

# SIMULATION AND ASSESSMENT EDUCATIONAL FRAMEWORK FOR MOBILE ROBOT ALGORITHMS

Submitted: 15<sup>th</sup> May 2013; accepted 5<sup>th</sup> July 2013

Nelson Munoz Ceballos, Jaime Valencia, Alessa Alvarez Giraldo

DOI: 10.14313/JAMRIS\_1-2014/7

## Abstract:

A mobile robot simulator useful in research and education was implemented in Matlab, it models the differential kinematics as well as proximity sensors of the robot. It allows the performance assessment of navigation algorithms through various quality metrics that are useful for comparing and analyzing navigation algorithms of mobile robots. An example that simulates and compares two autonomous navigation algorithms is presented.

**Keywords:** educational robotics, mobile robots, navigation algorithms, performance metrics

## 1. Introduction

The development of a robust autonomous navigation system for mobile robots is a broadly studied topic, and an open field for research. Those systems are continuously evolving and new approaches and applications are constantly emerging.

Different ways to address the robot navigation arise frequently, each of them with new valuable contributions. As time goes on, new problems and its possible solutions are studied according to the specific application.

Many methods have been used just to solve specific problems while their strengths or weaknesses are not completely understood. The comparison of algorithms with reference frames or standard procedures is usually a relegated task [1], [2].

This paper presents a framework for simulation and assessment of mobile robotics navigation algorithms, useful for teaching and research in robotics. In section 1, the simulator is described, in section 2, various performance metrics used in the navigation of mobile robots are defined, in section 3, two navigation algorithms are presented, in section 4, the process to be followed for assessment of algorithms is shown. Finally, in section 5, the conclusions are presented.

## 2. Mobile Robot Simulator

A graphic 2D simulator has been developed, it is useful for teaching and researching on navigation algorithms for mobile robots, it offers the possibility to evaluate the performance of the implemented naviga-

tion algorithm. The framework was carried out using Matlab, chosen for its potency to create mathematic programs, its easy use and the possibility of adding toolboxes as neural networks, fuzzy logic, etc.

The mobile robot simulator allows creating a robot, the environment obstacles as well as algorithms that process information of the robot state and act upon it. Furthermore, the framework includes a set of performance metrics [3], [14].

All the tasks in the simulator are carried out using graphical user interface and commands, some features are displayed in the Figures 1 and 2, as control algorithm window, generated path window and the buttons with functions to command the framework.

The user can design the own functions using the basic commands and Matlab programming environment. The framework can be run in various platforms (Windows, Linux, and MAC OS X).

As a result of the navigation algorithm performance testing, the framework returns the angles, velocities and some quality indexes of the generated

