Project Proposal

Wall-following robot using reinforcement learning

Anna Sellani
up202107998

Matheus Teixeira
up202110548

Miguel Lopes
up201909924

Faculty of Science of the University of Porto, Porto, Portugal
Bachelors of Artificial Intelligence & Data Science

1 Goals

The goal of this paper is to develop and implement a reinforcement learning-based controller for a wall-following robot, using deep reinforcement learning (DRL) techniques instead of the traditional P controller methods. The input to the DRL comprises the LIDAR sensor readings from the e-puck robot, enabling the agent to perceive its environment. Regarding the output, it can either consist of discrete actions or continuous values, dictating the robot's linear and/or angular velocity for optimal wall-following behaviour. By exploring the efficacy of DRL in this context, the paper aims to provide insights into the potential advantages of using advanced machine learning techniques for robotic control tasks, particularly in scenarios where traditional methods may fall short. Through empirical validation and comparative analysis, this study seeks to demonstrate the effectiveness and efficiency of the proposed DRL-based controller in achieving accurate and robust wall-following behaviour.

2 Proposed approach

2.1 Description of the setup

As it is not possible to use a physical robot for the experiments, Webots is the chosen virtual simulator for the project. The robot to be used is the e-puck: simple but robust enough to fulfil the necessities. The robot's LIDAR data is going to be the main source of information to train the DRL models, as the data is detailed enough to satisfy the quality requirements.

For parameters of the environments and scenarios, the base environment it's going to be the same as the one in exercise 4, in the second practical exercise of the course. For the different worlds for the tests (and possibly training), this is yet to be determined, as at the current state of development this is not clear yet.

2.2 Algorithms

On-policy and off-policy algorithms are going to be used. The following table displaying the various algorithms was taken from Stable-Baselines3, very useful for a birds-eye point of view.

- Off-policy:
  - Q-Learning
  - DQN
- On-policy:
  - PPO

3 Proposed experiments

The experiments consist of training the different algorithms, followed by using these trained models on different test scenarios. The models are then going to be submitted through the following experiments:

1. Measure the time (in seconds) taken by the robot to complete the course
2. Calculate the average error (in centimeters) between the robot's actual distance from the wall and the designed distance
3. Evaluate the smoothness of robot movements empirically by analyzing the robot's trajectories
4. Count the number of collisions with the wall
5. Evaluate the algorithm's processing efficiency by recording peak and average CPU usage
6. Evaluate the algorithm's memory efficiency by recording peak and average memory usage

Every experiment is going to be tested in three different scenarios, given three different algorithms:

- Q-learning (does not involves neural network)
- DQN (version of Q-learning that involves neural networks; off-policy)
- PPO (state of the art algorithm; on-policy)

4 Objectives for each upcoming week

The proposed objectives to have until the suggested dates are the following:

- (01/04) Submission of the project proposal
- (09/04) Implementation of the DRL algorithms
- (16/04) Implementation of the DRL algorithms
- (23/04) Connection between the ML models and the simulator; creation of different environments (at least two)
- (30/04) Model training
- (07/05) Model training
- (14/05) Project's checkpoint and first evaluation of the models
- (21/05) Parameter tuning and test in the different environments
- (28/05) Final results and completion of the scientific paper
- (03/06) Final delivery

5 Objectives to be accomplished and presented in the checkpoint

1. Implementation of the algorithms
2. Connection between the models and the simulator
3. Creation of different environments
4. Model's first analysis on base environment