Lab 2 - exercise
Lab 2 Report

Kristoffer Rakstad Solberg

February 8, 2018

Department of Engineering Cybernetics

1 The Algorithm

diagram.jpg

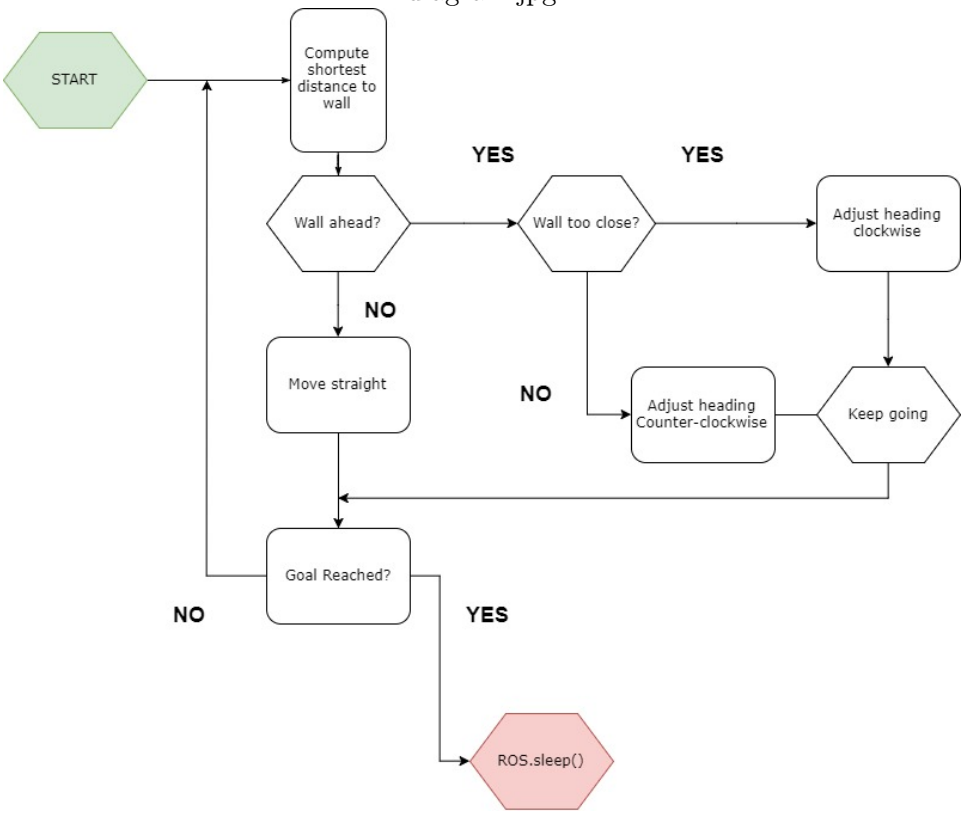

Figure 1: The Algorithm

1.1 Step 1 - Drive straight

The Robot will drive straight ahead with a constant velocity while constantly listening to the topic $/scan$. This topic publishes a range of distances on a vector form $std :: vector < float >$.

1.2 Step 2 - Scan for shortest distance to wall

When the Robot localizes a wall within 1.2 units ahead, it will start turning right to make sure the wall is always on the left handside.

1.3 Step 3 - Follow the wall

The drone will always try to follow the wall on its left handside. If the wall gets to close the robot will turn slight right. If the drone comes to far away

from the wall it will turn right. If there is no wall in sight, the robot will keep going straight.