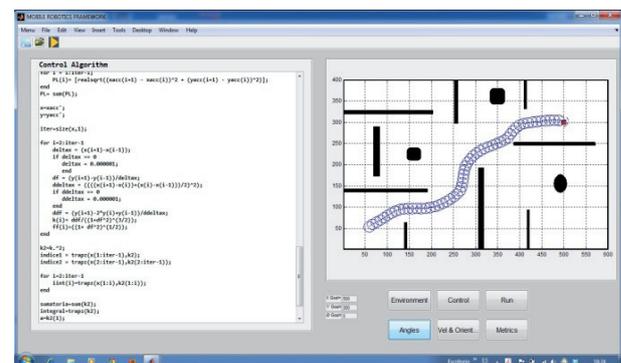


Fig. 1. Graphic user interface of the framework

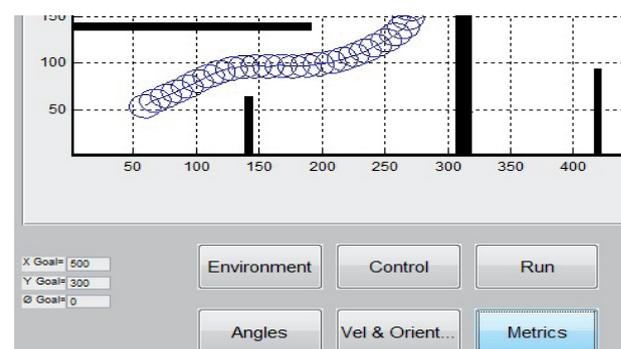


Fig. 2. Detail showing some framework features

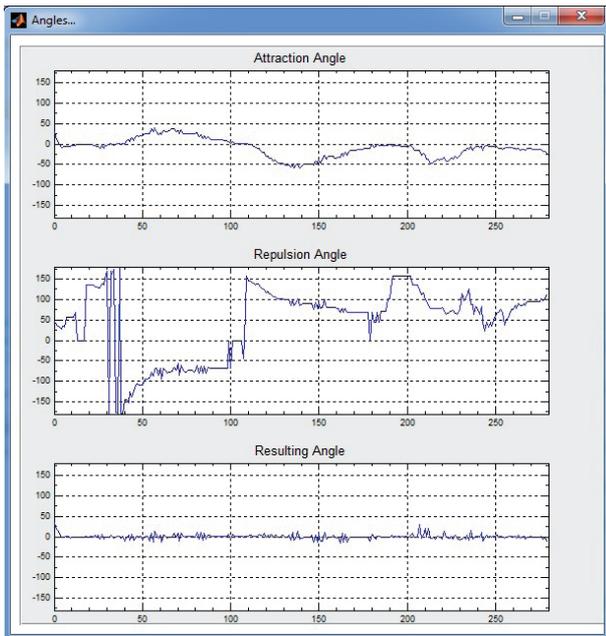


Fig. 3. Graphic record of the angles

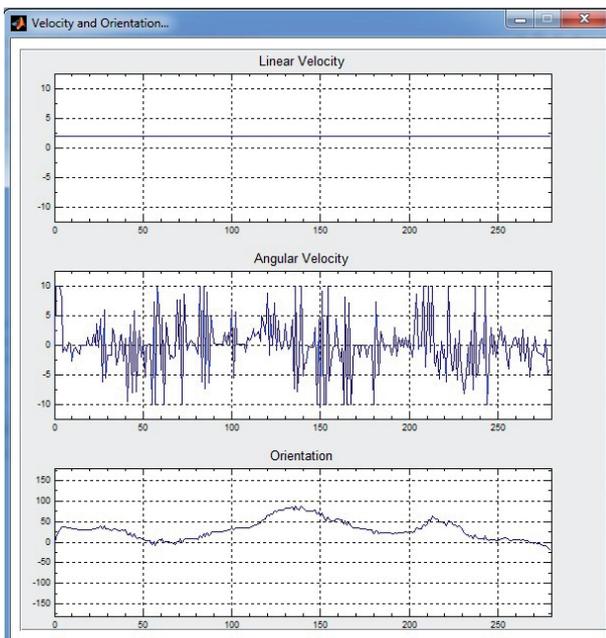


Fig. 4. Graphic record of velocity and orientation

path. Figure 3 displays the graphic record of the attraction, repulsion, and resulting angles of the navigation mission. Figure 4 displays the graphic record of the linear and angular velocities, as well as the robot orientation during the whole mission. Figure 5 displays the result of applying some performance metrics to the navigation algorithm.

2.1 Robot Environment

The environment is loaded in the simulator as an image. The white areas are understood as empty while the black areas are taken in as objects in the environment. The image boundaries are taken as walls.



Fig. 5. Metrics interface

2.2 Robot Movement

In order to simulate the robot movement, its kinematics should be taken into account which is subject to the next equations for a differential locomotive robot:

$$\begin{aligned} \dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= w \end{aligned}$$

Where  $\dot{x}$  and  $\dot{y}$  are the speed in the axes  $x$  and  $y$ ;  $\theta$  is the angle of the robot with the axis  $x$ ;  $v$ ,  $w$  are the linear and angular speed of the robot (movement and spin speed respectively). The linear and angular speeds are controlled by the navigation algorithm.

It is necessary to break down the previous expressions in differential equations to allow the computational estimate. In each sampling period  $T$ , the new  $x$  and  $y$  position regarding the center of the robot is calculated as well as its orientation, then the robot is drawn in that position.

2.3 Robot Sensors

Proximity sensors should be defined by the user and the amount can be set according to the user needs, indicating their position in the robot periphery, the opening angle and the scope of the sensor (Figure 6).

