IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 19, 2017, accepted November 23, 2017, date of publication December 6, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2780082

Human Expertise in Mobile Robot Navigation

MOHAMMED FAISAL[1,2], (Member, IEEE), MOHAMMED ALGABRI[2], BENCHERIF MOHAMED ABDELKADER[2], AND HABIB DHAHRI[1], AND MOHAMAD MAHMOUD AL RAHHAL[1], (Member, IEEE)

[1]College of Computer and Information Sciences, King Saud University, Al Muzahimiyah 19676, Saudi Arabia
[2]Center of Smart Robotics Research, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Corresponding author: Mohammed Faisal (mfasial@ksu.edu.sa)

This work was supported by the Deanship of Scientific Research at King Saud University through the Research Group under Grant RG-1438-071.

ABSTRACT Numerous applications, such as material handling, manufacturing, security, and automated transportation systems, use mobile robots. Autonomous navigation remains one of the primary challenges of the mobile robot industry; many new control algorithms have been recently developed that aim to overcome this challenge. These algorithms are primarily related by their adoption of new strategies for avoiding obstacles and minimizing the travel time to a target along an optimal path. In this paper, we introduce four different navigation systems for an autonomous mobile robot (PowerBot) and compare them. The four systems are based on a fuzzy logic controller (FLC). The FLC of one system is tuned by an inexperienced human (naive), while the three other FLCs are optimized through a genetic algorithm (GA), particle swarm optimization (PSO), and a human expert. We hope the comparison answers the question of which is the best controller. In other words, ''who can win?,'' the naive, the GA, the PSO, or the expert, in fine tuning the membership functions of the navigation and obstacle avoidance behavior of the mobile robot? To answer this question, we used four different techniques for optimization (the naive FLC, GA, PSO, and FLC-expert) and used many criteria for comparison, whereas other research papers have dealt with two techniques at a time.

INDEX TERMS Mobile robot, genetic algorithm, partial swarm optimization, fuzzy logic control, robot navigation, avoid obstacles.

I. INTRODUCTION

Researchers expect that mobile robots will be responsible for several tasks in human life. Examples include warehouse management, packet distribution and arrangement, product handling inside stockrooms, and in working in accessible but dangerous sites [1]–[3]. Navigation is definitively one of the strategic tasks for mobile robots. Many advanced approaches have been used for autonomous navigation; however, this subject has not been thoroughly elucidated to date [4]. Diverse formulations have been developed for the autonomous mobile robots navigation, during the last decades, moreover, these tremendous developments could not cope with the new robotic challenges that are becoming more challenging. These new types of situations are essentially owed to the dynamic and incomplete knowledge about the new complex and unknown environments. The diverse self-control techniques, such as fuzzy logic, neural networks, genetic algorithms, have widely been used to tackle this type of dynamic and compensate for some unknown knowledge [5].

Cao and al. used multiple types of sensors (sonars and cameras) and stored maps with a fuzzy logic navigation approach for the mobile robot [6]. Unfortunately, authors did not show all the simulation conditions and the experimental design in their paper, making it difficult to repeat their experiments, in a controlled environment. Another FLC approach for indoor navigation developed by [7], where authors used a fuzzy logic controller for tracking a target and controlling their Wheeled Mobile Robot (WMR). The authors concentrated their focus on the robot navigation, without interest to avoid obstacles; they used the FLC uniquely to control the motion of the WMR. Faisal [8] have developed an online navigation system for their WMR (Scout II robot), including two FLC, within an unknown environment. A tracking Fuzzy Logic Controller (TFLC) is used for navigate control, and an Obstacle Avoiding Fuzzy Logic Controller (OAFLC) is used for obstacles avoiding. An indoor FLC system for an autonomous vehicle was presented in [9], where authors used a camera sensor and guided the robot to its goal through FLC. Nevertheless, the authors concentrated on navigation without

