

A New Trajectory Optimization Approach for Safe Mobile Robot Navigation: A Comparative Study (Khepera II Mobile Robot)

Walid Ellili^(✉), Abdelfetteh Lachtar, and Mounir Samet

Laboratory of Electronics and Information Technology,
University of Sfax, ENIS, BP 1173, Sfax 3038, Tunisia
elliliwalid@gmail.com, lachtar.abdelfetteh@gmail.com,
mounir.samet@enis.rnu.tn

Abstract. Through the present paper, a new approach useful for solving the obstacle avoidance and trajectory optimization problems during robot navigation for certain tasks to be performed at minimum costs. In its real sense, the obstacle avoidance approach is based on the application of two fuzzy controllers, the first of which is designed to join the object, while the second is conceived to serve as an obstacle avoiding device. The trajectory optimization approach is based on the gradient method. Prior to implementing the solution on the real robot, the simulation has been integrated in an immersive virtual environment, for a more effective movement analysis and safer testing purposes. The study findings prove to reveal well that the proposed approach turns out to exhibit a good average speed and a satisfactory target-reaching success rate, while the optimization oriented gradient method has turned out to be rather efficient in respect of the genetic algorithm approach.

Keywords: Fuzzy controller · Mobile robot · Obstacle avoidance · Gradient method · Genetic algorithm · Optimization

1 Introduction

A moving robot is a vehicle equipped with a sensor enabling it to move in an appropriate environment to accomplish certain tasks [1]. A remarkable progress has been made in all areas of robotics: perception, environmental modeling, automatic control of actuators, motion planning as well as scheduling tasks [2]. Technological progress has turned out to be progressively needed, sometimes critically versatile in our life for many problems to be solved. Originally, automated systems have initially been designed to substitute human beings in performing tedious tasks, dangerous applications or accomplishing works that are beyond one's physical capacities [3]. Technological developments, particularly those relating to the fields of electronics and information technology, prove to contribute a great deal in enhancing the promotion of robotics, along with intelligent machines and devices.

On further equipping the system with extra perception abilities, action and decision executing capacities, roboticists are nowadays increasingly interested in providing mobile robots with extra autonomy in a bid to make them interact more easily with their environments [4]. Thus, mobile robots have more recently been conceived to move more independently around rather structured environments.

Noteworthy, however, certain problems do actually persist in the robotics field, namely, the reactive obstacle avoidance to reach a predefined target in advance [5], as well as the performance of such a task at a minimal cost. In this respect, the present work is focused on investigating the robot navigation problem in presence of environmental obstacles, while optimizing the obtained trajectories. Actually, the navigation strategy rests on two types of fuzzy controllers. First, in case of inexistent obstacles, the fuzzy controller will monitor the robot to return to the desired configuration. Second, in case of obstacle existence, the controller charged with avoiding obstacles will be activated. The resulting trajectories' optimization is maintained by means of the gradient algorithm. In a first place, detection is ensured by means of the robot's own sensors. In a second place, detection is maintained through the introduction of new infrared sensor. In a last stage, an implementation phase of the developed algorithms is implemented on the Khepera II robot, for the simulation results to be checked. For this sake, this work is structured as follows: Sect. 2 is designed to sum up the trajectory optimization methods, while Sect. 3 provides an overview of the materials applied. As for Sect. 4, it involves a presentation of the different approaches undertaken to preserve the obstacle avoidance process along with the trajectory optimization, followed by an evaluation stage maintained through simulation and experimentation, subjects of Sects. 5 and 6. Finally, Sect. 7, bears the major concluding remarks, while paving the way for prospective future work and new horizons.

2 Trajectory Optimization Framework

Among the optimization methods, one might well cite the genetic algorithms as well as the gradient one. To note, the genetic algorithm is used in a number of well' known undertaken projects. As part of the control process, genetic algorithms can be applied as powerful methods whereby optimal research could be implemented in areas the scope of which is, apriori, unknown, or for the sake of determining the relevant effective control strategies within a complex environment [7]. Actually, the genetic algorithms' interest resides essentially in the fact that they allow for searching the most optimum solutions in areas whose extent is unknown, thus, allowing for critical processes to be attained within a large number of situations. In this respect, Genetic algorithms constitute a set of procedures that may stand as natural selection mechanisms. Their basic fundamental principle lies in simulating natural evolution processes within a hostile environment. In so far as the present work is concerned, the gradient algorithm has been applied as an optimization algorithm, whose principle consists in starting from a random point prior to moving towards the steeper inclination direction [8]. On applying a number of iterations, the algorithm converges up to a point which constitutes an the criterion extremum to be minimized. In this way, the gradient algorithm consists in