The distance measure is given in pixels and it is estimated taking the length between the point where the sensor is located and the closest point of any object within its detection scope.

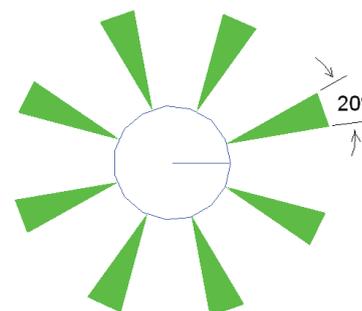


Fig. 6. Proximity sensors (opening angle: 20°, detection scope: 25 cm)

### 3. Performance Metrics on Navigation

There are various metrics that can be used to evaluate the performance of a navigation system, but none of them is able to indicate the quality of the whole system. Therefore it is necessary to use a combination of different indexes quantifying different aspects of the system. Having a good range of performance measurements is useful for: Optimizing algorithm parameters, testing navigation performance within a variety of work environments, making a quantitative comparison between algorithms, supporting algorithm development and helping with decisions about the adjustments required for a variety of aspects involved in system performance [13].

Navigation performance metrics can be classified in the following order of importance: i) Security in the trajectory or proximity to obstacles indexes, ii) metrics that consider the trajectory towards the goal and, iii) metrics that evaluate the smoothness of the trajectory.

#### 3.1. Security Metrics

These metrics express the robot security while it travels through a trajectory, taking into account the distance between the vehicle and the obstacles in its path [5].

Security Metric-1 (SM1): Mean distance between the vehicle and the obstacles through the entire mission measured by all the sensors; the maximum value will be produced in an obstacle free environment. If the deviation of the index from its maximum value is low, it means that the chosen route had fewer obstacles.

Security Metric-2 (SM2): Mean minimum distance to obstacles. This is taken from the average of the lowest value of the  $n$  sensors. This index gives an idea of the risk taken through the entire mission, in terms of the proximity to an obstacle. In an obstacles free environment  $SM1 = SM2$  is satisfied.

Minimum Distance (Min): Minimum distance between any sensor and any obstacle through the entire trajectory. This index measures the maximum risk taken throughout the entire mission.

#### 3.2. Dimension Metrics

The trajectory towards the goal is considered in its time and space dimensions. In general, it is assumed that an optimal trajectory towards the goal is, whenever possible, a line with minimum length and zero curvature between the initial point  $(x_i, y_i)$  and the final point  $(x_n, y_n)$ , covered in the minimum time.

Length of the Covered Trajectory ( $P_L$ ) is the length of the entire path covered by the vehicle from the initial point to the goal. For a trajectory in the  $x$ - $y$  plane, composed of  $n$  points, and assuming the initial point as  $(x_i, f(x_i))$  and the goal as  $(x_n, f(x_n))$ ,  $P_L$  can be calculated as:

$$P_L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (f(x_{i+1}) - f(x_i))^2} \quad (1)$$

Where  $(x_i, f(x_i))$ ,  $i = 1, 2, \dots, n$  are the  $n$  points of the trajectory in Cartesian coordinates [6].

The length of a trajectory given by  $y = f(x)$ , in the  $x$ - $y$  plane between the points  $(a, f(a))$  and  $(b, f(b))$ , can also be calculated as [10]

$$P_{Lapprox} \cong \int_a^b \sqrt{1 + (f'(x_i))^2} dx \quad (2)$$

Mean distance to the goal (Mgd): This metric can be applied to robots capable of following reference trajectories. An important aspect when determining the quality of the robot navigation system is the ability to follow a trajectory that aims to reach a goal, so, to evaluate the quality of the execution of the trajectory, the mean distance between the vehicle and goal is analyzed. The difference is more significant if the covered distance is shorter [9]. The mean distance to the goal is defined by the square of the proximity to the goal distance  $l_n$ , integrated across the length of the trajectory and normalized by the total number of points  $n$ :

$$l_n = \min \left( \forall n \left( \sqrt{(x_i - x_n)^2 + (f(x_i) - f(x_n))^2} \right) \right) \quad (3)$$

$$Mgd = \frac{\int_0^l l_n^2 ds}{n} \quad (4)$$

Control Periods (LeM): It is the amount of control periods. This metric relates to the number of decisions taken by the planner to reach the goal, if the robot moves with lineal and constant speed ( $v$ ). This gives an idea of the time needed to complete the mission [5].

