Wall-following algorithm for reactive autonomous mobile robot with laser scanner sensor

Ana Rafael
Porto, Portugal
up201405377@fe.up.pt

Cássio Santos
Porto, Portugal
up201802025@fe.up.pt

***Abstract*—Mobile robots with wall-following capacity can be used in a number of applications, such as intelligent inspection, security patrol and venue service. In this paper, we present the implementation of a wall-following algorithm for an autonomous mobile 2 wheeled robot with a laser scanner. The perception of the environment is achieved with a scan laser, attached to the top-front part of the robot. The proposed wall-following algorithm makes a robot wander at random until a wall is found, then follows the wall - through an implemented proportional control to keep a constant distance from it - in the outside and inside of a V" and "W" shaped wall, respectively. The system consists of a reactive robot since it responds to an unknown environment without the need for wall modelling. Experiments were conducted in terms of wall finding and following in various environments for the production of correct and robust robot behaviour.**

***Index Terms*—Autonomous navigation, Mobile robots, Reactive robot, Wall-following**

I. Introduction

Autonomous mobile robots have a wide range of applications, namely as sweeping robots, inspection robots, service robots among other possibilities. Wall-following behaviour is among one of the most used navigation methods in autonomous robots. It consists in the ability to move along contours of walls and edges wile maintaining a safe and constant distance from it [1]. Most frequently this practice is achieved with distance sensors, such as ultrasonic or laser sensors. Such algorithms can be used as part of indoor environment's circulation control, maze solving and environment exploration [2]. Some problems in wall-following control are related with noisy data from the sensor, thus in the designed algorithm a mean of the wall distance measurements was considered. For the past few decades several algorithms have been proposed for wall-following. One of the most common category of these algorithms is related with reactive robot. Table I comprises a summary of the most relevant published papers in the field of wall-following algorithms for reactive robot.

In this paper a wall-following control of a mobile robot is presented with the use of a laser sensor. The considered robot is a deferentially steered robot; with two wheels driven independently. Thus in the next chapter an analysis of existing solutions for wall following used sensors and algorithms are presented. This document is organised by first presenting the used robot and machine states in section II, the virtual environment assessment is presented in section III. In section IV, the implemented algorithm for laser reading treatment and different virtual wall configurations is exposed and the results are given in section V. Finally, a discussion of the developed approach and future research is summarised.

TABLE I: Wall following algorithms and sensors .

Author (Year)	**Sensor**	**Method**	**Notes**
Turennout et al. (1992)	Sonar Sensor	Straight wall following	
Ando et al. (1995)	12 sonar sensors	Finite-state machine	No wall finding and object collision avoidance [3].
Carelli et al. (2003)	Sonar and odometry sensor	Corridor navigation; wall-following; object avoidance.	State variables combine the information from the two sensors.
Toibero et al. (2009)	Laser radar sensor and odometry	Continuous wall-following controller, obstacle avoidance with switching scheme	Has no saturation of angular velocity. Only the steering angle is controlled. Slow average speed of wall-following (15cm/s).
Charifa et al. (2009)	Boundary-following algorithm	Artificial potential fields	Has the local minima problem. Needs prior knowledge of the environment.
Wei et al. (2017)	Laser range finder	Wall-following and obstacle avoidance	Move along a virtual wall smoothly at adaptively adjusted speed [1].

II. System structure

The proposed algorithm flow is represented in Fig. 1. Thus, the systems produces an action in response to a stimulus of the environment. The admissible robot state machines are 1) Random wandering, for the initial state in which all the sensor readings are further than 5m and the robot wonders randomly; 2) Following wall, for the case where the robot has found a wall, thus state (1) is not verified; 3) Rotating, in the event of verification of positioning patters detailed in section IV-C. Thus, the developed algorithm is much dependent on the robot's sensor characteristics, responsible for the collection of data from the environment and the functioning of the robot. The robot, Fig. 2 is 50 x 30 x 7 centimetres and 6 kilograms. It is composed of a caster wheel - in the front centre - and 2 wheels - one on the right and the other on the left - connected to the chassis through joints that condition the possible movements of the wheels with relation to the chassis.

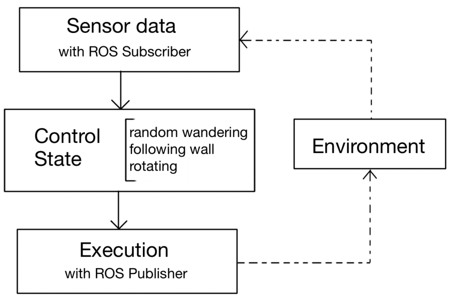


Fig. 1: Proposed algorithm flow.

