

Research Article

Same Fuzzy Logic Controller for Two-Wheeled Mobile Robot Navigation in Strange Environments

Awatef Aouf^{1,2} Lotfi Boussaid^{1,2} and Anis Sakly³

¹National Engineering School of Sousse (ENISO), University of Sousse BP 264 Sousse Erriadh 4023, Sousse, Tunisia

²Laboratory E&E, Faculty of Sciences of Monastir, University of Monastir, Av. Ibn El Jazza Skanes 5019, Monastir, Tunisia

³Research Unit of Industrial Systems Study and Renewable Energy (ESIER), National Engineering School of Monastir (ENIM), University of Monastir, Av. Ibn El Jazza Skanes 5019, Monastir, Tunisia

Correspondence should be addressed to Awatef Aouf; aoufawatef@gmail.com

Received 24 September 2018; Revised 2 December 2018; Accepted 18 December 2018; Published 3 January 2019

Academic Editor: Shahram Payandeh

Copyright © 2019 Awatef Aouf et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For any mobile device, the ability to navigate smoothly in its environment is of paramount importance, which justifies researchers' continuous work on designing new techniques to reach this goal. In this work, we briefly present a description of a hard work on designing a Same Fuzzy Logic Controller (S.F.L.C.) of the two reactive behaviors of the mobile robot, namely, "go to goal obstacle avoidance" and "wall following," in order to solve its navigation problems. This new technique allows an optimal motion planning in terms of path length and travelling time; it is meant to avoid collisions with convex and concave obstacles and to achieve the shortest path followed by the mobile robot. The efficiency of employing the proposed navigational controller is validated when compared to the results from other intelligent approaches; its qualities make of it an efficient alternative method for solving the path planning problem of the mobile robot.

1. Introduction

A mobile robot is able to navigate intelligently in an uncontrolled environment without the need for physical or electromechanical guidance devices using different control techniques based on sensors. This autonomous agent is being increasingly used nowadays in various fields such as security, medicine, industry, space exploration, rescue operations, disaster relief, etc. The mobile robot is flexible enough to perform these tasks satisfactorily in both static and dynamic environments.

How to avoid obstacles during robot navigation, either global or local, is the main concern of robotics researchers. Global navigation means that the environment is completely known to the mobile robot. Different approaches have indeed been applied looking for a solution to global navigation problems, notably Artificial Potential Field [1], Grids [2], Visibility Graph [3], Cell Decomposition [4], and Voronoi Graph [5]. In local navigation, by contrast, the environment is completely or partially strange to the mobile robot which controls its motion using different equipped sensors. Various researches have been conducted so far to tackle the problem

of local navigation like Vector Field Histogram [6] which employs a two-dimensional Cartesian histogram grid as a world model. This model is updated continuously using onboard range sensors, Dynamic Window Approach [7]. This strategy is a local planner which calculates the optimal collision-free velocity for a mobile robot, Neural Network [8], Neurofuzzy [9], Particle Swarm Optimization [10], Genetic Algorithm [11], Ant Colony Optimization Algorithm [12], Cuckoo Algorithm [13], Simulated Annealing Algorithm [14], and Fuzzy Logic.

The fuzzy logic controller, first developed by Zadeh [15], is widely employed in various engineering applications including the mobile robotics.

Ren et al. [16] used the fuzzy logic techniques to solve navigation problems in a strange and changing environment. In [17], Pradhan et al. applied the fuzzy logic controller to ensure the navigation of one thousand robots in a wholly unknown environment. It was built using triangular, trapezoidal, and Gaussian membership functions. Then performances were compared, and Gaussian membership functions were proven to be more powerful and efficient. In their work, Yousfi et al.

[18] constructed a fuzzy logic controller to solve the navigation problems of the mobile robot. They developed the Gradient method to optimize the variables of the membership functions of the Sugeno fuzzy controller. In [19], authors presented a behavior-based control using fuzzy logic controller for mobile robot navigation in an obscure environment. They designed four basic tasks: go to goal behavior, obstacle avoidance behavior, tracking behaviour, and deadlock disarming behavior.

The navigation of the mobile robot based on equipped sensors using fuzzy logic controller is proposed in [20, 21]. Wu et al. [22] designed a fuzzy logic with genetic algorithm controller for mobile pipeline robot in a pipe environment. The fuzzy controller furnished the initial membership function and the fuzzy rules. The genetic algorithm took the best membership value for optimizing the fuzzy controller of path planning problems. In [23], Farooq et al. conducted a comparative study between Takagi-Sugeno and Mamdani fuzzy logic model for autonomous mobile robot navigation in unpredictable environments. They found out that Mamdani fuzzy model yielded better results and that Takagi Sugeno fuzzy model took less memory space in the real-time microcontroller implementation. Reference [18] used a mobile robot equipped with three sensors and designed a fuzzy logic controller which guarantees autonomous navigation in strange environment. In [24], the author developed a real-time fuzzy logic control to achieve obstacle avoidance in an unknown environment.

Several recent works in the literature are interested in solving the mobile robot navigation problem by using approaches based on fuzzy logic, but they did not focus on local minimum problem in unknown environment [25–27]. As a result, they could only handle simple environment (without deadlock) while standing helpless in more complex environment (with deadlock).

The contribution of our work is to construct a suitable architecture based on two behaviours and same fuzzy logic controller (S.F.L.C.) for solving the navigation problem in complex and unknown environments. This control architecture ensures a safe and short trajectory. A comparative study is made to illustrate the performances of our approach against PSO-PID [28], ABC-PID, and FA-PID controllers.

This paper has eight main sections. Section 2 introduces the kinematic modeling of the differential mobile robot. The proposed path planning algorithm is discussed in Section 3. Section 4 presents the PID controller and describes two evolutionary algorithms. Section 5 gives the details of the fuzzy logic controller. Section 6 presents the final results from different simulations. A comparative study is described in Section 7. Section 8 concludes and outlines the future of our work.