#### 3.3. Smoothness Metrics

The smoothness of a trajectory shows the consistency between the decision-action relationship taken by the navigation system, as well as the ability to anticipate and to respond to events with enough speed [9]. The smoothness of the generated trajectory is a measure of the energy and time requirements for the movement; a smooth trajectory translates into energy and time savings [4]. Additionally, a smooth trajectory is also beneficial to the mechanical structure of the vehicle.

Bending Energy ( $B_E$ ): This is a function of the curvature,  $k$ , used to evaluate the smoothness of the robot's movement. For curves in the  $x$ - $y$  plane, the curvature,  $k$ , at any point  $(x_i, f(x_i))$  across a trajectory is given by:

$$k(x_i, f(x_i)) = \frac{f''(x_i)}{(1 + (f'(x_i))^2)^{\frac{3}{2}}} \quad (5)$$

The bending energy can be understood as the energy needed to bend a rod to the desired shape [11].  $B_E$  can be calculated as the sum of the squares of the curvature at each point of the line  $k(x_i, f(x_i))$ , along the length of the line  $L$ . So, the bending energy of the tra-

jectory of a robot is given by:

$$B_E = \frac{1}{n} \sum_{i=1}^n k^2(x_i, f(x_i)) \quad (6)$$

Where  $k(x_i, f(x_i))$  is the curvature at each point of the trajectory of the robot and  $n$  is the number of points in the trajectory.

The value of  $B_E$  is an average and does not show with clarity enough that some trajectories are longer than others. Therefore,  $TB_E$  can be used instead; this metric takes into account the smoothness and length of the trajectory simultaneously.

$$TB_E \text{ is defined by } TB_E = \int_a^b k^2(x) dx \quad (7)$$

$$\text{and numerically, } TB_E = \sum_{i=1}^n k^2(x_i, f(x_i)) \quad (8)$$

In a straighter trajectory, the values  $B_E$  and  $TB_E$  will be lower, which is desirable since the energy requirement is increased according to the increase in the curvature of the trajectory.

Smoothness of Curvature (Smoo) is defined by the square of the change in the curvature  $k$  of the trajectory of a vehicle with respect to the time, integrating along the length of the trajectory and normalized by the total time  $t$  [9].

$$Smoo = \frac{\int_0^t \left( \frac{dk}{dt} \right)^2 ds}{t} \quad (9)$$

## 4. Navigation algorithms

The navigation algorithms provide basic capabilities for the mobile robot, such as the ability to evade obstacles and to generate a trajectory towards a goal. (goal-seeking obstacle-avoidance).

### 4.1. Algorithm 1

This is a reactive algorithm based on a potential field method, which produces two different behaviors: first, goal attraction, and second, obstacles repulsion (keeping away from objects). The planning of the movement consists in the proper combination of both behaviors in such a way that the robot reaches the goal without collisions. This combination is achieved using a vector sum [7].

### 4.2 Algorithm 2

This algorithm is based on reactive behaviors, denominated AFREB "adaptive fusion of reactive behaviors" [12]. By using a neural network, an appropriate combination of the behaviors can be achieved, in such a way that the system is able to perform complex tasks, as navigation towards a goal, while evading obstacles in its path. The AFREB basically consists of the following modules: behavioral fusion, fusion supervisor, behavior primitives (1, 2, ..., n), and executor.

## 5. Simulations and Results

The simulation framework for comparing the performance of the algorithm 1 and algorithm 2 was used. This software enables teaching and researching in mobile robot navigation.

The robot simulated is Giraa02 [8], it has a differential locomotion system, 8 proximity sensors and odometry sensors; its diameter is 30 cm.

As didactic example two different scenarios are used to test algorithms. The environment is similar to offices, it means, A 6 m × 4 m frame, structured environment with static obstacles, some obstacle borders are sharp, there are also straight lines obstacles, and narrow zones, Figure 7.