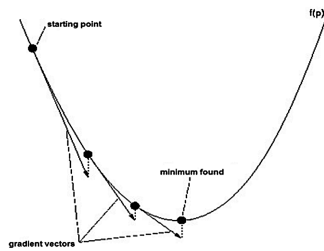


Fig. 1. The gradient descent principle, relevant to the function-to-variable case.

moving from a starting point (see Fig. 1) or one iteration u_k following the maximum slope line, as associated with the cost function f .

The descent direction, corresponding to this particular greatest slope outcome u_k line provided by the gradient $\frac{\partial f}{\partial u_k}$.

A further iteration can be even rendered by:

$$U_{K+1} = U_K - \varepsilon \frac{\partial f}{\partial u_k} \quad (1)$$

Where ε stands for a fixed positive parameter.

The algorithm appears to converge whenever the $\frac{\partial f}{\partial u_k}$ gradient approaches zero. Hence, the algorithm turns out to involve the following steps: Initialization, the beginning of the optimization loop (iteration k), the state equations' resolution, the cost function computation, cost function gradient calculation, Computation of the new vector relevant control variables, the optimization loop end.

3 Hardware Architecture

3.1 The Khepera II Platform Description

The robot subject of study application is the khepera II mobile robot. It is a two-wheel equipped mobile vehicle, developed by the Autonomous Systems Laboratory of the Federal Polytechnic School of Lausanne (EPFL). It has been the outcome and subject of several elaborated projects.

3.2 Detection Tools

To acquire an autonomous feature, the mobile robot must fulfill a number of capabilities. It must be able to sense its environment and to be located in it. For this purpose, special sensors, such as sonars, must be installed. They consist in particular devices useful for measuring distances separating it to any nearby obstacles. The notion of perception [9] in mobile robotics is related to the robot's ability to receive such necessary information as process and format for the robot to act and react suitably well in compliance with the

surrounding environment. For this reason, we try hard to get as much information as possible on the environment for the robot behavior to be adequately adjusted. More particularly, some interesting information turns out to be essential to apply, namely, data concerning the robot distance relative to a wall or to an object on the ground. Once the distance is recognized, the robot should be able to move from one point to another through retrieving effective ways to avoid collisions with the existing obstacles. For this sake, the khepera II robot has been equipped with eight infrared sensors whose detection scope does not exceed five centimeters. To solve the problem of detectors' short scope, we considered it essential to add an extra sensor, in an attempt to further increase the detection range.

4 Applied Methods

The robot is assimilated to a particle moving along the x and y coordinates in the reference frame, in accordance with an angle orientation, as shown in the following figure (Fig. 2):

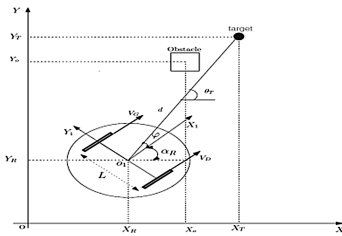


Fig. 2. Robot navigation area with an obstacle.

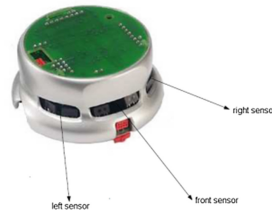


Fig. 3. Arrangement of the three obstacle-detecting sensors.

4.1 Obstacle Avoidance

A fuzzy logic method has been selected to ensure the mobile robot's navigation and acquire the obstacle avoiding behavior. The obstacle is assumed static and square in shape with a 5 cm side. According to this framework, the robot is equipped with three sensors: a frontal sensor, along with two other ones placed on both sides (right and left) (as shown in Fig. 3). These sensors are responsible for detecting the obstacle in the three indicated directions. The approach is based on application of two fuzzy controllers. While the first is designed to join the object, the second is conceived to serve as an obstacle avoiding device (Fig. 4).

The applied controller is the Takagi-Sugeno one, of order 0 [10].