2. Kinematic Model of the Mobile Robot

In this work, we have used the differential mobile robot Khepera III to simulate the suggested navigation algorithms. Khepera III is equipped with nine infrared sensors used for distance measurements, two DC motors, and two encoders

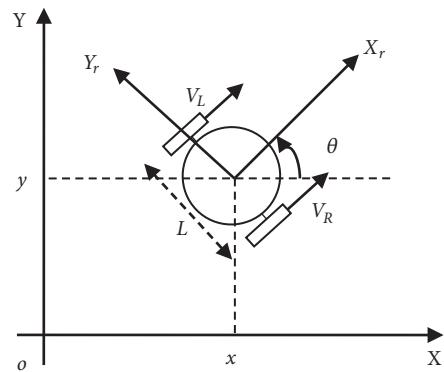


FIGURE 1: Schematic model of the mobile robot.

to give its real position in the environment. The schematic model of the robot is illustrated in the Figure 1.

The mathematical kinematic model of the two-wheeled mobile robot is given by the following equations:

$$\frac{dx}{dt} = \frac{V_R + V_L}{2} \cos \theta \quad (1)$$

$$\frac{dy}{dt} = \frac{V_R + V_L}{2} \sin \theta \quad (2)$$

$$\frac{d\theta}{dt} = \frac{V_R - V_L}{L} \quad (3)$$

where

(i) V_R and V_L represent, respectively, the right and left velocities of the two-wheeled mobile robot

(ii) θ is the orientation of the two-wheeled mobile robot

(iii) L is the distance between the two wheels of the robot

The kinematic motion of the two-wheeled mobile robot in the discrete time is described as follows:

$$x_{k+1} = x_k + T \frac{V_{R,k} + V_{L,k}}{2} \cos \theta_k \quad (4)$$

$$y_{k+1} = y_k + T \frac{V_{R,k} + V_{L,k}}{2} \sin \theta_k \quad (5)$$

$$\theta_{k+1} = \theta_k + T \frac{V_{R,k} - V_{L,k}}{L} \quad (6)$$

where T is the sampling time.

3. The Proposed Path Planning Algorithm

In this work, a behavior-based technique is proposed. This method combines two behaviors, namely, “go to goal_avoidance obstacle” and “wall following.” According to the information sent through the sensors in each step in the environment, an arbitration mechanism selects the convenient controller to offer the two-wheeled mobile robot a smooth motion.

Firstly, the wheeled mobile robot proceeds in a strange environment towards the fixed goal. In this case, the problem of the path planning does not arise. When the mobile robot

detects the presence of the obstacles, it should move far away without hitting with obstacles.

Once the robot enters into a deadlock, it will be trapped. In this case, the “wall following” controller interferes in order to allow the two-wheeled mobile robot to follow the boundary of the deadlock and take it out from the obstacle. In this point, the robot exists and ultimately starts to move toward the destination; “wall following” is not demanded anymore.

3.1. “Go to Goal_Obstacle Avoidance” Behavior. The aims of “go to goal_obstacle avoidance” controller are to ensure the safety of the mobile robot during its navigation in the unknown environment and to steer the robot to the desired goal.

There are many ways to avoid obstacles. In order to clarify the strategy used in our work, we need to follow the upcoming steps:

(i) Step 1: Transform the IR distance measured by each sensor to a point in the reference frame of the robot.

As shown in Figure 2, the point P_i measured by the sensor i can be written as a vector (in the reference frame of the sensor):

$$v_i = \begin{bmatrix} d_i \\ 0 \end{bmatrix}, \quad i = 1..9 \quad (7)$$

Use the location and the orientation of the sensor in order to transform this point in the reference frame of the robot. This transformation is defined by the following equation:

$$v'_i = M_t(x_{s_i}, y_{s_i}, \theta_{s_i}) \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (8)$$

where M_t is the transformation matrix that makes a translation by (x, y) and a rotation by θ :

$$M_t(x, y, \theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & x \\ \sin(\theta) & \cos(\theta) & y \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

(ii) Step 2: Transform the point P_i again to determine its location in the reference frame of the world. For that, we need the pose of the two-wheeled mobile robot (x, y, θ) in order to transform the robot from the robot’s reference frame to the world’s reference frame. This transformation is given by the following equation:

$$v''_i = M_t(x, y, \theta) v'_i \quad (10)$$

(iii) Step 3: Compute a vector u_i to each point from the robot as shown in Figure 3.

(iv) Step 4: Choose a weight α_i for each vector depending on how much importance you attach to a particular sensor to avoid obstacle.

The weighted vectors will be summed into a single vector u_{ao} as illustrated in Figure 4.

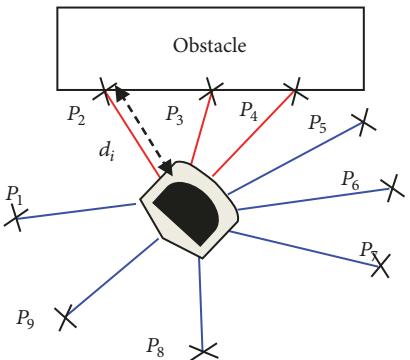


FIGURE 2: Transformation of IR distance in sensor’s frame.

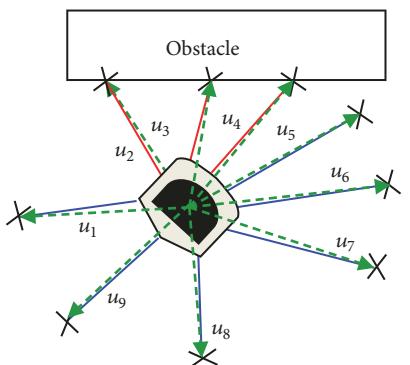


FIGURE 3: Computation of the vector u_i .

The value of this vector is given by the following equation:

$$u_{ao} = \sum_{i=1}^9 \alpha_i \times u_i \quad (11)$$

(v) Step 5: Employ the vector u_{ao} and the pose of the mobile robot to calculate the heading that steers the mobile robot away from any object.

$$\theta_{ao} = \tan^{-1} \frac{u_{ao}(y)}{u_{ao}(x)} \quad (12)$$

(vi) Step 6: Compute the vector u_{gtg} that points from the robot to the goal as shown in Figure 5.