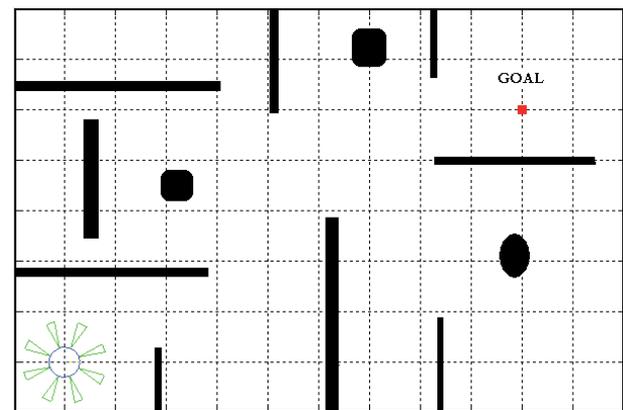


Fig. 7. Test Scenario for Mobile Robot Navigation

The metrics for autonomous navigation consider the security of the trajectory and measure the risk taken by the robot in its movement towards the goal, similarly measure aspects related to the planning of the trajectory and the quality of the trajectory according to the energy and time required for the movement.

For general purposes, only one metric is required for each one of the 3 categories described in section 2, but the use of various metrics helps to improve the analysis.

### 5.1. Simulations

The paths generated by the algorithms, in all scenarios are shown in Figures 8 and 9. Table 1 summarizes the results obtained from the simulation using both navigations algorithms according to the quality metrics described.

### 5.2. Analysis of Results

In scenario 1, the algorithm 1 uses less control periods, and consequently takes less time to complete the mission, and covers a safer and shorter path, the figure 8 shows that algorithm 1 produces a great orientation change for each control period. Algorithm 2 covers a smoother path, there is a smaller change in the orientation during each control period, resulting in energy saving and less structural stress on the robot.

From Table 1, it can be deduced that the difference between both algorithms in the trajectory and time taken is approximately 3.3% and 3.1% respectively. The robot programmed with algorithm 2 passed at

minimum 7 cm from any obstacle, it showed approximately 65% less bending energy than algorithm 1.

In scenario 2, the algorithm 1 uses more control periods, and consequently takes more time to complete the mission, it covers a safer and longer path. The Figure 9 shows that algorithm 2 covers a smoother path, there is a smaller change in the orientation during each control period, with consequent energy saving and less structural stress on the robot. Algorithm 2, makes the robot able to transit through narrow zones like corridors, keeping a safe distance from the obstacles and also generating smooth trajectories.

These results are an example to demonstrate this is a useful way to test robots navigation algorithms, but more test scenarios are necessary.

### 5.3 Other Features of the Simulation Framework

The mobile robot simulator is useful both to quantitatively compare navigation algorithms for robots and to observe the performance of the algorithms at different cases of study e.g. the problem of local minimums.

Both of the studied algorithms have movement planning strategies based in sensors, with local obstacles avoidance. These features imply the local minimums problem, which occur when the robot is navigating to a goal but gets trapped oscillating in the presence of obstacles in an area with the exit direction opposite to the goal, or when the movement results in a direction getting away to the goal.

The described situations create a conflict in the reactive behavior commanding the robot navigation, Figure 10. The simulator evidences that the problem is more noticeable in algorithm 1, because the navigation direction is a result only of the vector sum of the attraction potential to the goal, and the repulsion potential, may enter in a local minimum when the robot navigated in a direction getting away of the goal.

In Figures 11 and 12, the navigation mission is similar to that in scenery 2, it implies the movement from the point (50,350) to the point (160,175), and the goal is marked with a red point, which is 45 cm away from the original goal in Figure 9. This slight modification causes makes the robot with algorithm

