Marble Finding Robot – Project details for the Artificial Intelligence part
October 2018

According to the general project description available on Blackboard, you must:

- Use fuzzy control for local obstacle avoidance.
- Apply Q-learning to find effective navigation strategies.

This document provides details on what each of the two bullet points encompasses.

Use fuzzy control for local obstacle avoidance

You must define linguistic variables, linguistic values, universes of discourse, and one or more rule-bases that enable a robot to navigate effectively from its current location to a predefined target location without colliding with obstacles. This means that the robot must know where the target location is relative to its own location and orientation.

The focus is on low-level control, for instance, safe navigation inside a room or navigation from one room to a neighboring room. Don't try to design your fuzzy controller to solve complex navigation tasks, such as solving mazes or similar; that is not what fuzzy control is suitable for. Instead, focus on obstacle avoidance.

The priority is on demonstrating your knowledge of fuzzy control and your ability to use and test it correctly. You should explain your design choices and present results of tests that demonstrate the performance of your controller. Your experiments should demonstrate in which scenarios your controller performs well, and in which scenarios it performs poorly or even fails. Note that your focus should be on demonstrating proper engineering skills, and not solely on getting a robot to move as fast as possible from A to B. Hence, you can still get a top grade if your robot fails in certain situations – as long as there is a good reason and you can convincingly explain that reason.

Your report should contain:

- Definitions of variables (inputs and outputs), values and rules. Justify your design choices.
- Definition of your experimental procedure: test setups and environments in which your controller is tested. Justify your experimental design.
- Presentation and analysis of test results.
- (if necessary:) Additional experiments and analysis to test any hypotheses that arose as a result of the initial set of experiments.

Apply Q-learning to find effective navigation strategies

Assume that you have an environment, such as `bigworld`, where there are multiple rooms. A certain number of marbles are distributed in the rooms according to a predefined probability distribution. You must use Q-learning (or a related reinforcement algorithm) to learn effective navigation strategies that enables a robot to maximize its expected return in terms of marbles collected within a fixed time *T*.

The focus here is on high-level control, e.g. in which order to visit rooms, and not on low-level motor control. Note that learning need not be done in the full-blown Gazebo simulator but can be done in a simplified abstract environment.

Your report must include:

- Definition of states (make sure that the task has the Markov property, for instance, by keeping track of which rooms have been visited). Justify your choices.
- Definition of your experimental procedure: test setups and environments in which your Q-learning is performed, and the resulting controller is tested. Justify your experimental design.
- Presentation and analysis of test results.
- Demonstration of a learnt navigation strategy in the Gazebo simulator (if feasible).