Simple Reactive Robot

1st Ricardo Carvalho
Department of Informatic Engineering
Faculty of Engineering of University of Porto
Porto, Portugal
up201503717@fe.up.pt

2nd Vítor Magalhães
Department of Informatic Engineering
Faculty of Engineering of University of Porto
Porto, Portugal
up201503447@fe.up.pt

***Abstract*—In robotics, it is important to develop techniques to detect and avoid obstacles in order to traverse the environment. Utilizing Robot Operating System (ROS) and the Simple Two Dimensional Robot Simulator (STDR) Simulator package, a method for a robot to follow an angled wall was developed. With Light Detection and Ranging (LIDAR) sensors, wall detection was possible. Tests were made using walls with different angles and thickness to evaluate efficiency of the technique.**

***Index Terms*—Robot Operating System, LIDAR, STDR, ROS, Reactive Robot**

I. Introduction

The usage of robots has become more common and, with it, the necessity to develop ways for robots to cross the environment they're located in. New methods for robots to travel will have a positive impact on their functionality, allowing them to complete their goals faster.

The objective of this project is to create a reactive robot that allows a robot to follow angled walls using ROS and the STDR Simulator. The robot will be equipped with a laser sensor paired with LIDAR capabilities that will tell where the wall is located.

II. State of the Art

Robots capable of tracking walls and obstacles have been around for several years and they can range from simple reactive robots that follow a wall with only laser sensors to extremely complex systems using more sensors, such as cameras or sonars. There are plenty of tutorials on how to make a robot that follows a wall, using ROS. One example are swimming pool cleaning robots, that follow the wall as they clean the pool.

III. Robot's Architecture and Implementation

A. Architecture

The system follows a subsumption model that couples sensory information to action selection in an intimate and bottom-up fashion. [1]

B. Robot Design

The robot has differential locomotion with a round shape and a diameter of 0.2m. It uses a laser sensor (detailed in table I) in order to obtain information about its surroundings.

Rays	Range	Fan Span	Frequency
200	[0.1, 0.8]	[-100º, 100º]	10Hz

TABLE I: Laser Sensor Specification

C. Behaviour

The created robot is capable of detecting two scenarios: either the sensor detects a wall or there is no wall in range of the sensor. In the former, the robot is prepared to wander the map until it finds a wall. In the latter scenario, once a wall is detected by the sensor, the robot starts following it at a predefined distance, frequently correcting its trajectory.

D. STDR Simulator

The Simple Two Dimensional Robot Simulator is an open-source project. It is used to simulate the robot's position and navigation with a visual aid.

E. Solution

The problem presented can be divided into two simpler ones, according to the aforementioned scenarios.

Upon starting the program, if the sensor is not able to detect a wall, the robot starts roaming the map, with a constant linear velocity of $0.4m/s$, until it finds one. In order to keep the robot from wandering off the map, it was made aware of the map's dimensions. Therefore, and using odometry to estimate the robot's positions at all times, when the robot comes close to the map's edges, it randomly rotates θ degrees, with $\theta \in [90, 270]$, and continues to roam in a new direction. [Fig. 1]

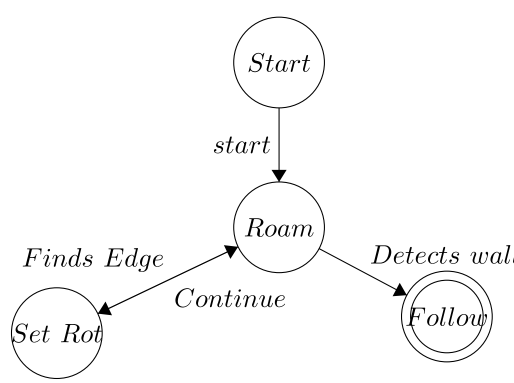


Fig. 1: Roaming State Machine

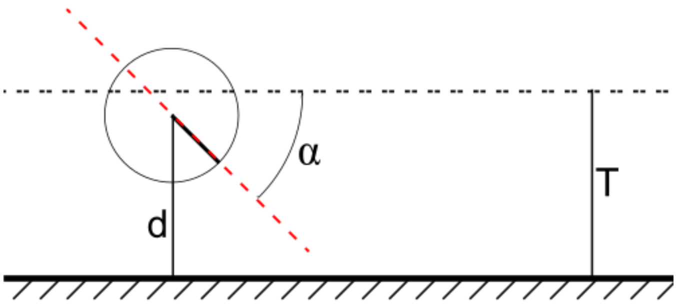


Fig. 2: Schematics of the Robot Orientation Problem

The second and more complex scenario is to, once a wall is found, follow it at a given target distance. This scenario is schematically represented in figure 2.