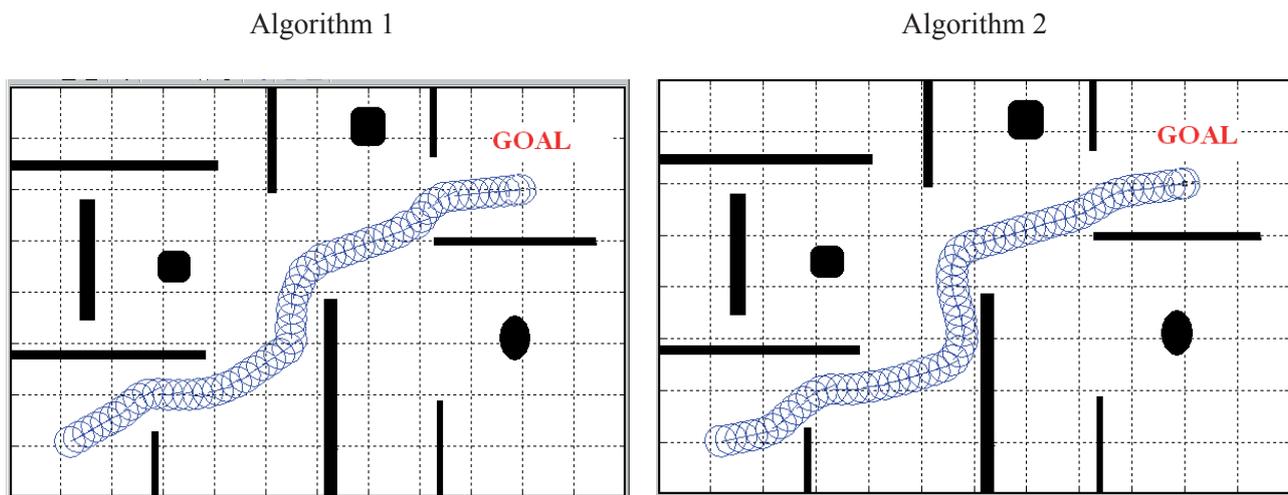


Fig. 8. Paths generated by the control algorithms: Start point (50,50), Goal (500,300)

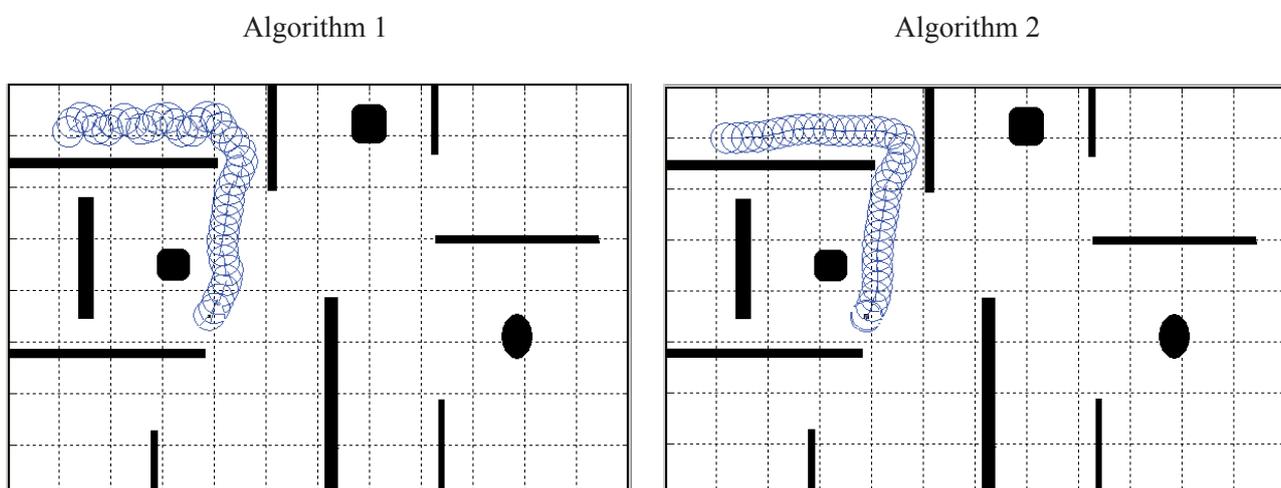


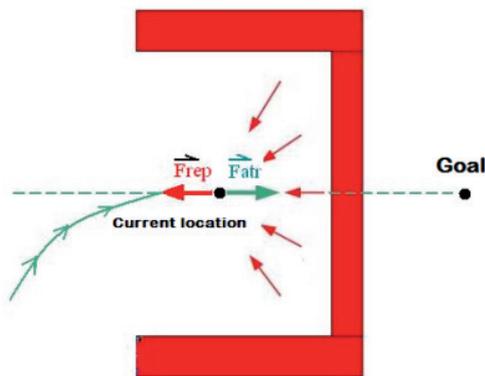
Fig. 9. Paths generated by the control algorithms: Start point (50,350), Goal (195,175)