The vector u_{gtg} is written as follows:

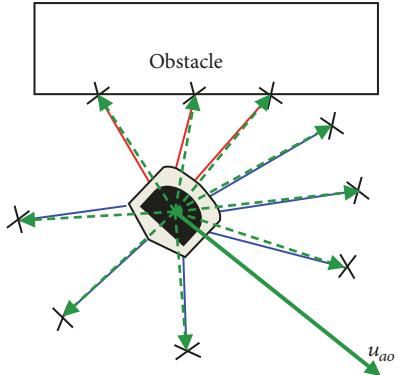
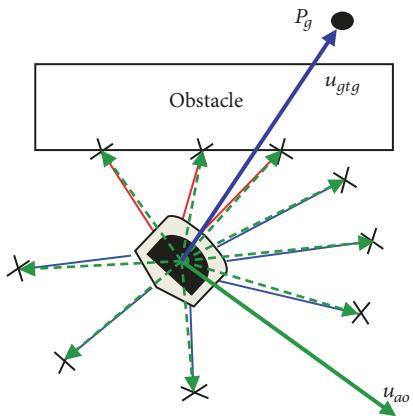
$$u_{gtg} = [x_{P_g} - x, y_{P_g} - y] \quad (13)$$

(vii) Step 7: Combine the two vectors u_{ao} and u_{gtg} into a single vector u_{gtg-ao} that points the wheeled mobile robot both away from any obstacle and towards the target as shown in Figure 6. It is given by the following equation:

$$u_{ao-gtg} = \alpha u_{gtg} + (1 - \alpha) u_{ao} \quad (14)$$

where α is the weight of the vector. It should be carefully tuned to get the best response.

(viii) Step 8: Use the vector u_{ao-gtg} and the pose of the mobile robot to compute the heading that steers the mobile

FIGURE 4: Computation of the vector u_{ao} .FIGURE 5: Computation of the vector u_{gtg} .

robot away from any obstacle and allows the robot to reach its target:

$$\theta_{ao-gtg} = \tan^{-1} \frac{u_{ao-gtg}(y)}{u_{ao-gtg}(x)} \quad (15)$$

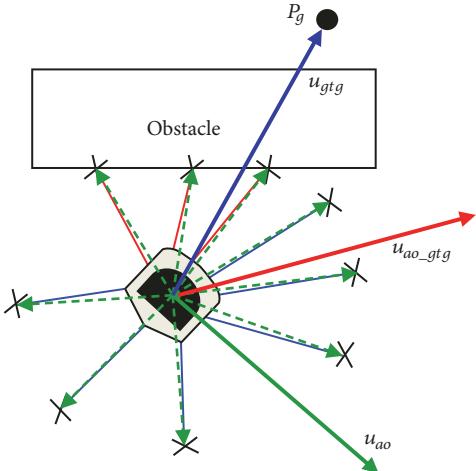
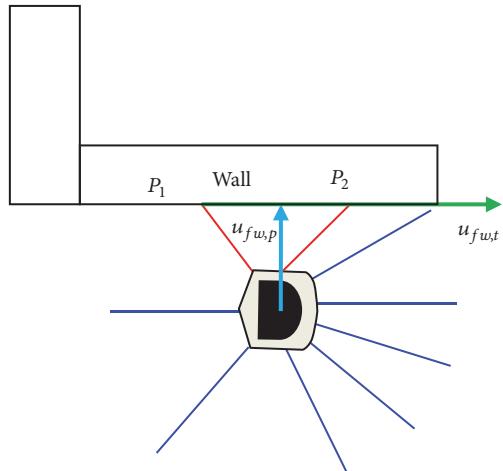
The input of the fuzzy controller of this behavior is the error (Φ_{ao-gtg}) between the angle of “go to goal and avoidance obstacle” (θ_{ao-gtg}) and the current heading of the robot (θ).

$$\Phi_{ao-gtg} = \theta_{ao-gtg} - \theta \quad (16)$$

The outputs of this controller are the linear (v) and angular velocity (w).

3.2. Wall Following Behavior. The mobile robot drives from a starting point to a target one without hitting with any obstacles. These obstacles are fairly easy if they are convex but rather hard to overcome when they are concave in Shape. If the robot starts from a point A to reach a point B existing behind the concave obstacle, the mobile robot will be trapped. To save the robot, we need to use another behavior known as the “wall following.” This behavior allows the robot to follow the contour thus avoiding the obstacle and reaching its target.

The algorithm of the wall following is explained in the following steps:

FIGURE 6: Compute of the vector u_{ao-gtg} .FIGURE 7: Computation of the vector $u_{fw,p}$.

(i) Step 1: Calculate a vector $u_{fw,t}$ that estimates a section of the wall, using the IR sensors of the mobile robot as shown in Figure 7.

The vector $u_{fw,t}$ is determined as follows:

$$u_{fw,t} = [P_2 - P_1] \quad (17)$$

$$u'_{fw,t} = \frac{u_{fw,t}}{\|u_{fw,t}\|} \quad (18)$$

(ii) Step 2: Compute a vector $u_{fw,p}$ that points from the mobile robot to the closest point on $u_{fw,t}$ as illustrated in Figure 8. It can be calculated as follows:

$$u_p = \begin{bmatrix} x \\ y \end{bmatrix}, \quad (19)$$

$$u_a = P_1$$

$$u_{fw,p} = (u_a - u_p) - ((u_a - u_p) u'_{fw,t}) u'_{fw,t} \quad (20)$$

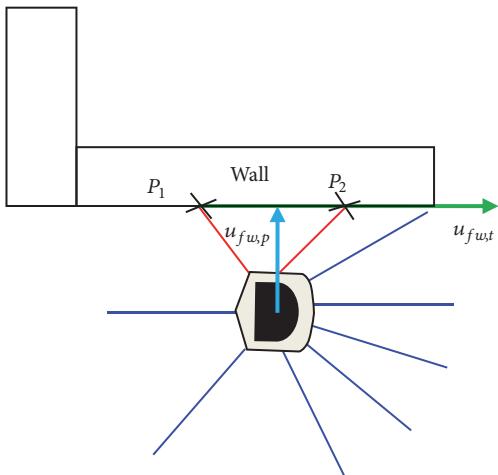
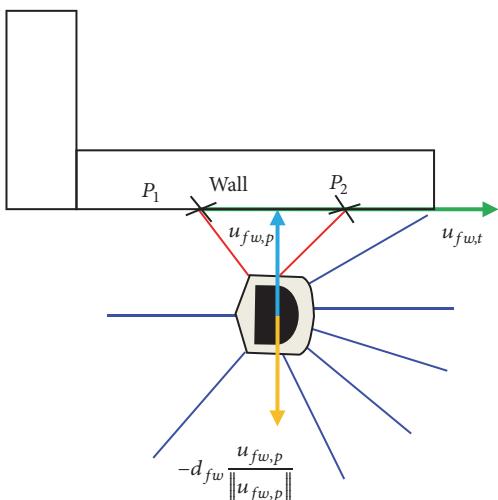
FIGURE 8: Computation of the vector $u_{fw,p}$.

FIGURE 9: Determination of the opposite vector.

(iii) Step 3: Find a vector that points in the opposite direction of the perpendicular vector and maintain some distance d_{fw} from the wall as shown in Figure 9.

$$u'_{fw,p} = u_{fw,p} - d_{fw} \frac{u_{fw,p}}{\|u_{fw,p}\|} \quad (21)$$

(iv) Step 4: Combine the two vectors $u_{fw,t}$ and $u_{fw,p}$ into a single vector u_{fw} that allows the mobile robot to follow the wall at some distance d_{fw} .

It is given by the following equation:

$$u_{fw} = d_{fw} \cdot u'_{fw,t} + (u_{fw,p} - d_{fw} \cdot u'_{fw,p}) \quad (22)$$

(v) Step 5: Use the vector u_{fw} and the pose of the mobile robot to compute the heading θ_{fw} :

$$\theta_{fw} = \tan^{-1} \frac{u_{fw}(y)}{u_{fw}(x)} \quad (23)$$

The input of the fuzzy controller of this behavior is the error (Φ_{fw}) between the angle of wall following (θ_{fw}) and the current heading of the mobile robot (θ). It is given by the following equation:

$$\Phi_{fw} = \theta_{fw} - \theta \quad (24)$$

The outputs of this controller are the linear and angular velocity.

Inputs and outputs variables of this controller have the same partition of membership functions and the same rules of the previous behavior “go to goal_avoidance obstacle.”

4. Design of the PID Controller

In the literature, numerous researches use the PID controller for the path planning. However, the parameters of the PID controller can affect the navigation steps. For that, its accuracy can be enhanced by tuning the parameters (k_p, k_i, k_d).

This shows that the parameters of the PID controller are not chosen in optimal way. In order to improve the performances of the navigation of the mobile robot, we have applied the Firefly and the Artificial Bee Colony algorithms to the PID controller.

The PID discrete law that assists the mobile robot to reach the desired goal without hitting with obstacles can be written as follow:

$$u_i(k) = k_{Df} e_i(k) + k_{Ii} T \sum_{j=0}^k e_i(j) + K_{DI} \frac{1}{T} \Delta e(k) \quad (25)$$

where k and T are, respectively, the current discrete instant time and the sampling period.

The objective function employed to get an optimized PID controller is the integral square error (ISE).

4.1. Artificial Bee Colony: Overview. Artificial bee colony is inspired by the intelligent foraging behavior of honey bee swarm. It is developed in 2005 [29] by Derviş Karaboga. In this clever algorithm, there are three types of bee colony that is based on the role it plays: the onlooker, employed (forager), and the scout bee.

The onlooker bees wait on the area of the waggle dance in purpose to choose a source of food. The forager bees keep visiting the food sources that are visited previously to get nectar. The scout bees perform random searches to discover novel sources of food.

The artificial bee colony can be divided into four essential steps.

(i) Initialization phase: The random initial population is given by the following equation:

$$x_i = x_{\min} + \text{rand}(0, 1) * (x_{\max} - x_{\min}) \quad (26)$$

(ii) Employed bee phase: Each employed bee was affected to a food source. After that, each employed bee discovers a novel neighboring food source of its presently attributed food source throughout (26) and calculates for each iteration the nectar amount of the novel food source. The employed bee is

displaced to the novel food source when the nectar amount of the new food is more raised compared to the previous one.

$$V_{i,j} = x_{i,j} + \text{rand}(-1, 1)(x_{i,j} - x_{k,j}) \quad (27)$$

(iii) Bee Source Selection: In this phase, the employed bees displace depending on the income rate of their sources. Food sources which have the higher income rates are more probable to be chosen. The probability function is defined as follows:

$$P_i = \frac{\text{fit}(x_i)}{\sum_{n=1}^N \text{fit}(x_n)} \quad (28)$$

where $\text{fit}(x_i)$ represents the fitness value of the nth solution.

Fitness is given by the following equation:

$$\text{fit}(x_n) = \begin{cases} \frac{1}{1 + f(x_n)}, & \text{if } f(x_n) \geq 0 \\ 1 + \text{abs}(f(x_n)), & \text{if } f(x_n) < 0 \end{cases} \quad (29)$$

where $f(x_n)$ is the objective function value of bee source x_n

(iv) Population elimination: In this stage, if we assume that some solution gains do not get improvement after updates, it is then supposed to be taken into local optimum and is disused; after that, the corresponding onlooker bees transformed into scouting bees and generate a novel solution to substitute the removed solution.

4.2. Firefly Algorithm: Overview. Firefly algorithm is a new metaheuristic algorithm inspired by the social lighting behavior of fireflies. It was first introduced by Yang [30].

In nature, there are many species of fireflies and most of them produce rhythmic and short flashes. The rate and the rhythmic flash and the amount of time form a part of the signal system which brings both sexes together. Therefore, the main part of the firefly's flash is to act as a signal system to attract other fireflies.

The FA adopts three particular rules based on few of the features of real fireflies:

(1) All fireflies are unisex. They will proceed towards the most attractive and brightest ones independently of their sex.