Once entries are recognized, the outputs will be determined on the basis of the fuzzy rules' pertinent degrees. This controller can be activated only if the distance separating the obstacle to the robot is inferior to the distance covered by the robot sensors' range. The controller input allocating variables: For the designed controller,

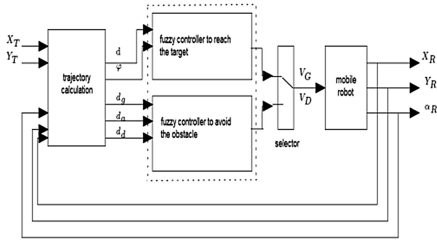


Fig. 4. Control chart with an obstacle avoidance device.

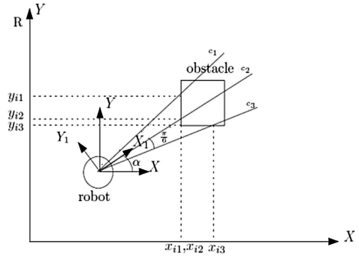


Fig. 5. Obstacle as detected by the robot's three sensors.

we have chosen as input variables the right side distance d_d , the left side one d_g , as well as the distance separating the robot to the obstacle front side d_a . Figure 5, below, displays the approach graphic presentation.

With: c_1 , c_2 and c_3 denoting the lines respectively passing through each of the left, front and right sensors' center. x_{i1} , x_{i2} , x_{i3} , y_{i1} , y_{i2} and y_{i3} : are, respectively, the X and Y coordinates of the straight lines intersection points' d_g , d_a and d_d with the obstacle. If the obstacle is placed in front of the robot, it will be simultaneously detected by the three sensors so that the distances d_g , d_a and d_d separating the robot and the obstacle would stand as the fuzzy controller's inputs, allowing the robot to avoid the obstacle. The expressions corresponding to the three distances are given by:

$$d_g = \frac{1}{\tan(\alpha_R + \frac{\pi}{6})} (Y_{i1} - b_1) \tag{2}$$

$$d_a = \frac{1}{\tan \alpha_R} (Y_{i2} - b_2) \tag{3}$$

$$d_d = \frac{1}{\tan(\alpha_R - \frac{\pi}{6})} (Y_{i2} - b_2) \tag{4}$$

With

$$b_1 = Y_R - \tan(\alpha_R + \frac{\pi}{6})X_R \tag{5}$$

$$b_2 = Y_R - \tan(\alpha_R)X_R \tag{6}$$

$$b_3 = Y_R - \tan(\alpha_R - \frac{\pi}{6})X_R \tag{7}$$

(X_R , Y_R) stands for the robot x and y coordinates. At this level, the discourse input universe is divided into three pertinent functions. The controller outputs correspond to both of the right and left wheels' respective speeds. The fuzzy controller which serves

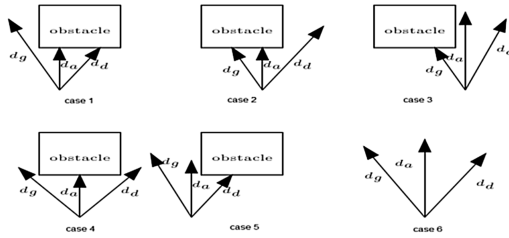


Fig. 6. Different obstacle detection cases as ensured through the three sensors.

to avoid the obstacle is activated once one of the three distances d_g , d_a and d_d proves to be inferior to the robot’s sensors’ range. Figure 6, below, depicts the obstacle detection capabilities via the three sensors.

Case 1: Obstacle detection case via both of the front and right sensors only. Case 2: obstacle detection case exclusively maintained by the front and left sensors. Case 3: Obstacle detection exclusively through the left sensor. Case 4: The obstacle is detected exclusively by the front sensor. Case 5: The obstacle is detected exclusively by the right sensor. Case 6: No obstacle is being detected. All possibilities have been studied for the purpose of allowing the robot to navigate and avoid obstacles without difficulty. The range reach of the robot sensors ranges between 4 and 5 cm. As such result, the robot must approach the obstacle closely to be able to detect it. Such a situation is likely to result in sudden changes to occur in the robot’s speed. To overcome this problem, we have envisaged adding an extra infrared sensor whose range could reach 30 cm. Indeed, this sensor would have the capacity and advantage of detecting the obstacle from a farther distance and, thereafter, avoiding sudden speed changes on the one hand, and maintaining smoother and more optimal trajectories, on the other. Both of the d_g and d_d inputs’ discourse universe remains unchanged (as it has been) i.e. it varies between 0 and 50 mm, whereas the previously scored distance [0; 50] has grown to reach the extent of [0; 300], thanks to the new sensor’s range, which is able to reach 300 mm. These three inputs are divided into three membership functions of a Gaussian type.