**Table 1. Robot performance**

Metric	SM1 [cm]		SM2 [cm]		Min [cm]		PL [cm]		LeM		TB <sub>E</sub>	
	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2
Scenario 1	26.1	25.6	18.3	17.3	11	7	562.7	581.9	283	292	0.2463	0.0846
Scenario 2	25.0	24.4	13.0	12.4	7	3	395.7	359.9	199	181	0.4007	0.0140

SM1 maximum = 26.5 cm

1 stay trapped and the attraction and repulsion potentials are in conflict. Algorithm 2 achieves a satisfactory performance because the goal is located in a direction not totally opposed to the movement direction and the behaviors as searching of free areas and line following sum in the movement direction, allowing the robot exit this area and arrive to the goal.

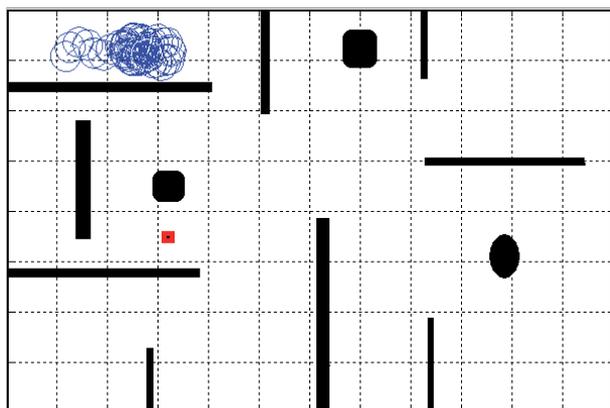


**Fig. 10. Local minimum example [7]**

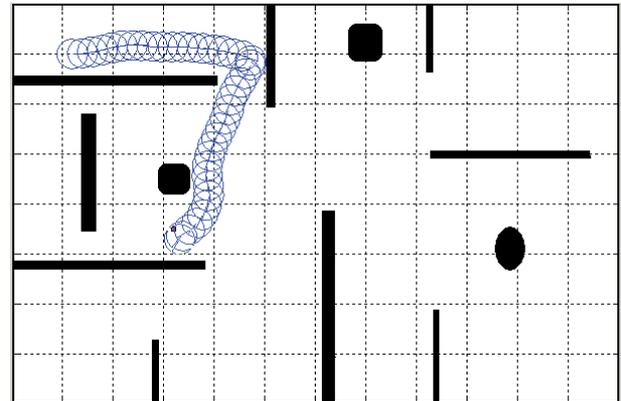
In [15] and [16] there are some research works about local minimum and solving alternatives.

**5. Conclusions**

This paper describes a framework with several performance metrics, useful to compare mobile robots navigation algorithms including safety, dimension and smoothness of the trajectory. The suggested metrics are quite straight forward. However, it was



**Fig. 11. Path generated by algorithm 1**



**Fig. 12. Path generated by algorithm 2**

shown that they can be used together to systematize simulated or experimental studies on control algorithms for mobile robot navigation.

A simple didactic example was presented. The obtained results demonstrate the need to establish a procedure that can be used to analyze and compare navigation algorithms for mobile robots using several performance metrics. This is an open topic of research. It has become necessary to establish proper approaches and benchmarking procedures, for example, using a benchmarking standard framework for navigation algorithm and performance evaluation.

This metrics can be applied in simulated environments, but the performance metrics evaluation is more important in real environments. Many of the challenges in robot navigation come from the challenges of real environments, such as uncertainty in the sensors and the errors as odometry, which are generally not considered in simulation.

**Acknowledgements**

This work was partial result of the 2061080241 project. The authors would like to thank to ITM, UdeA, and Polytechnic Jaime Isaza Cadavid.