(2) The degree of their attraction is proportional to their luminosity which diminishes as the distance from other fireflies raises. The firefly will move arbitrarily if there is not any shinier or more attractive one.

(3) The brightness of a firefly is given by the value of the objective function of a specific problem.

Firefly algorithm has three important aspects, attractiveness being the first; each firefly has its special attractiveness which determines how many other fireflies of the swarm it attracts.

The attractiveness of a firefly is the monotonically decreasing function given by the following equation:

$$\beta(r) = \beta_0 e^{-\gamma(r_{ij})^m}; \quad m \geq 1 \quad (30)$$

where r represents the distance between two fireflies, β_0 represents the value of the initial attractiveness when $r = 0$,

and γ represents an absorption coefficient which controls the decrease of the light intensity.

The second one is the Distance. If we consider two fireflies i and j , at positions x_i and x_j , respectively, the distance between these two ones can be determined as follows:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (31)$$

where $x_{i,k}$ represents the k th component of the coordinate x_i of the i th firefly and d represents the number of dimensions.

The third issue is the Movement. The motion of a firefly i which is attracted to another more brighter firefly j is defined by the following equation:

$$x_i = x_i + \beta_0 e^{-\gamma(r_{ij})^2} (x_j - x_i) + \alpha \left(\text{rand} - \frac{1}{2} \right) \quad (32)$$

where the first term represents the actual position of a firefly; the second term is used for considering a firefly's attractiveness to light intensity seen by adjacent fireflies and the third term is employed for the arbitrary motion of a firefly when no shining ones exist. α is a randomization variable and rand represents a random number generator uniformly distributed in the space $(0,1)$.

5. Design of the Fuzzy Logic Controller

The optimized PID controller allows the two-wheeled mobile robot to find a safe path but not the short one.

To obtain better results, we developed, for the two behaviors, a Same Fuzzy Logic Controller (S.F.L.C.) that helps the robot to follow the shortest trajectory. It is more expressive and interpretable; the results of rules are more simple and intuitive.

The constructed fuzzy logic controller uses only one input depending on the scenario in the strange environment. In other words, if the "go to goal and obstacle avoidance" behavior is activated, the input of the fuzzy controller is the error between the angle of "go to goal and obstacle avoidance" and the current heading of the robot ($\Phi_{ao,gtg}$). If the "wall following" behavior is activated, the input of the fuzzy controller is the error between the angle of "wall following" and the current heading of the mobile robot (Φ_{fw}).

The range of inputs of this fuzzy logic controller is divided into eleven linguistic variables.

Figure 10 illustrates the distribution of membership functions for input variables of the two basic behaviors: "go to goal and obstacle avoidance" and "wall following."

They have been extracted after several trials based on their possible variation as to obtain the best system response.

The range of the first output of this fuzzy logic controller (angular velocity) is divided into eleven linguistic variables while the range of the second output (linear velocity) is divided into four linguistic variables.

Figures 11 and 12 show the distribution of membership functions for outputs variables of the two developed behaviors.

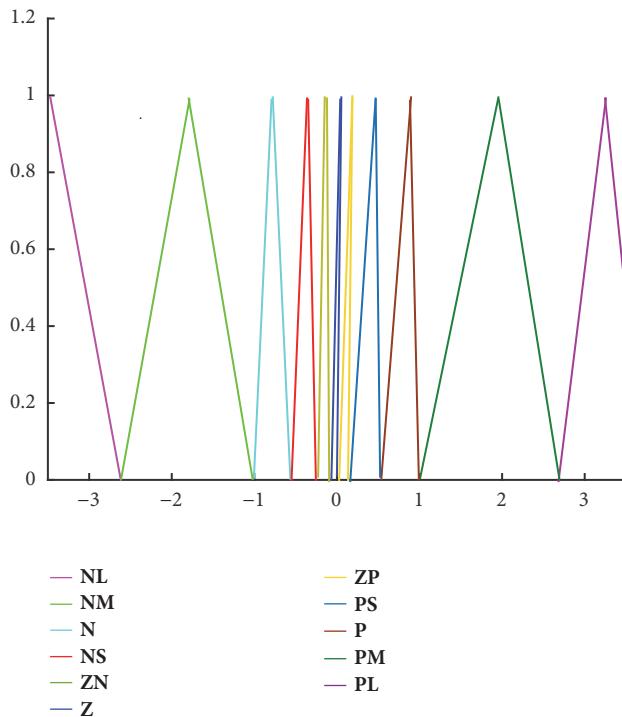


FIGURE 10: Membership functions of input variables for the two behaviors.

Since the fuzzy membership functions of inputs and outputs share similar names, common abbreviations have been used and given in Table 1.

Table 2 summarized the obtained fuzzy rule base for the linear and angular velocities. These rules are manually developed after several simulations.

6. Simulation Results

This part is dedicated to test the effectiveness of the proposed approach. For this aim, we consider a robot which starts from an initial position (x, y) = (1.3, 1.3) and tries to move directly to the final position (-1.2, -1.2).

Firstly, we considered an environment which contains scattered obstacles without deadlocks, using the S.F.L.C. approach. Figure 13 shows the trajectories covered by the two wheeled mobile robot.

It is obvious that the two-wheeled mobile robot can proceed towards the goal until reaching the desired target, which confirms the efficiency of the developed method.