considering obstacles avoidance; FLC was used only for navigation. A real-time FLC tracking system using dual robots and infrared sensors was proposed in [10]. The first WMR served as the followed target and the second as the following tracker. FLC, genetic algorithms and neural networks have been exploited to ameliorate the control strategy. In order to determine the set of optimal parameters of the FLC Narvydas *et al.* [11] used a genetic algorithm. A multi FLC system compose of 4 FLC was used in [12], these controllers were designed in a hierarchical control system, three navigation controllers were used for navigation and obstacle avoidance, while the four controllers acted as a supervisor node. Another approach using GA to tune fuzzy logic in order to modify the shape of the membership functions has been developed by [13], The experimental results have shown that this Geno-Fuzzy system enhanced the navigation in some and not in all the experiments. A different FL navigation system using a gradient and a genetic method was developed in [14]. A Multi-Objective Genetic Algorithm (MOGA) for path planning was described in [15], where authors have shown how to use two fitness assignments for the autonomous mobile robot. Adriansyah and S. H. Amin designed a technique for membership function tuning and generated fuzzy rules using Particle Swarm Optimization (PSO) [16]. The simulation, using MATLAB/SIMULINK, has shown that this technique had the fastest response time compared to several other techniques when new situations occur. Wong *et al.* [17] have determined automatically through a PSO the appropriate parameters of the membership functions, of the two-wheel robot fuzzy controller. The method developed under MATLAB was applied on a real case, where a real soccer robot has shown good performance. Martínez-Marroquín *et al.* [18] described an optimizing method for the parameters of the member functions of the FLC through a PSO tuning, in a MATLAB/SIMULINK simulation. Gupta *et al.* [19] have developed a Takagi-Sugero fuzzy motion controller, in which the parameters of the controller were optimized also optimized by a PSO using the statistical approximation method. Obstacles were recognized by a vision sensor.

For this study, we built different navigation modules using naive, fuzzy-genetic, fuzzy-PSO, and FLC-Expert systems and later compared these modules to determine which performed best based on the path followed, time to target and, in particular, the obstacle avoidance strategy.

The Naive, fuzzy-genetic (FG), fuzzy-PSO (FPSO), and FLC-Expert (FLCE) systems are used to find obstacle free paths and guide the mobile robot as it navigates to its target. In both FG and FPSO systems, the fuzzy system is used to create the elementary membership functions; next, the GA and PSO algorithms enhance these memberships to enhance the performance of the PowerBot navigation during operation.

There has been research focused on optimizing the membership of the FLC using different techniques (i.e., PSO, NN, GA). However, our research is different in several aspects.

1) We compared four optimizing techniques (Naive, GA, PSO, and FLC-Expert), while both (FG, PSO) [20] compared only two techniques.

2) We applied our modules to a mobile robot (PowerBot), while [20] only used simulations.

3) We used many criteria for comparison (distance traveled, time to target, obstacle avoidance strategy, and Membership Tuning), while [20] used only distance and time to target, while other used only path length and the energy of the path.

We organized the paper as follows. A review of the literature is presented in section 2. Section 2 explains the proposed system. Comparable systems are described in section 3. Section 4 provides an analysis of the proposed modules. Finally, the study's conclusions are presented in section 5.

II. PROPOSED SYSTEMS

The proposed system contains two fuzzy logic modules: a Navigation Behavior Module and an Avoiding Obstacles Module. These two modules are initialized by the fuzzy logic "Naive" module and then enhanced by the GA, PSO, and human expert. The navigation module is used for navigation and the avoiding obstacles module is used to simulate the behavior of avoiding obstacles.

A. NAIVE FUZZY LOGIC MODULES

We first introduce the Naive fuzzy logic system for navigation, we next describe the method for avoiding obstacles, and we finally optimize the model using GA, PSO, and the expert.

1) NAIVE: NAVIGATION BEHAVIOR MODULE (NBM)

Naive-NBM is used to navigate the mobile robot to its target point. NBM has two inputs and two outputs, as showed in Figure 1.