**AUTHORS**

**Nelson Munoz Ceballos\*<sup>1,2</sup>, Jaime Valencia<sup>3</sup>, Alessa Alvarez Giraldo<sup>1,3</sup>** –

<sup>1</sup> Colombian Politechnic Jaime Isaza Cadavid,

<sup>2</sup>Metropolitan Technological Institute

<sup>3</sup>University of Antioquia, Medellín-Colombia

\*Corresponding author. E-mail addresses: ndmunoz@elpoli.edu.co, nmunoz345@gmail.com

## REFERENCES

- [1] J. Evans, E. Messina, "Performance Metrics for Intelligent Systems". In: *Proceeding of the Performance Metrics for intelligent Systems Workshop*, Gaithersburg, MD, 14–16 August 2000.
- [2] S. Wong, L. Middleton, B. MacDonald, "Performance metrics for robot coverage task". In: *Proceedings Australasian Conference on Robotics and Automation ACRA*, Auckland, New Zealand, 2002, 7–12.
- [3] N. Muñoz, J. Valencia, N. Londoño, "Evaluation of Navigation of an Autonomous Mobile Robot". In: *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*, Washington, DC, EEUU, 2007.  
[http://www.nist.gov/el/isd/ks/upload/PerMIS\\_2007\\_Full\\_Proceedings.pdf](http://www.nist.gov/el/isd/ks/upload/PerMIS_2007_Full_Proceedings.pdf)
- [4] S. Dongqing, *Aerial robot navigation in cluttered urban environments*, PhD Thesis, The Florida State University, Florida, USA, 2006, 87.
- [5] J. Álvarez, *Planificación del movimiento de vehículos autónomos basada en sensores*. PhD Thesis, University of Oviedo, Oviedo, Spain, 1998, p. 178,
- [6] Y. Guo, J. Wang, "A new performance based motion planner for nonholonomic mobile robots". In: *Proceedings of the 3<sup>rd</sup> performance metrics for the Intelligent Systems Workshop (PerMIS'03) NIST*, Gaithersburg, MD, September 2003.
- [7] J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 4<sup>th</sup> Edition, Boston, 1996.
- [8] N. Muñoz, C. Andrade, N. Londoño, *Diseño y construcción de un robot móvil orientado a la enseñanza e investigación*, Ingeniería & Desarrollo Ed. 9, 2006.
- [9] J. Rosenblatt, *DAMN: Distributed Algorithm for Mobile Navigation*. PhD. Thesis, Carnegie Mellon University Robotics Institute, Pittsburg, PA, 1997.
- [10] M. Selekwka, E. Collins, J. Combey, "Multivalued Versus Univalued Reactive Fuzzy Behavior Systems for Navigation Control of Autonomous Ground Vehicles". In: *Proceedings from the 17<sup>th</sup> annual Florida Conference on the Recent Advances in Robotics FCRAR2004*, 20 May 2004.
- [11] E. Aguirre, A. Gonzalez, "Fuzzy Behaviors for mobile robot navigation, design, coordination and fusion", *International Journal of Approximate Reasoning*, vol. 25, 2000, 255–289.
- [12] A. Zalzalá, A. Morris, "Neural Networks for Robotic Control", Ellis Horwood, 1996, p. 278
- [13] G. Cielniak, A. Treptow, T. Duckett, "Quantitative Performance Evaluation of a People Tracking System on a Mobile Robot". In: *Proceedings of the European Conference on Mobile Robots (ECMR)*, Ancona, Italy, 2005.
- [14] Nelson David Munoz Ceballos, Jaime Alejandro Valencia, Nelson Londono Ospina, "Quantitative Performance Metrics for Mobile Robots Navigation, Mobile Robots Navigation", Alejandra Barrera (Ed.), 2010. ISBN: 978-953-307-076-6, InTech, Available from: <http://www.intechopen.com/books/mobile-robots-navigation/quantitative-performance-metrics-for-mobile-robots-navigation>
- [15] C. Ordóñez, *Design and implementation of a limit cycle negotiation strategy for robot navigation*. MSc. Thesis, The Florida State University, 62 pp, Florida, EEUU, 2006.
- [16] L. Qing, W. Lijun, C. Bo, Z. Zhou, "An improved artificial potential field method for solving local minimum problem". In: *2<sup>nd</sup> International Conference on Intelligent Control and Information Processing (ICICIP)*, vol. 1 1, July 2011. ISBN: 978-1-4577-0813-8.