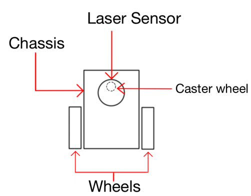


Fig. 2: Differential-drive robot structure.

III. Perception of the Environment

To be able to follow the wall smoothly, two main objectives must be achieved: 1) the distance between the robot and the wall should be kept as close as possible to a specified value; and 2) the robot's travel angle must be as parallel as possible to the followed wall. To achieve these goals the robot must be able to perceive the environment around it. Hence, robot's input signal is obtained by reading the laser sensor installed on top of it, with an amplitude of $[-126°, 126°]$, resolution of $0.25°$. The maximum velocity of the robot was set to 30 cm/s at 20Hz. The $0°$ are at the robot's front, and 1008 distance measurements are obtained by the sensor from points $0.25°$ spaced, Fig. 3. Instead of directly using all these points to control the robot along the wall, with the objective of reducing the computational cost and simplifying the actions control, only 7 values are selected to represent the information from the environment. The values are defined as the shortest sensed distance in each of the 7 regions presented in Fig. 4. The laser range is 5m, if no points with a distance smaller than this threshold are found in the sensing region the value for that region is defined as *Infinite* (wall away from the robot's sensor range).

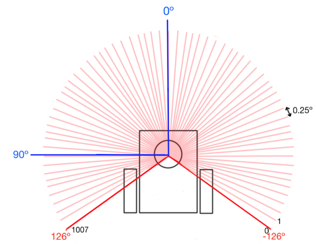


Fig. 3: Robot's sensor read distance points (not all the radial readings are illustrated).

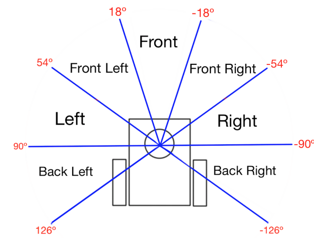


Fig. 4: Region separation of sensor measuring amplitude.

IV. Algorithm

The analysis of the values for the shortest distance to the wall obtained for each section will regulate the actions taken by the robot. When a mobile robot is performing a wall-following algorithm, it may encounter various kinds of wall shapes, such as straight, inner corner and outer corner. Generally, typical indoor environments can be seen as composed of these three kinds of walls [1]. The identification of inner and outer corners is important to decide when to perform the return action to stay on a specific side of the wall.

The configuration of the sensors selected empirically to decide which type of wall the robot is seeing is the following: for an inner corner the wall must be detected in the *front*, *front left* and *front right* regions, independent of the information present in the others regions (Fig. 5.a); for an outer corner the wall should be sensed either in the *back left* region or in the *back right* region with all the other sensors stating that there is no wall near (*Infinite*) (Fig. 5.b).

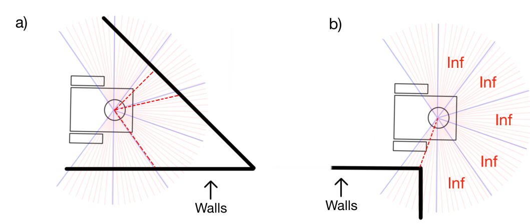


Fig. 5: a) Inner corner detection; b) Outer corner detection

The information from the regions, together with the verification of the robot's positioning pattern, are also used to decide which behaviour will be executed by the robot according to the three possible states presented in Fig. 1. Then the velocity (v, w) of the robot should be determined according to these states.

A. Random Wandering

Random Wandering state performs the random wandering situation of the robot. If there is no wall detection in any of its sensor regions the *Control State* defines its state as Random Wandering and starts publishing random values for the linear velocity (v) and for the angular velocity (w) of the robot. To ensure the smoothness of movement while wandering, the velocity values of the previous time instant are incremented by a random value at the range $[-1, 1]cm/s$ for linear velocity and $[-0.1, 0.1]rad/s$ for angular velocity.

In order to prevent undesired velocity values, the linear velocity is then limited between a minimum of $10cm/s$ so that the robot does not stop and consequently can not find walls and the maximum of $30cm/s$ as previously presented in section III. With regards to the angular velocity this was limited to a range of $[-1, 1]rad/s$ so that the robot does not spin constantly in the same place, thus when this limit is reached, the angular velocity is reset to zero.

B. Wall-following

As told on section III to be able to follow the wall the first objective is to keep the shortest distance between the robot and the wall near to a specified value (chosen as $50cm$ for this project). Therefore, the linear velocity is set to four levels.

1) $0m/s$ in the case the minimal wall distance is inferior than the set wall distance;
2) $0.5 * max\_speed = 0.15m/s$ for the case where the measured wall distance is inferior than two times the wall distance;
3) $0.4 * max\_speed = 0.12m/s$ when the orientation of the minimal distance to the wall is higher in absolute value than 1.75 - thus, with a lower linear velocity there is more time for the correction of the orientation through angular velocity; and lastly,
4) $max\_speed = 0.3m/s$ allowing for a faster movement towards the wall.