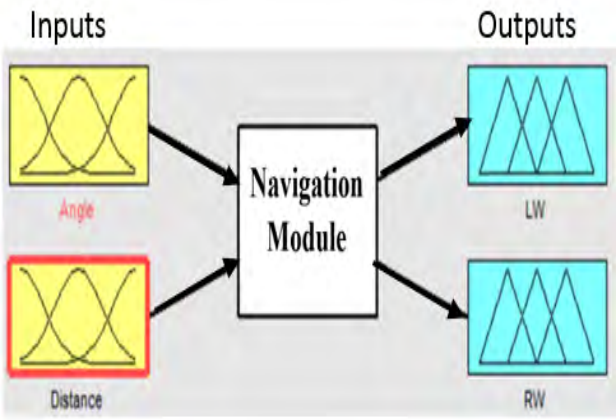

FIGURE 1. Navigation module.

The heading angle between the robot and the target "α" and the distance between the robot and the target "d", will be the inputs of the Naive navigation module, as illustrated in Figure 2.

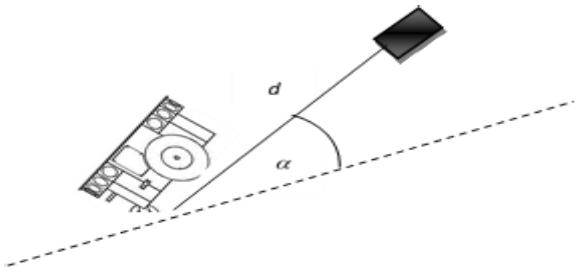

FIGURE 2. Inputs of the Navigation module.

The left and right velocities of the motors will be the outputs of the Naive navigation module. Seven membership functions for inputs (α, d) are used in the Naive navigation module. Figures 3 and 4 show the membership functions of α and d.

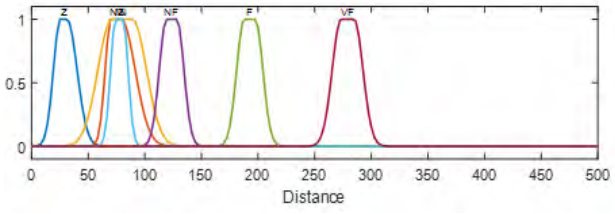

FIGURE 3. Naive-NBM module (Membership functions of d; Z symbolizes for Zero, NZ symbolizes for Near Zero, N symbolizes for Near, M symbolizes for Medium, NF symbolizes for Near Far, F symbolizes for Far, VF symbolizes for Very Far).

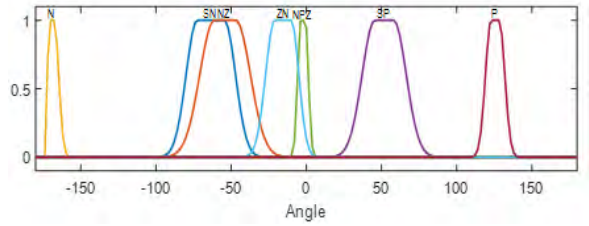

FIGURE 4. Naive-NBM module (Membership functions of α; N symbolizes for Negative, SN symbolizes for Small Negative, NNZ symbolizes for Near Negative Zero, Z symbolizes for Zero, NPZ symbolizes for Near Positive Zero, SP symbolizes for Small Positive, and P symbolizes for Positive).

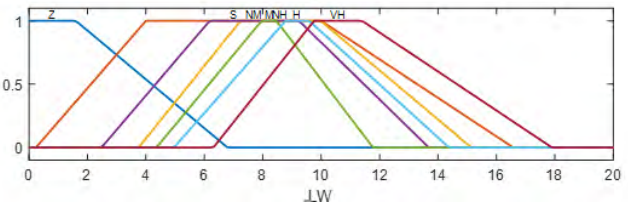

FIGURE 5. LV's membership functions of Naive-NBM module.

