BitBot, a simple wall follower reacting robot

Vitor Minhoto, up201303086@fe.up.pt, MIB

***Abstract*—This project is about a little reactive robot: BitBot. It has two sonar sensors and its goal is to be a Wall Follower Robot. BitBot was simulated in STDR and developed in ROS. It was tested on a map with the D shape and it successfully achieved its objective. Moreover the architecture, results, and conclusions are presented and discussed also presented and discussed.**

***Index Terms*—Reactive behaviour, subsumption architecture, wall following.**

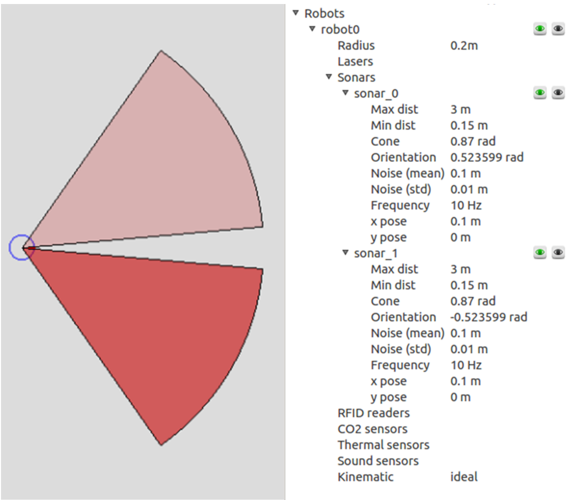
I. Introduction

INTELLIGENT Robotics is a vast area that has been growing over the past years. This happens for a reason: Society is interested in robotics because it makes peoples life easier. From autonomous driving cars, medical robots, educational robots to company robots, these machines help and improve Humans daily tasks. Even though there are numerous robots with a huge number of tasks and goals, one thing is common among them. They are intelligent! This means that there are able to autonomously make decisions and behave accordingly to the environment and changes in the world. Following this topic, the aim of this work is to build a simple robot that the final goal is: Follow a Wall. In this paper is presented a little robot, named BitBot, that has reactive behaviour. It is simulated in STDR (Simple Two Dimensional Robot Simulator) and developed with ROS (Robotic Operative System). The remainder of this paper is organized as follows: Section II gives details about the BitBot Specifications and Architecture. Section III shows results obtained. Finally, Sections IV provides brief conclusions.

II. Robot Specifications and Architecture

BitBot is a simple circular robot with two sonars placed at +30 degrees and -30 degrees (+0.52 or -0.52 radians) measured from the front of the robot. These sensors can measure a max distance of 3 meters and a minimum distance of 0.15 meters. The cone radius is 0.87 radians.

The design and development of BitBot architecture was an iterative process with trials and errors. The first try was just to connect to the sensor and print the readings. But the robot needed to move! So, it was developed Block1: Move randomly. This had a big problem! BitBot was always bumping into walls, not following them. To solve this problem the Block 2: If near a wall: STOP, was implemented. This way our robot no longer hits walls! But it didnt achieve is goal either! It still doesnt follow walls. It just stares at one infinitely. In other to achieve the final goal, it was introduced a feature to the previous block and a new one was created. Block 2 was modified from If near a wall, STOP to If near a wall: STOP, then rotate in Clockwise (inverse direction). After this rotation Block 3: Keep the wall at your Left at constant distance would become active. After this, BitBot will follow the wall until the rest of his days. So, the final Architecture (Figure 2) of this robot is a kind of subsumption architecture. Block 2, Avoid Obstacles, will always have precedence over the other two blocks. Between Block 1 and Block 3, Follow a Wall, will always have precedence over Move Randomly. A curious fact is that the Blocks were not implemented accordingly to his final precedence. If they were developed in order, probably there wouldnt be so much failed trials and wall bumping.



```
▼ Robots
  ▼ robot0
      Radius            0.2m
      Lasers
    ▼ Sonars
      ▼ sonar_0
          Max dist      3 m
          Min dist      0.15 m
          Cone          0.87 rad
          Orientation   0.523599 rad
          Noise (mean)  0.1 m
          Noise (std)   0.01 m
          Frequency     10 Hz
          x pose        0.1 m
          y pose        0 m
      ▼ sonar_1
          Max dist      3 m
          Min dist      0.15 m
          Cone          0.87 rad
          Orientation   -0.523599 rad
          Noise (mean)  0.1 m
          Noise (std)   0.01 m
          Frequency     10 Hz
          x pose        0.1 m
          y pose        0 m
      RFID readers
      CO2 sensors
      Thermal sensors
      Sound sensors
      Kinematic         ideal
```

Fig. 1. BitBot technical details.

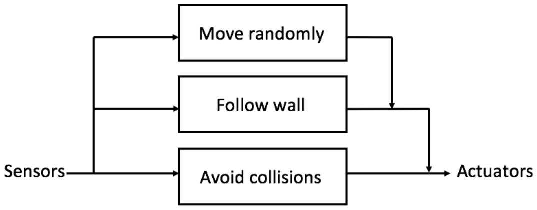


Fig. 2. Robot Architecture.

III. Results

The main task was to follow the walls of a specific map: a D shaped map both in the outside and on the inside. So, the map build was a D with a tunnel in the straight part (Figure 3) . This way BitBot will follow the wall on the inside, pass through the tunnel, following its wall, and continue its pathing following the outside wall.