In the next simulation, the mobile robot navigates in an environment containing deadlock. To choose the correct reactive behavior, the suggested path planner controller should study all information. As it is illustrated in Figure 14, at the beginning of navigation, the mobile robot turns and proceeds towards the goal. When the robot senses an object in front of it, and the goal is located on the other side of the object, the arbitration mechanism realizes that the robot does not progress towards the goal and switches to the wall following behavior. The robot changes its orientation angle when it is close to the obstacles. At the end, the track leading

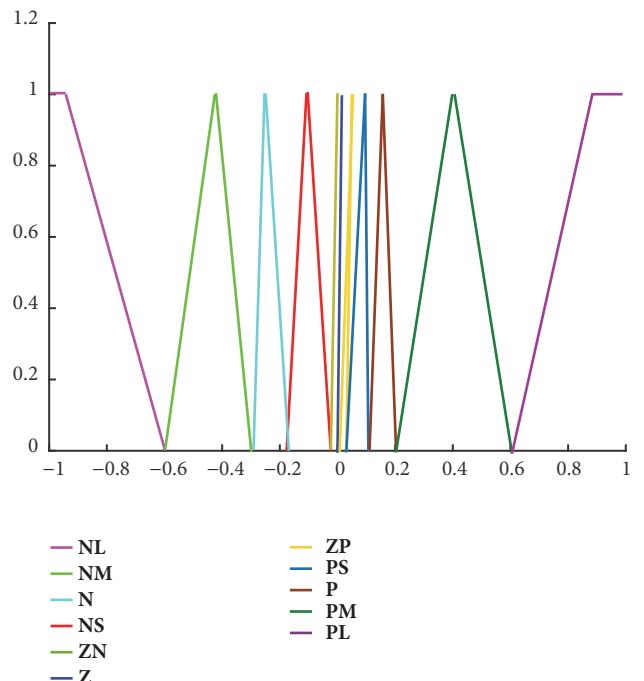


FIGURE 11: Membership functions of angular velocity for the two behaviors.

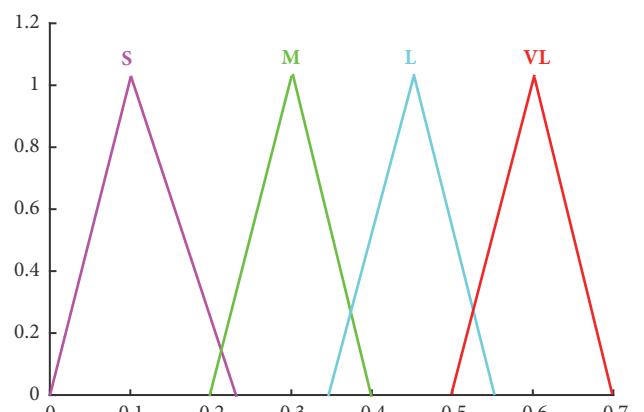


FIGURE 12: Membership functions of linear velocity for the two behaviors.

to the goal is clear; the robot is safe and reaches its desired target.

7. Comparison with Other Approaches

The main contribution in this work is that the designed control method based on S.F.L.C. deals with the complexity of the environment and guarantees the optimization of the process.

To introduce the validity of our proposed technique, we have compared it with three other methods based on PID controller tuned with the PSO [28] and ABC and FA algorithms (PSO-PID, ABC-PID and FA-PID). Performances

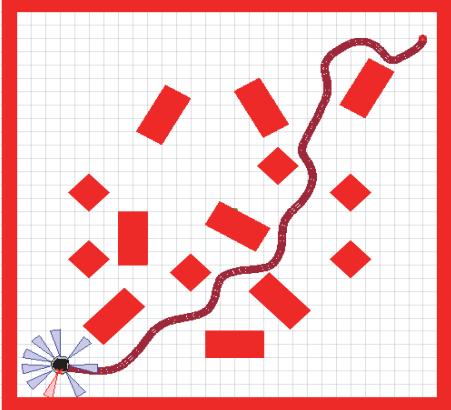


FIGURE 13: Trajectory of the mobile robot using S.F.L.C. without deadlock.

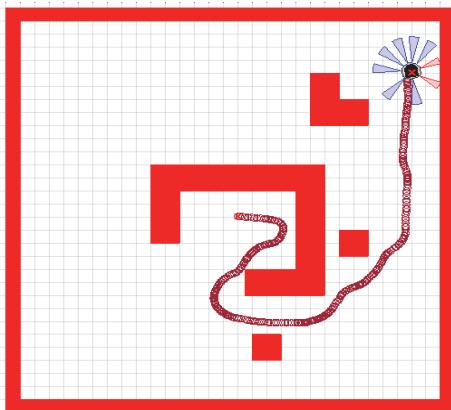


FIGURE 14: Trajectory of the robot using S.F.L.C. in an environment with deadlock.

TABLE 1: Abbreviations of linguistic values.

Linguistic value	Abbreviation
NL	Negative Large
NM	Negative Medium
NS	Negative Small
N	Negative
ZN	Zero Negative
Z	Zero
ZP	Zero Positive
PS	Positive Small
P	Positive
PM	Positive Medium
PL	Positive Large
S	Small
M	Medium
VL	Very large

are illustrated in Figures 15, 16, and 18. It is clear that the S.F.L.C. approach performs better.

Figures 17(a) and 17(b) show the evolution of the linear and angular velocities in the two unknown environments.

In order to evaluate the performance of the two-wheeled mobile robot using the PSO-PID [28], ABC-PID, FA-PID,

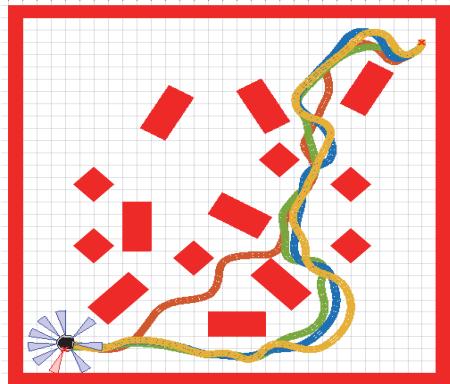


FIGURE 15: Path planning of the mobile robot in an environment without deadlock using S.F.L.C. (red), PSO-PID (yellow), ABC-PID (blue), and FA-PID (green).

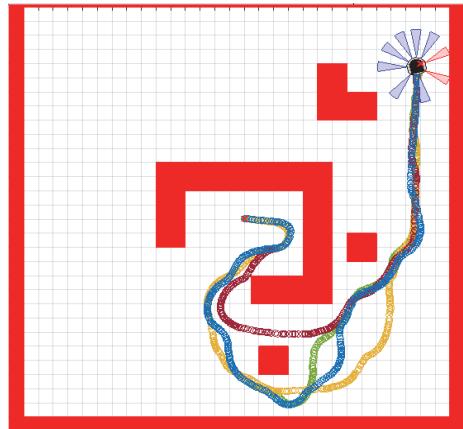


FIGURE 16: Path planning of the mobile robot in an environment with deadlock using S.F.L.C. (red), PSO-PID (yellow), ABC-PID (blue), and FA-PID (green).