4.2 Trajectory Optimization

The criterion consists in minimizing the distance separating the target position to that of the next sample step.

$$J = \frac{1}{2} [(X_R - X_T)^2 + (Y_R - Y_T)^2] \tag{8}$$

With: (X_R, Y_R) : designating the robot’s current position, (X_T, Y_T) : the target’s position. Each of the fuzzy controller’s output, or VG VD, is calculated in terms of the expressions 9 and 10. Noteworthy, however, is that the fuzzy inference table’s adjustment appears to cater exclusively for the fuzzy rules’ pertinent conclusions, which can be accomplished via the following equations:

$$C_{Di}^{mouv} = C_{Di}^{anc} - \varepsilon \frac{\partial J}{\partial C_{Di}} \tag{9}$$

$$C_{Gi}^{mouv} = C_{Gi}^{anc} - \varepsilon \frac{\partial J}{\partial C_{Di}} \tag{10}$$

The derived criterion relating to the fuzzy rules' conclusions are provided by:

$$\frac{\partial J}{\partial C_G} = \frac{\partial J}{\partial X_R} \frac{\partial X_R}{\partial V_G} \frac{\partial V_G}{\partial C_G} + \frac{\partial J}{\partial Y_R} \frac{\partial Y_R}{\partial V_G} \frac{\partial V_G}{\partial C_G} \tag{11}$$

$$\frac{\partial J}{\partial C_G} = T^2 \frac{V_G + V_D}{2L} \frac{\alpha_i}{\sum_{j=0}^r \alpha_j} [(X_R - X_T) \sin \alpha - (Y_R - Y_T) \cos \alpha] \tag{12}$$

$$\frac{\partial J}{\partial C_D} = \frac{\partial J}{\partial X_R} \frac{\partial X_R}{\partial V_D} \frac{\partial V_D}{\partial C_D} + \frac{\partial J}{\partial Y_R} \frac{\partial Y_R}{\partial V_D} \frac{\partial V_D}{\partial C_D} \tag{13}$$

$$\frac{\partial J}{\partial C_D} = T^2 \frac{V_G + V_D}{2L} \frac{\alpha_i}{\sum_{j=0}^r \alpha_j} [(X_R - X_T) \sin \alpha - (Y_R - Y_T) \cos \alpha] \tag{14}$$

5 Simulation Results

The simulations have been undertaken within a Matlab environment for the sake of testing the controller's ability to reach the designed target, while avoiding the encountered obstacle.

5.1 Obstacle Avoidance

In this respect, and regarding different robot positions, several obstacle locations and different targets, have been set up in advance. The relevant simulation results are depicted in Figs. 7, 8 and 9, below.

Based on Figs. 7, 8 and 9, one can note that starting from any location point, the robot turns out to be able to detect the obstacle and avoid it during navigation, by acting on both of the right and left wheels' respective speeds. Indeed, the actual improvement has been observed on the two speeds' pertaining values on approaching the obstacle. In Fig. 9, for instance, the right wheel's speed has been greater than that of the left wheel for the robot to move away rightwards from the obstacle and achieve the set target. For the purpose of assessing the validity of these achieved findings, a robustness test concerning the controller's performance has been undertaken, to evaluate its response to any change in either the initial position or in the target one. The obtained results appear in Fig. 10, below.

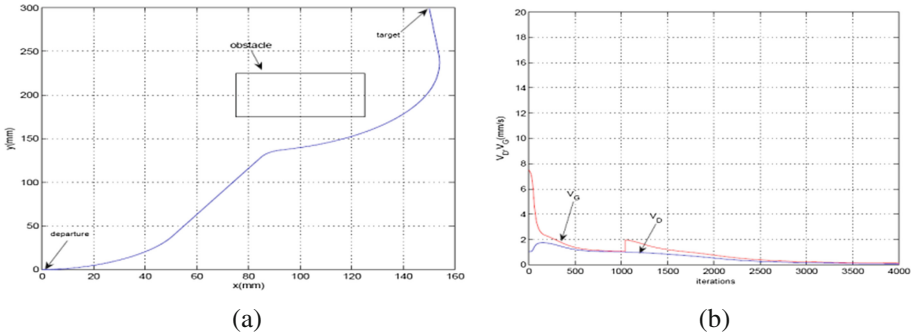


Fig. 7. (a): Simulation results for an obstacle placed at a level of (100, 200) and a target at (150, 300); (b): The robot's left and right wheels' corresponding speeds.