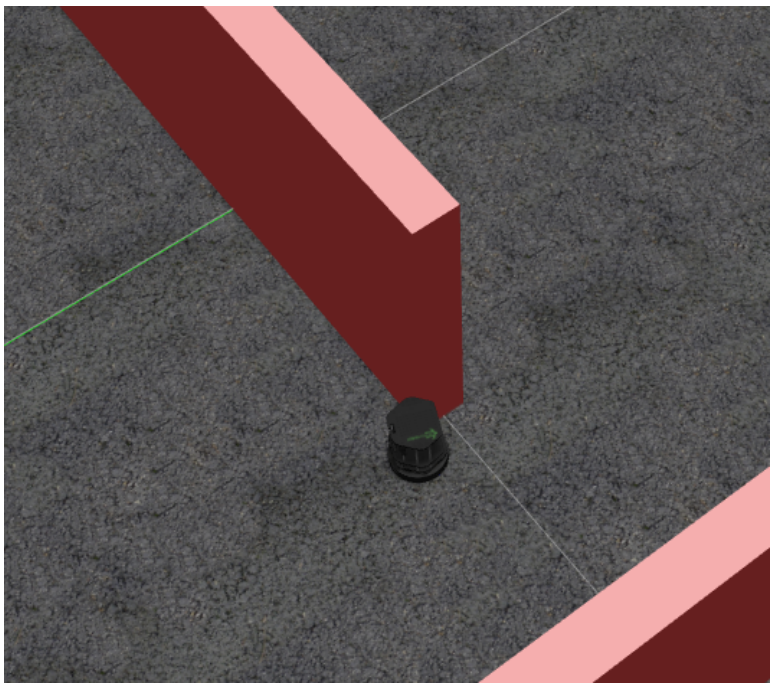

Figure 2: The Algorithm

1.4 Finding goal

The goal are defined as coordinates in the code. The robot will always check whether or not it is within the coordinates, if not it will keep driving. When the coordinates are found the robot will perform a sleep. After the sleep the robot will find the nearest wall, place the wall on its left handside and follow the labyrinth back to the beginning.

2 Important notes

2.1 Scanning shortest distance to surroundings

I found out that the scanner had issues when an obstacle was either too close or too far away. You would always have the risk of getting *nan* as a result of the measurements. To avoid this you could set up a for-loop within the Laserscan callback function that would always search for the shortest distance within reach. this way you would always avoid invalid measurements.

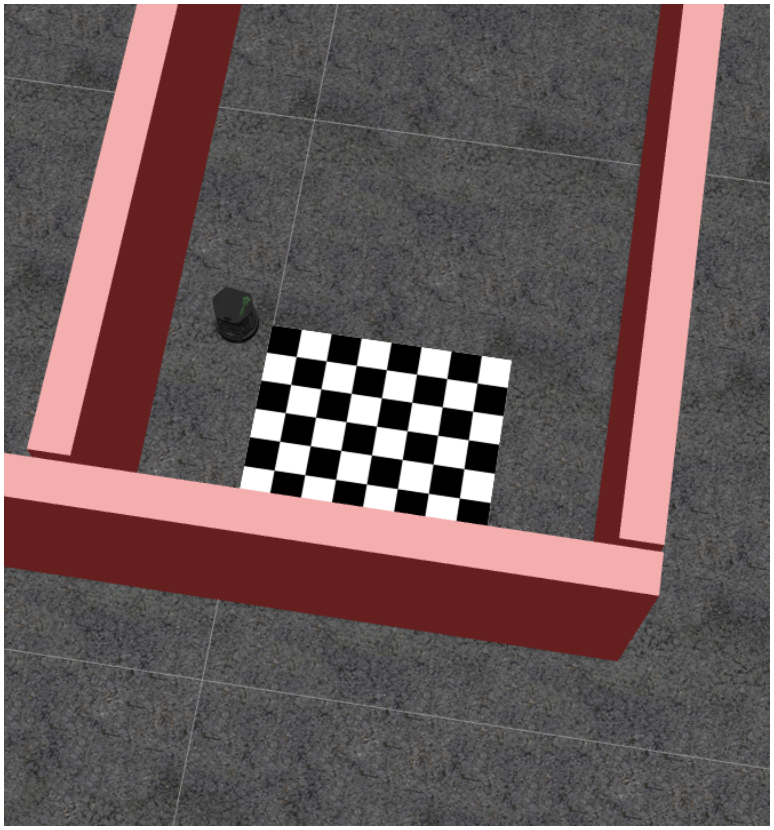

Figure 3: The Algorithm

2.2 Detecting a corner

If the sensor measurement from the left sensor somehow suddently get really large while the robot is following the wall, this would mean that there would be a sudden opening or a corner at the wall. The code will manage to complete the corner by trying to minimize this large sensor-value by turning the same direction. This is the only way the robot can make a left turn, otherwise the robot will always make a right turn.

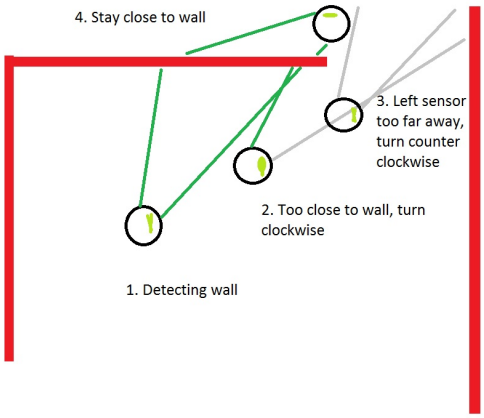

Figure 4: Screenshot task 2

3 Performance

The Robot makes it to the first goal in about 70 seconds, and a about 190 seconds back and forth.