Experiments with subsumption in robotics

João Mota
FEUP
up201303462@fe.up.pt

João Cabral
FEUP
up201304395@fe.up.pt

I. Introduction

WITHIN the context of the Robotics class available in the integrated masters course in Informatics Engineering and Computation offered by the College of Engineering of the University of Porto, the authors of this paper developed a reactive robot with a basic architecture which allows it to follow an inner or outer wall, whenever one is found. The robot, which is simulated in the *STDR* simulator for ROS, consists of a single body to which two lidars are attached on each side, at a 30 degree angle from the front of the robot. The robot will move in a map which consists of a simple D shaped wall in the middle of a square space, delimitated by straight walls. A top-down view of the map is provided in figure 1.

The contents of this paper are as follows: First the robot's architecture is presented in detail, discussing its programmed triggers and responses. After that, the results obtained by the authors' current implementation are presented and discussed. Next, the limitations of the current implementation are enumerated and explained. Finally, conclusions about the project and the implementation are written.

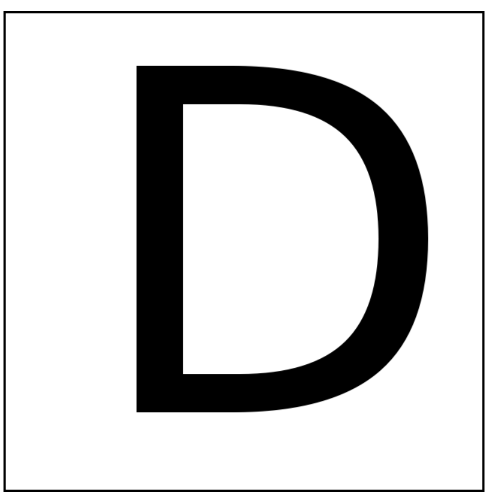

Fig. 1. Top-down view of the simulated map

II. Architecture

THIS robot follows a multi-layer subsumptive architecture wherein each layer implements a simple reactive behaviour, with the behaviour of the entire system emerging from these simple behaviours. The robot's simplified subsumption architecture consists of two layers. Each layer receives as an input the readings of the two lidar sensors present on the robot. Contrary to what is described by Brooks, there is no possibility of a layer masking or altering the inputs received by the layers below it.

Each layer outputs velocity commands to a multiplexing node, whose output can be controlled by all nodes on the system. Therefore, under this very simplified architecture, there is no possibility of two nodes providing commands to the robot at the same time. Furthermore the hierarchical features of the subsumptive architecture described by Brooks are not strictly enforced. Lower nodes can in theory inhibit the output of upper nodes by instructing the multiplexing node to select other outputs sent by nodes representing lower levels of the control system.

The first layer is responsible for making the robot wander randomly across the map, by instructing it to move forward at a constant velocity and by setting a random angular velocity on a second by second basis. This node does not use any of the available inputs to calculate its outputs, which allowed for the simplification of the subsumptive architecture described in the previous paragraph.

The second layer is responsible for making the robot follow a wall. As the node starts it monitors the sensor input from the lidars looking for an input corresponding to an obstacle. As the first such input arrives, this layer registers in which side of the robot the obstacle was detected. Afterwards it inhibits the output of the first layer and commands the robot according to a simple rule: it will rotate in the direction opposite to that of the discovered object when its sensors detect an obstacle on the side the wall was detected at, and it will rotate in the direction of the wall when it detects no objects.

III. Implementation

The architecture above described was implemented under the Robotic Operating System. A package was implemented which consists of two ROS nodes written in C++, one for each of the subsumption layers, of a launch file which starts the two nodes, the STDR server the simulation will run under, the STDR nodes which load the map and the robot, the configuration files which define the specifications of the robot, it's lidars, and the map, and a multiplexer node to control the inhibition of the output from the lower node.

IV. Results

AS mentioned in I, the purpose of this robot is to be able to follow walls when it finds them. In this regard, the paper's authors implmentation works within the expected bounds. At the beginning of the implementation, the robot will wander aimlessly through the map until it finds a wall to follow.

When one of the *lidar*s first reports a non-infinite distance, the wandering behaviour is immediately inhibited by the upper layer and the side the wall of the robot the wall was detected on is registered. As per the upper layer's programmed behaviour, the robot will turn away from the wall with an angle that follows the following equation:

$$\frac{2}{min(reportedDistRight1, reportedDistRight2)}$$

$$-$$

$$\frac{2}{min(reportedDistLeft1, reportedDistLeft2)}$$

This equation was obtained through extensive empirical experimentation, being the formula that produced better, smoother results.

After having registered the side the wall appears to be on, the robot will then iteratively behave as follows: Whenever the lidars start reporting an infinite distance again, the robot will turn toward the the wall at an angle of 1.7 radians, again, a figure obtained empirically. Upon once more sensing the wall, it will turn away from it according to the formula presented above.

This algorithm results in the robot being able to follow both inner and outer walls without any change to any parameters. It also can operate within the constraints of two walls as illustrated by figure 2.

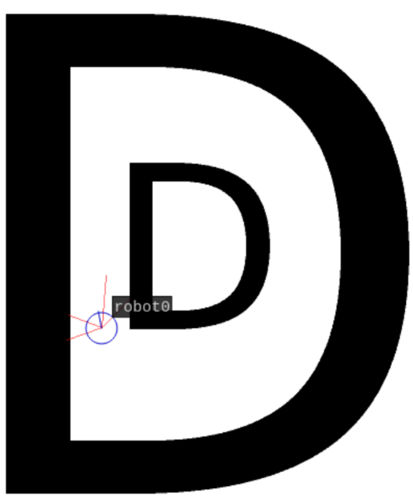

Fig. 2. Simulation view of the robot traversing a path delimited by two walls

V. Limitations

THE developed solution is a very simple reactive robot without any deliberative capabilities or internal representation of the state of the world, besides the side in which it first detected a wall.

It will always follow the first wall it encounters without the ability to switch to following walls which appear on the opposite side.

Additionally the approach used where the angle at which the robot turns is dependent on it's distance to the walls may cause the robot to fail to make turns where the curve is too slim by colliding with a wall, as well as turns where the curve is too wide by turning around and starting to follow the opposite wall.

VI. Conclusions

THIS project allowed this paper's authors not only to learn more about reactive architectures and their implementation, especially subsumption, but also to develop considerable technical skills regarding the *ROS* framework and some of its components, such as *stdr*.

All in all the project's goals were achieved, considering the results detailed in IV and the limitations presented in V.

VII. References

References

[1] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal on Robotics and Automation*, vol. 2, pp. 14–23, Mar 1986.

[2] R. Desai and D. Miller, "A simple reactive architecture for robust robots," 1998.