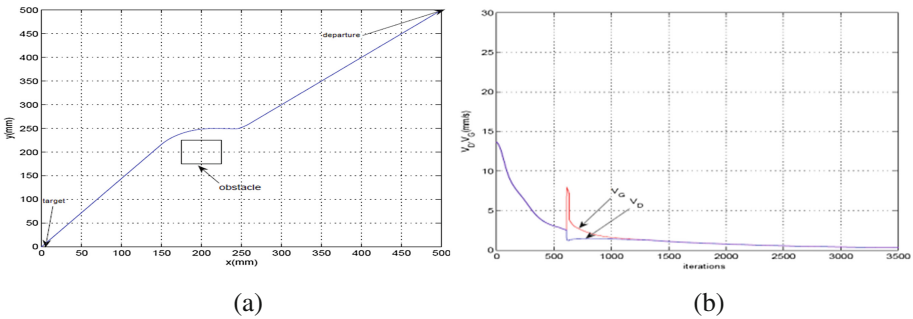


Fig. 8. (a): Simulation results for an obstacle placed at the (200, 200) level and a target at (0, 0) and a starting point of (500, 500); (b): The robot's left and right wheels' respective speeds.

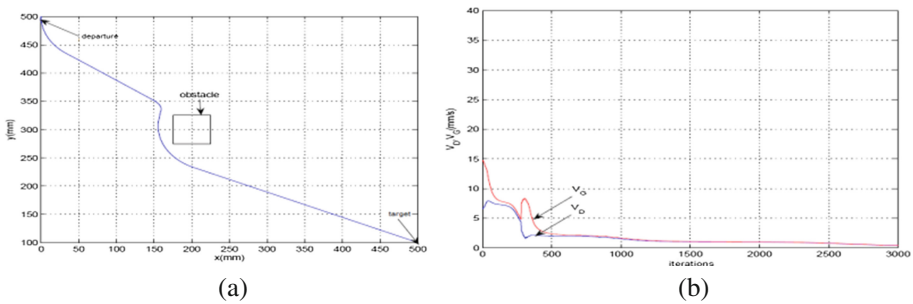


Fig. 9. (a): Simulation result relevant to an obstacle placed at (200, 300), (b): The robot's left and right wheels' respective speeds.

At this level, the developed fuzzy controller proves to fit perfectly well for ensuring the mobile robot's motion from any initial position to any desired position, without hurting any of the (avoided) obstacles. Figure 11, below, indicates the new sensors

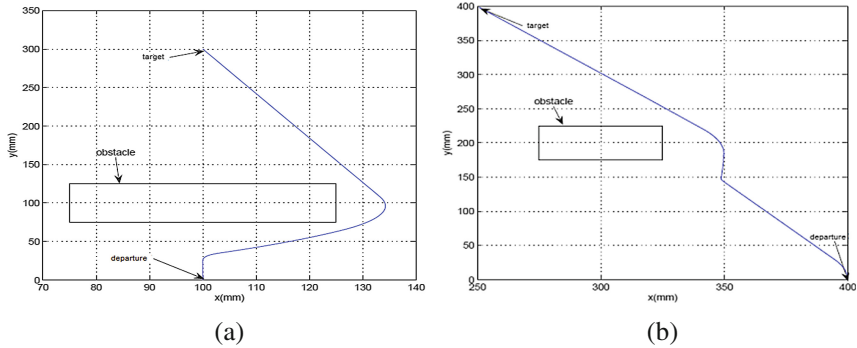


Fig. 10. (a): Simulation result relevant to an obstacle placed at point (100, 100); (b): Simulation Results relevant to an obstacle placed at point (300, 200).

achieved simulation results in respect of those reached by means of the khepera II robot’s proper sensors. Figure 11 reveals well that the robot’s navigation performance has recorded an improvement as the distance traveled starting from the same initial conditions has been noticeably reduced with application of the new sensor. The following table depicts a comparison of the distances traveled with and without introduction of the new sensor (Table 1).

