., Mouret, J.-B. & Eiben, A. E. (., 2015. Evolutionary robotics: what, why, and where to. *Frontiers in Robotics and AI,* Volume 2, p. 4.

Trujillo, L., Olague, G., Lutton, E. & Vega, F. F. d., 2008. *Behavior-based speciation for evolutionary robotics.* Atlanta, GA, USA, Genetic and Evolutionary Computation Conference, GECCO .