The left and right velocities (LV and RV) of the motors are the outputs of the Naive-NBM module. Figures 5 and 6 illustrate the membership functions of the LV and RV respectively, and the fuzzy rules are defined in Table 1.

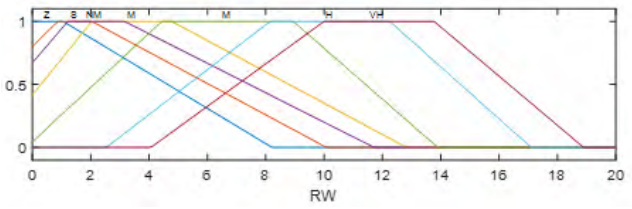

FIGURE 6. RV's membership functions of Naive-NBM module.

TABLE 1. Fuzzy rules of the LV and RV of Naive-NBM.

Angle / Dis.	N	SN	NNZ	Z	NPZ	SP	P
Z	L^Z	L^Z	L^Z	L^Z	L^{NM}	L^{NM}	L^M
	R^M	R^{NM}	R^{NM}	R^Z	R^Z	R^Z	R^Z
NZ	L^S	L^S	L^Z	L^S	L^M	L^{NH}	L^H
	R^H	R^{NH}	R^M	R^S	R^Z	R^S	R^S
N	L^S	L^S	L^S	L^{NM}	L^H	L^H	L^{VH}
	R^{VH}	R^H	R^{NH}	R^{NM}	R^S	R^S	R^S
M	L^S	L^M	L^S	L^M	L^H	L^H	L^{VH}
	R^{VH}	R^H	R^H	R^M	R^S	R^S	R^S
NF	L^S	L^S	L^S	L^{NH}	L^H	L^H	L^{VH}
	R^{VH}	R^H	R^H	R^{NH}	R^S	R^S	R^S
F	L^S	L^S	L^S	L^H	L^H	L^H	L^{VH}
	R^{VH}	R^H	R^H	R^H	R^S	R^S	R^S
VF	L^S	L^S	L^{NM}	L^{VH}	L^{NH}	L^H	L^{VH}
	R^{VH}	R^H	R^{NH}	R^{VH}	R^{NM}	R^S	R^S

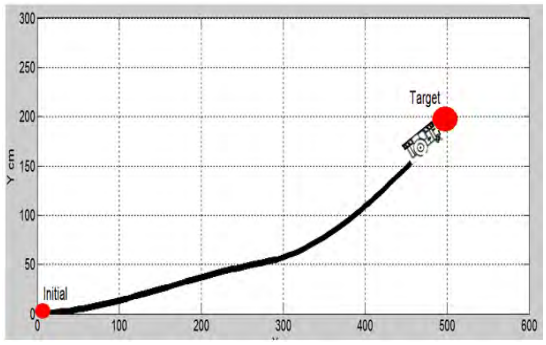

FIGURE 7. Path of the Naive-NBM module.

We applied the Naive-NBM module to the PowerBot mobile robot from the initial point (0, 0) to the target point (500,200) cm; the result is shown in Figure 7.

The Naive-NBM is not optimal in terms of time to target and path length, as we will see in the comparison section; therefore, there is a need to improve the Naive-NBM module.

2) NAIVE: AVOIDING OBSTACLES BEHAVIOR MODULE (AOBM)

The Naive-AOBM Module is used in conjunction with the Naive-NBM to find obstacle-free paths and navigate the PowerBot mobile robot to its target. The Naive-AOBM generates a left velocity and a right velocity (LV and RV) to avoid obstacles in an unknown dynamic environment.

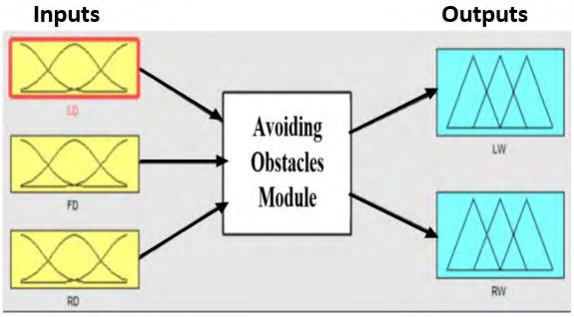

FIGURE 8. Naive-AOBM.