TABLE 2: Fuzzy rule base of the two reactive controllers.

Φ_{ao-gtg} or Φ_{fw}	Linear Velocity	Angular Velocity
NL	S	NL
NM	S	NM
N	S	N
NS	M	NS
ZN	L	ZN
Z	L	Z
ZP	L	ZP
PS	M	PS
P	S	P
PM	S	PM
PL	S	PL

and S.F.L.C., two criteria have been considered which are the navigation time and the travelled distance. Numerical results are illustrated in Tables 3 and 4. The comparison between these results proves the efficiency of S.F.L.C. controller.

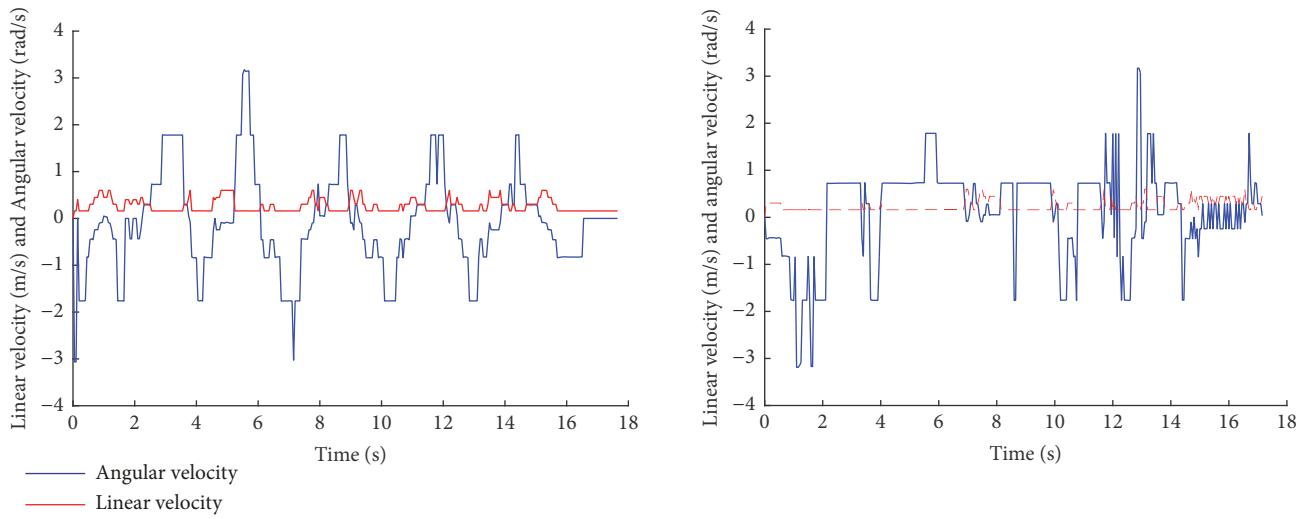
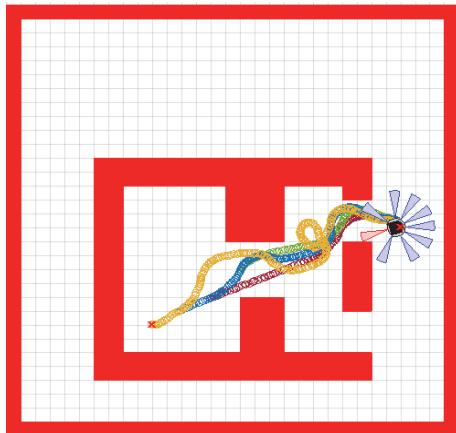
FIGURE 17: Evolution of the two velocities (w and v) in the environment (a) without deadlock and (b) with deadlock.

FIGURE 18: Path planning of the mobile robot in an environment with deadlock using S.F.L.C. (red), PSO-PID (yellow), ABC-PID (blue), and FA-PID (green).

TABLE 3: Comparison between navigation time using PSO-PID, ABC-PID, FA-PID, and S.F.L.C.

	Figure 15	Figure 16	Figure 18
PSO-PID	37.1	23.9	12.2
ABC-PID	36.35	22.45	9
FA-PID	33.95	21	7.5
S.F.L.C.	18	17	6.5

TABLE 4: Comparison between travelled distance using PSO-PID, ABC-PID, FA-PID, and S.F.L.C.

	Figure 15	Figure 16	Figure 18
PSO-PID	5.565	3.585	1.83
ABC-PID	5.4525	3.3675	1.35
FA-PID	5.0925	3.15	1.125
S.F.L.C.	3.843	2.784	0.936