The angular velocity component is used for correction of the robot's wall distance and orientation. Hence, for wall distance correction the following analysis was implemented. Wall distance *error* is continuously calculated as the difference between the desired wall distance and the shortest measured wall distance and correction based on proportional and derivative terms is applied. The wall distance *error* is given by equation 1 where t_n is the time of n-th iteration and $d_{min}(t_n)$ is the distance of the closest wall measured from laser sensor and d_{wall} is the desired distance. The intervention from the discrete PD controller can then be seen in equation 2 where k_p and k_d are, respectively, the proportional constant and derivative constant.

$$e(t_n) = d_{min}(t_n) - d_{wall} \tag{1}$$

$$w_{PD}(t_n) = k_p e(t_n) + k_d \frac{e(t) - e(t_{n-1})}{t_n - t_{n-1}} \tag{2}$$

The second objective is to keep a perpendicular angle between the reference vector of the robot (0 degree) and the normal vector of the wall. Thus, the minimum measured wall distance angle ($\varphi_{min}(t_n)$) has to be at $\frac{\pi}{2}$ for $dir = 1$ or $\frac{-\pi}{2}$ for $dir = -1$ depending the robot's orientation with the wall - wall at the left or right side, respectively. The error that will be minimised is given by equation 3 and the discrete angle proportional control is presented in equation 4 where k_{p2} is the proportional constant.

$$e_\varphi(t_n) = \varphi_{min}(t_n) - \frac{\pi}{2} * dir \tag{3}$$

$$w_P(t_n) = k_{p2} e_\varphi(t_n) \tag{4}$$

Therefore, the final angular velocity is calculated as a sum of interventions from PD distance control and P angle control, shown in equation 5.

$$w(t_n) = w_{PD}(t_n) + w_P(t_n) \tag{5}$$

C. Inversion of direction

After the identification of inner and outer corners as exposed in IV the specified turning sequence is assessed. In the "V" scene, in order to have movement in the outside wall of the shape the sequential corner's pattern can be ['Inner', 'Outer', 'Outer', 'Outer'] or ['undetermined', 'Outer', 'Outer', 'Outer'], with the identification of one of this patterns the robot has to reverse its movement. Considering the "W" world the turning sequence is ['Inner', 'Outer', 'Inner', 'Outer']. When one of this patterns is matched then the angular velocity is set to $2rad/s$ and the linear velocity is set as zero. The circular movement is kept until the left or the right regions of the robot are closer to the wall then the established target wall distance.

In order to reduce noise effects or errors in the detection of corners that could derive from the approach of the robot to the wall or oscillations during the wall following movement, a corner is only considered when 1) 5 consecutive data samples had the defined activation pattern for each of the regions, and 2) the previous corner had been detected 20 seconds apart.

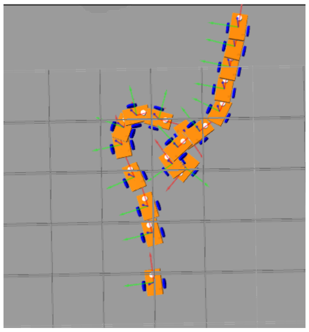
V. Simulations and Experiments

In order to verify the correct functioning of the developed algorithm simulations of the robot with two different wall configurations were carried out in a Gazebo 3D environment. Gazebo allows ROS plugins for message publication and subscription that are essential for robot control. The goal of these simulations were firstly the correct movement of the robot and capacity of wall detection detection. Secondly, the detection of corners and inner angles of the V and W wall shapes. Finally, after correctly detecting the inversion of direction patterns the capacity of correctly perform it was evaluated.

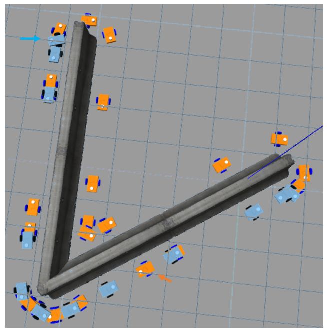
For a sampling frequency of $20Hz$ with a wall distance set to 0.5m the average distance error for a straight wall was 0.026m. PD tuning was conducted with the estimation of the average wall distance error for a straight section of the wall, presented in table II. Simultaneously, the published velocities were recorded and plotted in Fig. 7. The angular velocity gets saturated a lot more times for the case where the derivative control is switched on, Fig. 7, thus the chosen PD parameters were P=15 and D=0. As expected, the error increased with the introduction of corners in the wall contour. For scenes "V" and "W" the proposed algorithm is verified, as shown in Fig. 6.