This Naïve-AOBM has three inputs and two outputs, as illustrated in Figure 8.

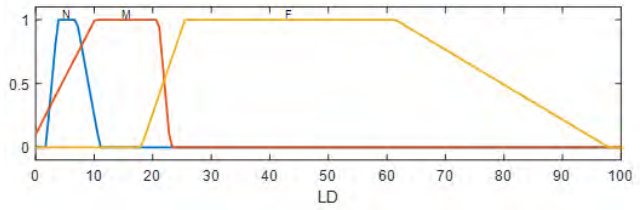

FIGURE 9. Left side obstacle distance of the Naive-AOBM (N: Near, M: Medium and F: Far).

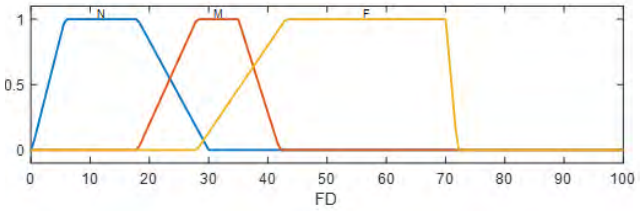

FIGURE 10. Front obstacle distance of the Naive-AOBM (N: Near, M: Medium and F: Far).

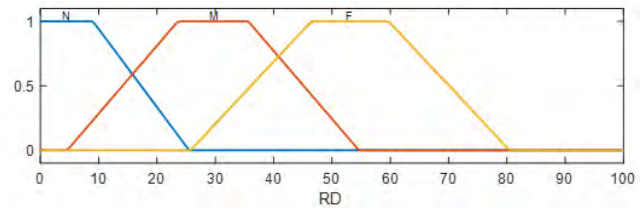

FIGURE 11. Right side obstacle distance of the Naive-AOBM (N: Near, M: Medium and F: Far).

The inputs of the Naive-AOBM are the distances between the front of the robot and the obstacle, the left, and right sides of the robot and the obstacle. Naive-AOBM used the laser rangefinder and the Ultrasonic sensors to measure those distances. The left velocity LV and right velocity RV of the motors are the outputs of the Naive-AOBM are the velocities of the left and right motors. As shown in Figures 9 to 11, Naive-AOBM used three membership functions to implement the inputs.

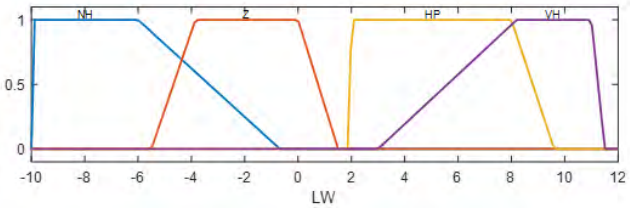

FIGURE 12. Membership functions for the LV of the Naive-AOBM (NH: Negative High, Z: Zero, HP: High Positive, and VH: Very High).

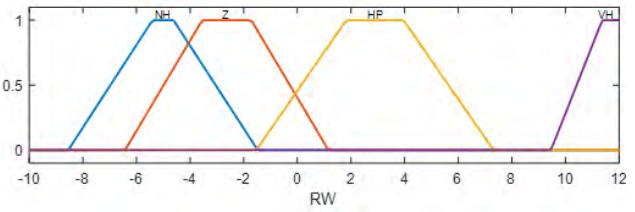

FIGURE 13. Membership functions for the RV of the Naive-AOBM (NH: Negative High, Z: Zero, HP: High Positive, and VH: Very High).

Each output of the Naive-AOBM used four membership functions to generate the LV and RV, as shown in Figures 12 and 13. The fuzzy rules of both velocities are shown in Table 2.