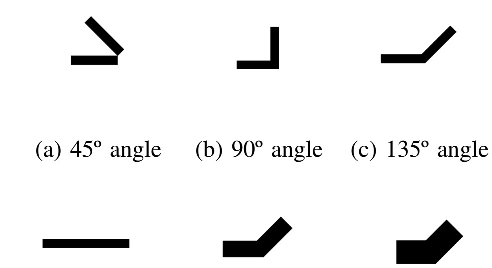
Thus, when the robot detects a wall at a distance, d, at an angle of $90 - \alpha$ degrees, it is possible to calculate the needed angular velocity using (1), adapted from [2].

$$\omega = -v * k * (sin(\alpha) - (d - T)) \tag{1}$$

The parameter T represents the target distance to keep from the wall, which was set to $0.05m$, v represents the linear velocity, set to $0.2m/s$, and k is an adjustable parameter that after some tests was set to 25.

IV. Experiments

In order to test multiple cases of angled walls, several maps were created with varying parameters such as the angle $\alpha \in [0, 180]$ and the wall thickness $d \in [0.2, 0.6]$ [Fig. 3]. All maps have width and height of 10m and a resolution of 0.01 meter per pixel.



(a) 45º angle (b) 90º angle (c) 135º angle

(d) 0º/180º angle (e) d = 0.4 (f) d = 0.6

Fig. 3: Some maps used for testing

V. Results

Upon testing the robot on several maps, it was concluded that it was able to find a wall from any starting point within the maps bounds and, once found, follow it in a counterclockwise direction.

In the following image, the robot was able to find the 45º angled wall and follow it.

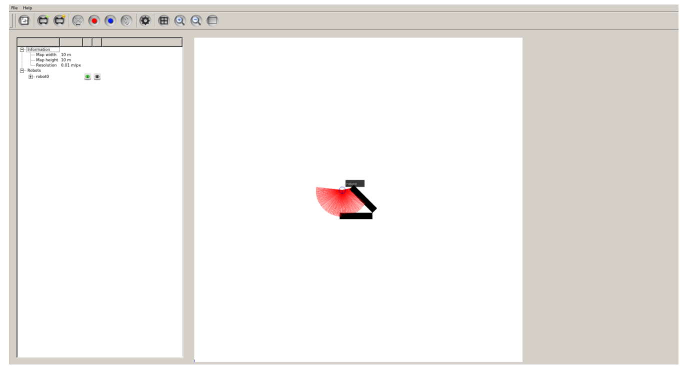


Fig. 4: Map with a 45º angled wall

VI. Limitations

During testing and refinement of the algorithm, some limitations were found:

- The initial wander to find a wall has a certain randomness to it, ergo, although unlikely, it is possible that the robot enters a loop bouncing between the maps edges without finding a wall;
- When testing the wandering feature, it was detected that the laser sensor mistook the top and bottom edges of the map for walls, although not a limitations of the robot itself, it is worth mentioning as the solution found for it is the cause of the forthcoming limitation;
- Given the aforementioned limitation to the STDR simulator, the solution found was to implement a check on the robot's position. Therefore, even when following a wall,if the robot is facing an edge of the map and said edge is in range of the laser sensor, the robot will change its orientation randomly as if no wall had been found. Ergo it becomes impossible for the robot to follow a wall close to the map's edges;
- The equation used to calculate the angular velocity can be adapted to allow the robot to follow a wall both clockwise and counterclockwise depending on which side the wall is detected on. Yet, upon testing, some difficulties were created when turning inner corners, as the robot would head towards the corner and not around it;
- Depending on the robot's speed and the target distance to the wall, it is possible that the robot becomes stuck when completing an inner corner as it cannot rotate itself fast enough.

VII. Conclusions and Future Work

This paper presents an effective solution for reactive robots following walls. The techniques used allowed the team to develop interesting methods to resolve wall following problems.

The results showed that the robot is able to navigate through several maps with different angled walls. This can be helpful for robots to navigate through real life environments in order to avoid bumping into obstacles.

The main features to improve are related to the mentioned limitations in the previous section, particularly the current necessity to follow the wall in a counterclockwise direction.

VIII. Acknowledgments

We would like to thank our friends and colleagues who helped and guided us in the world of robotics.

References

[1] Ronald Arkin, Behavior-Based Robotics. Cambridge, Massachusetts: The MIT Press, 1998, pp 130.

[2] K. Bayer, "Wall Following for Autonomous Navigation," SUNFEST, University of Pennsylvania, Summer 2012.