1 Test of Fuzzy Controller

Written by Frida Bech Schrøder, Nicoline Thomsen and Mikkel Skov Maarssø

1.1 Introduction

The purpose of this experiment is to observe the behaviour of the implemented fuzzy controller, and determine if it acts as expected according to the predetermined rule base.

1.2 Theory

Four tests are performed with different starting positions, however all are facing the same direction, towards the right side. Each robot will freely explore the map without any marbles for 8 minutes, only controlled by the fuzzy controller.

1.3 Experiment

1.3.1 Setup

The test will put the robot at four different positions in the premade environment `bigworld.world` in the Gazebo folder. Multiple versions of this world, all with different start positions of the robot, has been made. A C++ program (`gzserver_manager_fuzzy.cpp` in the TEST folder) has been written to start and stop the gazebo server with the different worlds. For this test it also starts and terminates the `robot_control` executable, which is configured to run the test.

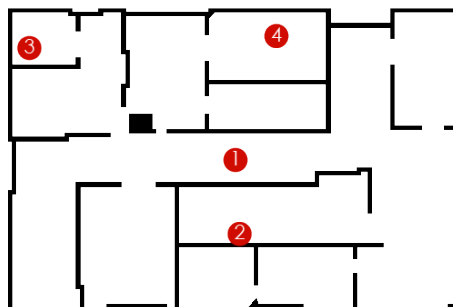The different start positions can be seen in figure 1.

Figure 1: Different starting positions of the robot. All robots will be facing to the right

The actual test is performed by the `testFuzzy.hpp` in the folder fuzzy_and_CV > scr. This program maps the position of the robot and print the whole path to a new image.

1.3.2 Procedure

Make sure the line `testFuzzy(&map, posex, posey);` is uncommented in `main.cpp`. Navigate to the folder fuzzy_and_CV > CMake and write the command `cmake`, then `make` in the terminal.

Navigate to the folder fuzzy_and_CV > TEST, and compile the source file `gzserver_manager_fuzzy.cpp`
(`g++ gzserver_manager_fuzzy.cpp`) and run the executable. This will perform the test
and save the map with the path to robot took.

1.4 Results

(a) Position 1
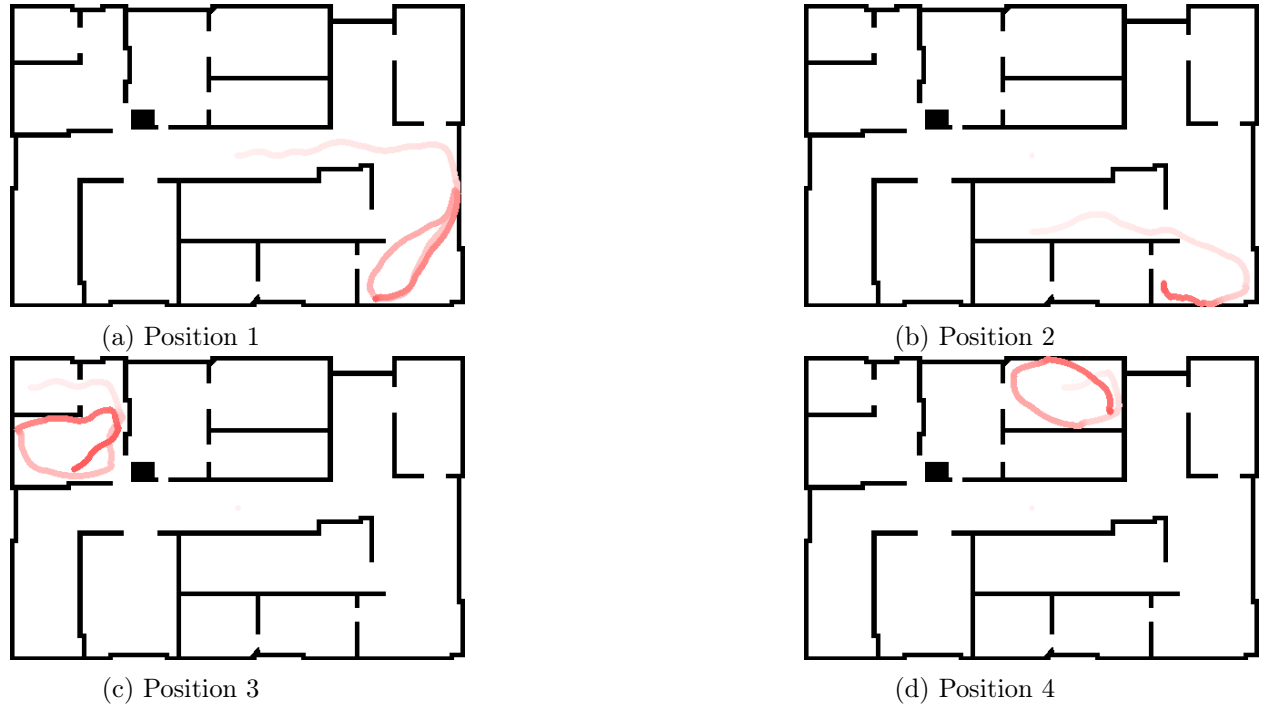
(b) Position 2

(c) Position 3

(d) Position 4

Figure 2: Fuzzy control path with different start positions defined in figure 1.

Figure 2 is shows the different paths the robot chose to take during the duration of the
run.

A common factor for all the tests is that the robot seems to cling to the wall. This is not
expected, since the rules state that if an obstacle on one side is close, turn away from it.

It does however not at any point collide with the wall, so it can be said that the fuzzy
controller succeeded in obstacle avoidance.

1.5 Conclusion

The robot does not behave the exact way as expected, but it is avoiding obstacles.