We applied the Naive-AOBM to the PowerBot mobile robot from the initial point (0, 0) to the target point (500 cm, 100 cm); the results are shown in Figures 14 and 15.

We can see that the Naive-AOBM was unable to avoid the obstacles. In this case, we need to improve the Naive avoiding obstacles module.

B. FUZZY-GENETIC SYSTEM (FG)

GA is used to optimize and tune the membership functions of the Naive-NBM and Naive-AOBM, and to improve the performance of PowerBot navigation during operation. The fuzzy-genetic module is illustrated in Figure 16.

1) FUZZY-GENETIC OF NAIVE NBM (FG-NBM)

We introduced the GA to enhance and tune the membership functions of the NBM. The GA used 100 generations; each generation has a population size of 100. We found that the best fitness was 16.896, which occurred in the 90th generation. The FG module was used to simulate the navigation of the PowerBot robot. The FG-NBM and Naive NBM have the same configuration for fuzzy rules, inputs, and outputs. Figures 17 to 20 illustrate the enhanced membership functions.

We applied the FG module of the NBM to the PowerBot mobile robot from the start point (0, 0) to the goal/target point (500 cm, 200 cm); the result is shown in Figure 21.

2) FUZZY-GENETIC OF NAIVE AOBM (FG-ABOM)

To enhance the AOBM, a GA is used to tune its membership functions. The FG-AOBM and Naive AOBM have similar

TABLE 2. Fuzzy rules of the LV and RV of Naive-AOBM.

Input			Output	
LD	FD	RD	RV	LV
Near	Near	Near	NH	Z
Near	Near	M	VH	Z
Near	Near	Far	VH	PH
Near	Medium	Near	PH	Z
Near	Medium	M	VH	PH
Near	Medium	Far	VH	PH
Near	Far	Near	PH	PH
Near	Far	M	VH	Z
Near	Far	Far	PH	VH
Medium	Near	Near	Z	NH
Medium	Near	M	NH	Z
Medium	Near	Far	Z	VH
Medium	Medium	Near	VH	Z
Medium	Medium	M	Z	HP
Medium	Medium	Far	HP	VH
Medium	Far	Near	VH	HP
Medium	Far	M	HP	VH
Medium	Far	Far	HP	VH
Far	Near	Near	VH	Z
Far	Near	M	VH	Z
Far	Near	Far	NH	Z
Far	Medium	Near	VH	HP
Far	Medium	M	VH	HP
Far	Medium	Far	Z	VH
Far	Far	Near	HP	Z
Far	Far	M	VH	HP
Far	Far	Far	z	VH

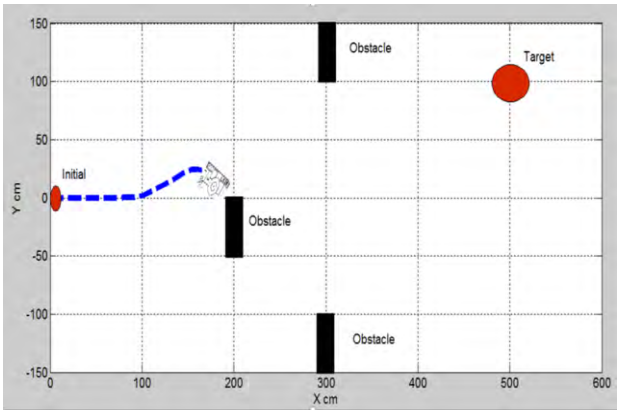

FIGURE 14. Scenario of the Naïve-AOBM module.

inputs, outputs, and fuzzy rules. Figures 22 to 26 illustrate the enhanced membership functions of both the inputs and outputs. The GA used 100 generations; each generation has

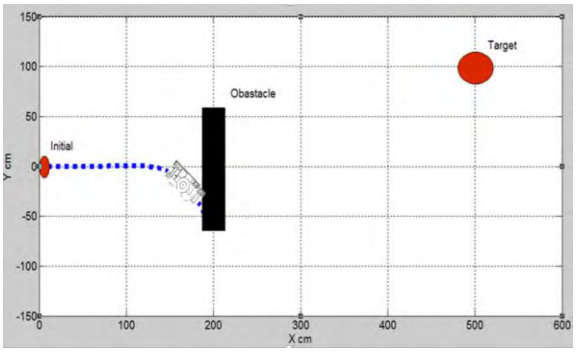

FIGURE 15. Alternative scenario of the Naïve-AOBM module.