VI. Discussion

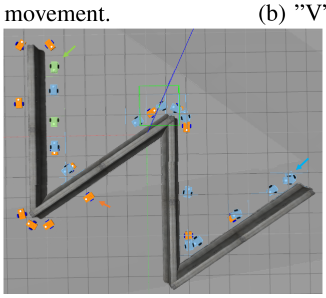
The proposed algorithm is capable of finding walls, thus in the simulations the robot will keep wondering until a wall is found. This approach may result in a very long waiting



(a) Random movement.



(b) ”V” world.



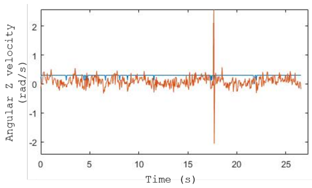
(c) ”W” world.

Fig. 6: Simulation in Gazebo and superposition of robot’s positions at different time intervals. fs= 20Hz; P=15; D=0; Pangle=1; dwall=0.5m. Orange: initial movement, arrow- starting position; Blue: movement on first inversion, arrow - inversion of direction; Green: Second movement inversion.

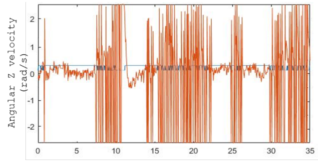
TABLE II: PD parameter tuning, average distance error.

PD specifications		AVG distance error	
		Wall at Left	Right
P = 15	D = 0	-0.0071	-0.0145
P = 15	D = 5	-0.0456	-0.0130

time, the converging time of the robot towards the wall could be increased by implementing a machine state in which the robots orientation would be aligned with the minimum wall distance angle, instead of doing proportional control at a high wall distance which saturates the angular response of the robot. An even smarter approach would be to replace the random movement with a circular movement with increasing radius. However, the proposed algorithm has some limitations. Namely, the proportional control sometimes gets saturated, thus it is limited to maximum angular velocities of [-2.5, 2.5] rad/s.



(a) P=15; D=0.



(b) P=15; D=5.

Fig. 7: Linear (blue) and angular (orange) velocities in m/s and rad/s, respectively, for each of the proportional control parameters.

VII. Conclusion

For a robot to detect correctly a wall and perform wall-following actions a sturdy wall-following method is necessary. Thus, in this document a wall-following method was proposed based in laser sensor inputs with data from the environment. The system has 20Hz sampling frequency, maximum linear velocity of 0.3m/s and angular velocity proportionally controlled for correct alignment with the wall. The goal of wall detection, random movement in the space until a wall is found and correct following of the wall was achieved. Future experiments can be performed in order to implement the system in a real environment, with a more appropriate response to environment’s noise or obstacle avoidance.

References

[1] Xin Wei, Erbao Dong, Chunshan Liu, et al. “A Wall-Following Algorithm Based on Dynamic Virtual Walls for Mobile Robots Navigation”. In: *2017 IEEE International Conference on Real-Time Computing and Robotics (RCAR)*. Okinawa: IEEE, July 2017, pp. 46–51. ISBN: 978-1-5386-2035-9. DOI: 10.1109/RCAR.2017.8311834.

[2] P. van Turennout, G. Honderd, and L. J. van Schelven. “Wall-Following Control of a Mobile Robot”. In: *Proceedings 1992 IEEE International Conference on Robotics and Automation*. May 1992, 280–285 vol.1. DOI: 10.1109/ROBOT.1992.220250.

[3] Yoshinobu Ando and Shin’ichi Yuta. “Following a Wall by an Autonomous Mobile Robot with a Sonar-Ring”. English. In: *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, 1995, pp. 2599–2606.

[4] Ricardo Carelli and Eduardo Oliveira Freire. “Corridor Navigation and Wall-Following Stable Control for Sonar-Based Mobile Robots”. In: *Robotics and Autonomous Systems* 45.3 (Dec. 2003), pp. 235–247. ISSN: 0921-8890. DOI: 10.1016/j.robot.2003.09.005.

[5] Juan Marcos Toibero, Flavio Roberti, and Ricardo Carelli. “Stable Contour-Following Control of Wheeled Mobile Robots”. en. In: *Robotica* 27.1 (Jan. 2009), pp. 1–12. ISSN: 1469-8668, 0263-5747. DOI: 10.1017/S026357470800444X.

[6] S. Charifa and M. Bikdash. “Adaptive Boundary-Following Algorithm Guided by Artificial Potential Field for Robot Navigation”. In: *2009 IEEE Workshop on Robotic Intelligence in Informationally Structured Space*. Mar. 2009, pp. 38–45. DOI: 10.1109/RIISS.2009.4937904.