Relying on this table, one can well deduce that the new sensor is suitably fit for the application presented regarding each of the treated cases. Actually, the distance traveled proves to decrease following implementation of this newly-devised sensor, as it appears to help significantly in detecting the obstacle at a farther distance, by pursuing the most optimal path.

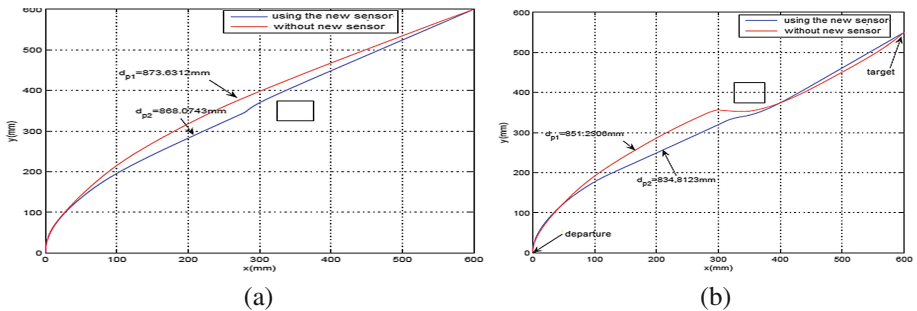


Fig. 11. (a), (b): Obstacle avoidance without and with the newly developed sensor, dp1: distance without application of the new sensor, and dp2: distance using the new sensor.

5.2 Trajectory Optimization

The Fig. 12 pertinent curves reveal well that the optimized controller reached distances turn out to be lower than the pre-optimization attained ones.

Table 1. Distances traveled with and without the new sensor with regard to different conditions

Starting point (mm)	Target (mm)	Obstacle position (mm)	Distance traveled without the new sensor (mm)	Distance traveled with new sensor (mm)
(0; 0)	(600; 500)	(350; 350)	824,2705	809,5474
(0; 0)	(600; 500)	(350; 400)	804, 8216	801,7906
(0; 0)	(600; 600)	(350; 350)	865,8441	855,9001
(0; 0)	(600; 600)	(350; 350)	873,6312	868,0743
(0; 0)	(600; 550)	(350; 400)	851,2306	834,8123
(0; 0)	(700; 600)	(350; 400)	952,1903	933,9707

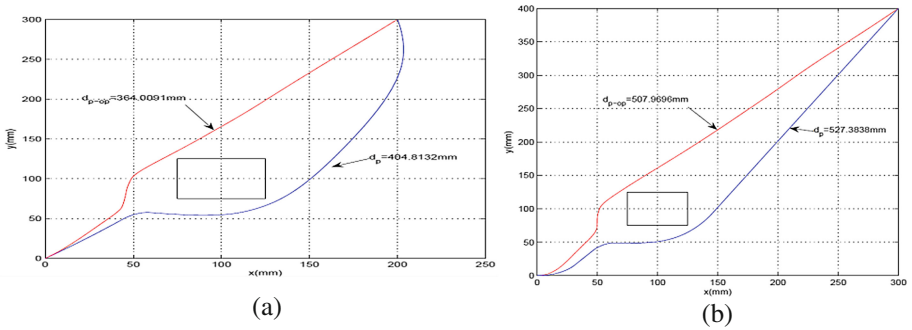


Fig. 12. (a): Represents the trajectory optimization regarding an obstacle set at (100, 100) and a target (200, 300), while (b) Denotes the trajectory optimization concerning an obstacle set at (100, 100) and a target (300, 400).

Table 2 depicts the gradient method as a rather efficient as compared to the trajectory optimization related genetic algorithm.

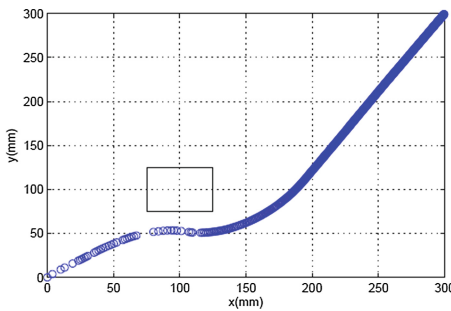
6 A Practical Implementation Case

The experiments implemented have been aimed to validate the obstacle avoidance applied approach. The mobile robot Khepera II departs from an initial configuration to reach a predefined goal, with a wooden cube being placed in front of it, which it has been able to detect by means of infrared sensors.