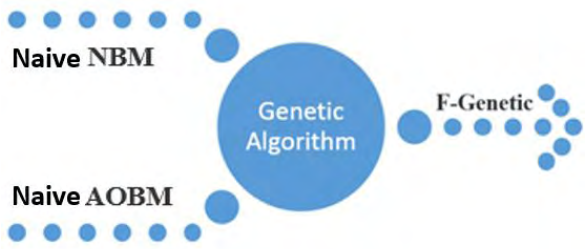

FIGURE 16. Fuzzy-Genetic Module.

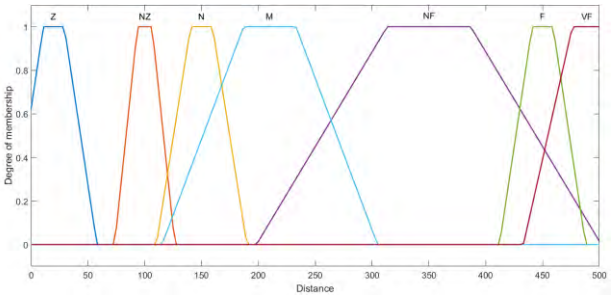

FIGURE 17. Enhanced membership of the distance of FG-NBM.

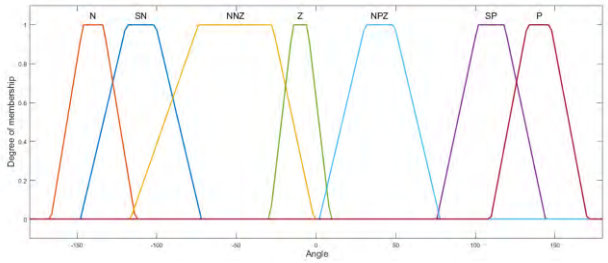

FIGURE 18. Enhanced membership of the angle of FG-NBM.

a population size of 100. The best fitness was 24.7, which occurred at the 100th generation.

The FG-AOBM was applied to the PowerBot mobile robot from the initial point (0, 0) to the target point (500 cm, 100 cm), as illustrated in Figure 27.

Note that, from Figures 17-20 and 22-26, the shapes of the membership functions, i.e., of the FG module of both NBM

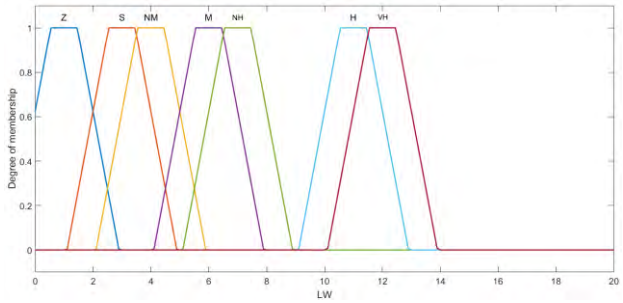

FIGURE 19. Enhanced membership of the LV of FG-NBM.

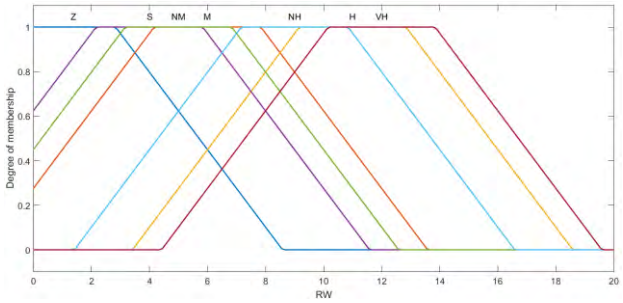

FIGURE 20. Enhanced membership of the RV of FG-NBM.