8. Conclusion

In this work, we have applied a new method, S.F.L.C., for controlling the navigation of the mobile robot in a strange environment. We have mainly used the two primitive reactive behaviors: “go to goal_obstacle avoidance” and “wall following” as well as an arbitration mechanism responsible for switching to the suitable behavior according to the circumstances in the unknown environment. The mobile robot is now able to avoid obstacles and to escape from deadlocks, reaching the target successfully. The proposed path planner controller has been compared with other related works, and it has been deduced that the current motion controller gives better results. For future work, we will focus on testing our proposed method on the real Khepera III mobile robot.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] E. J. Gómez, F. M. Santa, and F. H. M. Sarmiento, “A comparative study of geometric path planning methods for a mobile robot: potential field and voronoi diagrams,” in *Proceedings of the 2nd International Congress of Engineering Mechatronics and Automation, CIIMA '13*, 2013.
- [2] M. Weigl, B. Siemiaatkowska, K. A. Sikorski, and A. Borkowski, “Grid-based mapping for autonomous mobile robot,” *Robotics and Autonomous Systems*, vol. 11, no. 1, pp. 13–21, 1993.
- [3] E. Masehian and M. R. Amin-Naseri, “A voronoi diagram-visibility graph-potencial field compound algorith for robot path planning,” *Journal of Robotic Systems*, vol. 21, no. 6, pp. 275–300, 2004.
- [4] K.-H. Park, Y.-J. Kim, and J.-H. Kim, “Modular Q-learning based multi-agent cooperation for robot soccer,” *Robotics and Autonomous Systems*, vol. 35, no. 2, pp. 109–122, 2001.
- [5] S. Garrido, L. Moreno, D. Blanco, and P. Jurewicz, “Path planning for mobile robot navigation using voronoi diagram and fast marching,” *International Journal of Robotics and Automation*, vol. 2, no. 1, pp. 42–64, 2011.
- [6] J. Borenstein and Y. Koren, “The vector field histogram—fast obstacle avoidance for mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.
- [7] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [8] I. Engedy and G. Horváth, “Artificial neural network based local motion planning of a wheeled mobile robot,” in *Proceedings of the 11th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2010*, pp. 213–218, Hungary, November 2010.
- [9] I. Baturone, A. Gersnoviez, and Á. Barriga, “Neuro-fuzzy techniques to optimize an FPGA embedded controller for robot navigation,” *Applied Soft Computing*, vol. 21, pp. 95–106, 2014.
- [10] B. B. V. L. Deepak, D. R. Parhi, and B. M. V. A. Raju, “Advance Particle Swarm Optimization-Based Navigational Controller For Mobile Robot,” *Arabian Journal for Science and Engineering*, vol. 39, no. 8, pp. 6477–6487, 2014.
- [11] A. Ghorbani, S. Shiry, and A. Nodehi, “Using genetic algorithm for a mobile robot path planning,” in *Proceedings of the 2009 International Conference on Future Computer and Communication, ICFCC 2009*, pp. 164–166, Malaysia, April 2009.
- [12] M. A. P. Garcia, O. Montiel, O. Castillo, R. Sepúlveda, and P. Melin, “Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation,” *Applied Soft Computing*, vol. 9, no. 3, pp. 1102–1110, 2009.
- [13] P. K. Mohanty and D. R. Parhi, “Optimal path planning for a mobile robot using cuckoo search algorithm,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 1–2, pp. 35–52, 2016.
- [14] H. Miao and Y. C. Tian, “Dynamic robot path planning using an enhanced simulated annealing approach,” *Applied Mathematics and Computation*, vol. 222, pp. 420–437, 2013.
- [15] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning I,” *Information Sciences*, vol. 8, pp. 199–249, 1975.
- [16] L. Ren, W. Wang, and Z. Du, “A new fuzzy intelligent obstacle avoidance control strategy for wheeled mobile robot,” in *Proceedings of the 2012 9th IEEE International Conference on Mechatronics and Automation, ICMA 2012*, pp. 1732–1737, China, August 2012.
- [17] S. K. Pradhan, D. R. Parhi, and A. K. Panda, “Fuzzy logic techniques for navigation of several mobile robots,” *Applied Soft Computing*, vol. 9, no. 1, pp. 290–304, 2009.
- [18] N. Yousfi, C. Rekik, M. Jallouli, and N. Derbel, “Optimized fuzzy controller for mobile robot navigation a cluttered environment,” in *Proceedings of the 2010 7th International Multi-Conference on Systems, Signals and Devices, SSD-10*, Jordan, June 2010.
- [19] Q. Bao, S. Li, W. Shang, and M. An, “A Fuzzy Behavior-Based Architecture for Mobile Robot Navigation in Unknown Environments,” in *Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence*, pp. 257–261, Shanghai, China, November 2009.
- [20] S. A.-L. El-Teleti, Z. B. Nossair, H. M. Abdel-Kader Mansour, and A. TagElDein, “Fuzzy logic control of an autonomous mobile robot,” in *Proceedings of the 2011 16th International Conference on Methods and Models in Automation and Robotics, MMAR 2011*, pp. 188–193, Poland, August 2011.
- [21] S. M. Raguraman, D. Tamilselvi, and N. Shivakumar, “Mobile robot navigation using fuzzy logic controller,” in *Proceedings of the 2009 International Conference on Control Automation, Communication and Energy Conservation, INCACEC 2009*, India, June 2009.
- [22] S. Wu, Q. Li, E. Zhu, J. Xie, and G. Zhichao, “Fuzzy controller pipeline robot navigation optimized by genetic algorithm,” in *Proceedings of the Chinese Control and Decision Conference 2008, CCDC 2008*, pp. 904–908, China, July 2008.
- [23] U. Farooq, K. M. Hasan, G. Abbas, and M. U. Asad, “Comparative analysis of zero order Sugeno and Mamdani fuzzy logic controllers for obstacle avoidance behavior in mobile robot navigation,” in *Proceedings of the 2011 International Conference and Workshop on the Current Trends in Information Technology, CTIT'11*, pp. 113–119, UAE, October 2011.
- [24] M. Wang and J. N. K. Liu, “Fuzzy logic-based real-time robot navigation in unknown environment with dead ends,” *Robotics and Autonomous Systems*, vol. 56, no. 7, pp. 625–643, 2008.

- [25] M. Boujelben, C. Rekik, and N. Derbel, "A hybrid fuzzy-sliding mode controller for a mobile robot," *International Journal of Modelling, Identification and Control*, vol. 25, no. 3, pp. 155–164, 2016.
- [26] S. H. A. Mohammad, M. A. Jeffril, and N. Sariff, "Mobile robot obstacle avoidance by using Fuzzy Logic technique," in *Proceedings of the 2013 IEEE 3rd International Conference on System Engineering and Technology, ICSET 2013*, pp. 331–335, Malaysia, August 2013.
- [27] M. Boujelben, C. Rekik, and N. Derbel, "A multi-agent architecture with hierarchical fuzzy controller for a mobile robot," *International Journal of Robotics and Automation*, vol. 30, no. 3, pp. 289–298, 2015.
- [28] L. Lai, Y. Chang, J. Jeng, G. Huang, W. Li, and Y. Zhang, "A PSO method for optimal design of PID controller in motion planning of a mobile robot," in *Proceedings of the 2013 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*, pp. 134–139, Taipei, Taiwan, December 2013.
- [29] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep., Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [30] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver, Frome, UK, 2008.