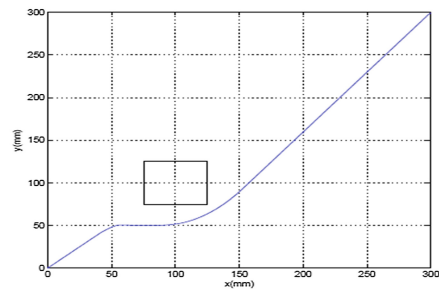
The following (Fig. 13) shows well that the robot khepera II has been capable of detecting the obstacle and avoiding it.

Table 2. Distances traveled with and without optimization

Starting point (mm)	Target (mm)	Obstacle position (mm)	Distance traveled without optimization (mm)	Distance traveled with optimization (Gradient method)	Distance traveled with optimization (Genetic algorithm)
(0,0)	(300,400)	(100,100)	527,3838	507,093	513,5699
(0,0)	(300,300)	(100,100)	441,3619	436,7311	440,9823
(0,0)	(200,300)	(100,100)	404,8132	364,0091	379,0217
(0,0)	(200,400)	(150,200)	458,0796	451,1089	455,1156
(0,0)	(100,200)	(100,100)	238,5747	225,5232	230,4522



(a)



(b)

Fig. 13. (a): Experimental Results regarding the obstacle avoidance process; (b): Simulation results concerning the obstacle avoidance process.

The results provided by both figures prove to be perfectly an significantly similar.

7 Conclusion

In the present work, a particular focus has been laid on investigating the obstacle-avoidance problem by means of fuzzy systems. In a first place, the Khepera II robot specific sensors have been deployed. Noteworthy, however, sudden changes have been noticed to prevail in the robot's both wheels' speeds. For this reason, a new extra sensor has been implemented with a relatively larger distance range. As a matter of fact, while the mobile robot's navigation and obstacle avoidance features have been based on the use of two controllers, monitoring and directing the robot, concerning the case in which no obstacle exists on the way to the goal, the additional controller has been devised to avoid any obstacle likely to occur on its path. Actually, the experimental implementation of a fuzzy controller in the mobile robot has led to a remarkable validation of the achieved simulation results. It is worth highlighting, in this respect that as the curves obtained is liable to optimization Distances traveled with Gradient

method and Genetic algorithm. In addition, we envisage elaborating on comparison in two methods; the Gradient method is advantage in Genetic algorithm.

References

1. Ye, J.: Tracking control of a nonholonomic mobile robot using compound cosine function neural networks. *Intell. Serv. Robot.* (2013). doi:[10.1007/s11370-013-0136-4](https://doi.org/10.1007/s11370-013-0136-4)
2. Cheng, P.-Y., Chen, P.-J.: Navigation of mobile robot by using D++ algorithm. *Intell. Serv. Robot.* **5**, 229–243 (2012). doi:[10.1007/s11370-012-0120-4](https://doi.org/10.1007/s11370-012-0120-4)
3. Schiffer, S., Ferrein, A., Lakemeyer, G.: CAESAR: an intelligent domestic service robot. *Intell. Serv. Robot.* **5**, 259–273 (2012). doi:[10.1007/s11370-012-0118-y](https://doi.org/10.1007/s11370-012-0118-y)
4. Ryu, H., Chung, W.K.: Local map-based exploration for mobile robots. *Intell. Serv. Robot.* **6**, 199–209 (2013). doi:[10.1007/s11370-013-0137-3](https://doi.org/10.1007/s11370-013-0137-3)
5. Hossain, A.: Undefined Obstacle Avoidance and Path Planning, American Society for Engineering Education (2013)
6. Hermawanto, D.: Genetic Algorithm for Solving Simple Mathematical Equality Problem. Indonesian Institute of Sciences (LIPI), Indonesia (2013)
7. Vandenberghe, L.: «Gradient method», EE236C (Spring 2016)
8. Bayle, B.: “mobile robotic”, Project at the National School of Physics of Strasbourg (2008)
9. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* **15**, 116–132 (1985)