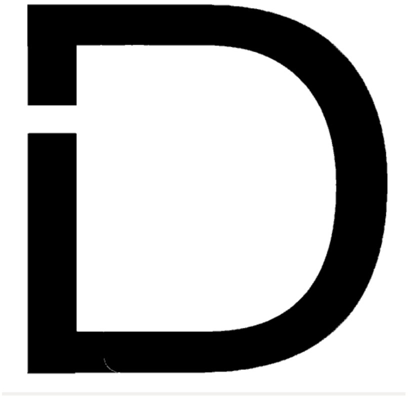


Fig. 3. D shaped map with tunnel.

The robot can follow the curve part and the straight part of the map without any problem (Figure 5) . Sometime the corners are a problem. For example,in (Figure 6) the corner of the map has an irregularity and the robot gets stuck. This problem might be due to the disposition of the sonar. The sensor doesnt detect an object, so BitBot thinks that there isnt an obstacle there, but in reality, he is blocked by the wall. Another map was proposed: A small D arena inside the big D arena. From the results in one specific area of the original map, the tunnel, we can take some conclusions about this challenge on this new map. This is true because the principle is the same: BitBot passing through two objects. The robot do not have any problem passing on this places, but it always follows the left wall. So if the robot is following the outer D, it will always be on the outer D. If it is on the inside D, the opposite happens.

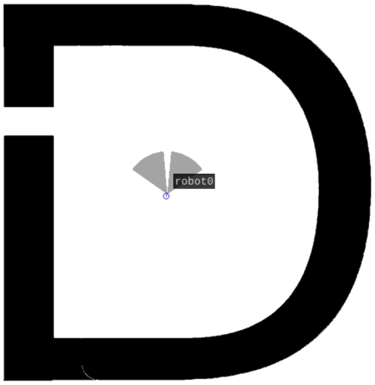


Fig. 4. Robot moving randomly.

.

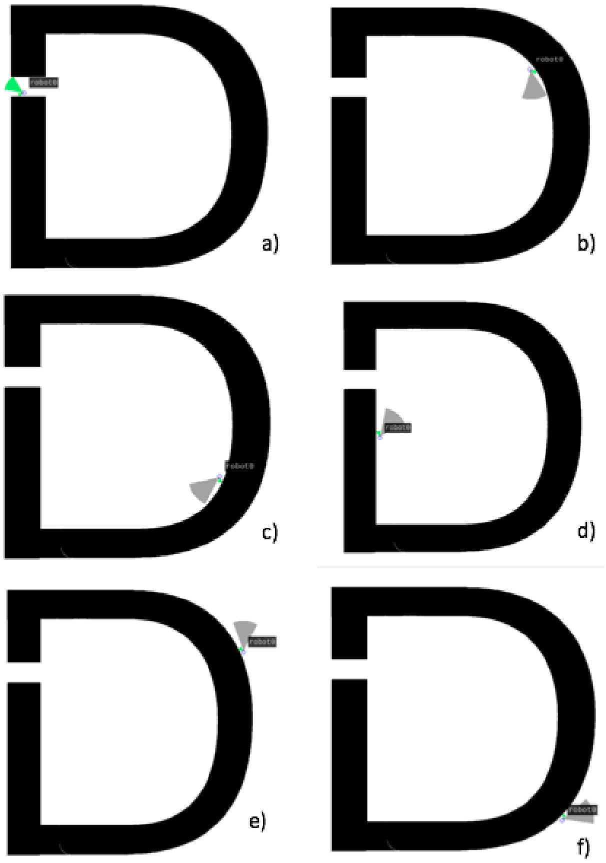


Fig. 5. BitBot traveling on the map: a) Passing through the tunnel. b) Following curve-top inside wall. c) Following curve-bottom inside wall. d) Following straight wall. e) Following curve-top outside wall. f) Following curve-bottom outside wall.

The big limitation of the implementation of this code is: all velocities are hard coded. Turning this hard code in some proportional value of the distance measured by the sensors could be beneficial to the algorithm, for example for smoother outer angle turn and faster coverage of the map.

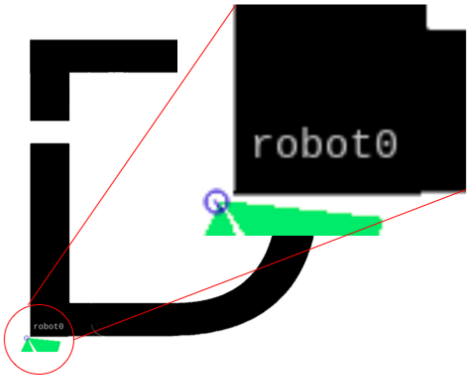


Fig. 6. Corner Problem

IV. Conclusion

It is presented a simple reactive robot, BitBot that can autonomously follow walls. It is simulated in STDR (Simple Two Dimensional Robot Simulator) and developed in ROS (Robotic Operative System). In order to perform this, some practise and exercise with this tools were performed. Also in order to get the architecture right some trial and error experiments were performed. This helped to both help understand basic architectures and motivate future improvement and developing of new blocks to build upon the already existent ones. Some improvements can be done in future work could be correcting the corner bug and replace the hard coded values velocity values.

V. References

OKane, Jason M. 2014. A gentle introduction to ROS. Columbia, SC: Jason M. OKane.

Sabanovi, Selma. 2014. ”Inventing Japan’s ‘robotics culture’: The repeated assembly of science, technology, and culture in social robotic.” Social Studies of Science (SAGE Publications) 342 - 367.

n.d. stdr simulator. http://wiki.ros.org/stdr_simulator.