

Wall Following STDR

Ângela Cardoso and Inês Caldas

Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

November 8, 2017

Abstract

bla bla bla

Contents

1	Introduction	1
2	Architecture	2
3	Results	2
4	Limitations	2
4.1	Map delimiting walls	2
4.2	Initial random movement	2
4.3	Desired wall distance	2
4.4	Initial robot position	3
5	Conclusions	3

1 Introduction

So [1] did.

2 Architecture

3 Results

4 Limitations

4.1 Map delimiting walls

In order to keep the robot from trying to leave the map, we had to create squared walls around the map. These walls are not the ones the robot is supposed to find and follow, but if it does find these walls it will follow them forever.

Of course this only happens for the small D, whose walls the robot is supposed to follow from the outside. In the cases where the robot is inside the large D or between the two D's, it cannot move through those walls, therefore it will never find and follow the square map delimiting walls.

For the small D, we defined an initial position for the robot that is closer to the D than to the outside wall, although it sees neither of these walls. Thus, even though the initial movement of the robot will be random, it is more likely to find the small D wall and follow it. In case it finds the outside wall, the program should be stopped, but if it is not, the robot will still follow the outside wall without any problem.

4.2 Initial random movement

Until it finds the wall, the robot moves randomly, that is, at each step it chooses random linear and angular velocities. Depending on the initial distance of the robot to the nearest wall and its orientation, this random search may take a long time.

We believe this wall finding time could be improved by making it more likely for the robot to move forward than for it to turn. This is clear if the robot is inside the wall it is suppose to find, like when it is inside the large D or between both D's. But in the case where the robot is outside the small D, going forward with higher probability might increase the likelihood of the robot finding the square walls delimiting the map, before it found the D walls.

Since the map delimiting wall is only there to stop the robot from trying to leave the map and not to be followed, we kept the robot moving randomly. In any case, the only downside is longer waiting time.

4.3 Desired wall distance

The distance the robot tries to keep from the wall can be configured by the user. If the chosen distance is too small, the robot may not be able to turn on sharp corners. This happens because while turning the robot gets too close to the wall and perceives that it has run into it, which makes it stop altogether.

Perhaps there is some way to make the robot resume its movements if it runs into a wall, but we did not pursue that. Especially because the correct way to handle this would have been to force the robot to avoid all obstacles, which we also did not try.

As an example, for the large D that the robot follows from the inside, if the defined distance to the wall is 0.6 or less, the robot runs into the wall at the left corners of the D .

Although we did not test it, we are convinced that for smaller angled corners, the desired distance that the robot should keep from the wall must increase in order for it not to crash.

4.4 Initial robot position

If the initial position of the robot is on top or too close to a wall, it will not move. This is related to the previous limitation, as we did not try to get it to move after running into a wall. So the only way to avoid this error is to make sure that a safe initial position is configured for the robot, according to each map.

5 Conclusions

References

- [1] A. N. Trahtman, *The road coloring problem*, Israel J. Math. **172** (2009), 51–60.