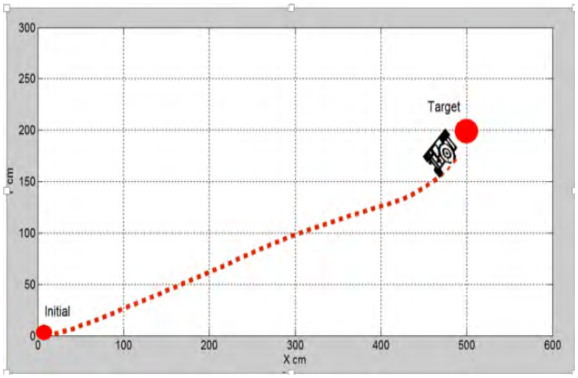

FIGURE 21. Path of the FG-NBM.

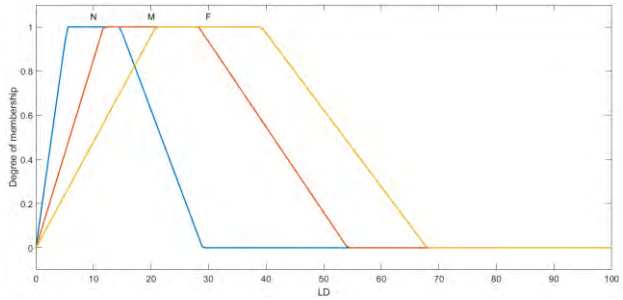

FIGURE 22. Enhanced membership of the left obstacle distance of the FG-ABOM.

and AOBM, are enhanced but still have deficiencies, such as overlapping shapes and non-optimal paths.

C. Fuzzy-PSO MODULE

In this module, the PSO is used to optimize and tune the membership functions of the Naive fuzzy logic controllers of

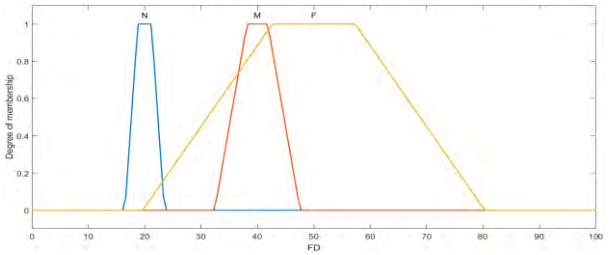

FIGURE 23. Enhanced membership of the front obstacle distance of the FG-ABOM.

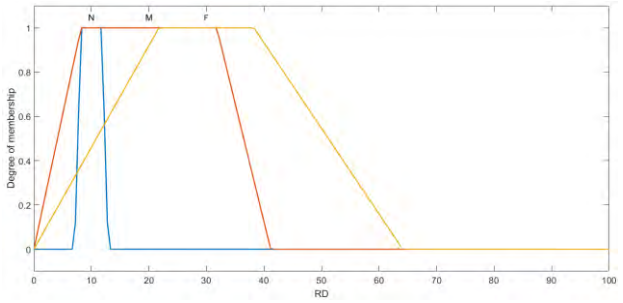

FIGURE 24. Enhanced membership of the right obstacle distance of the FG-ABOM.

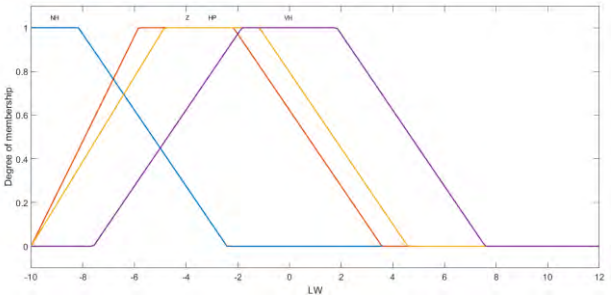

FIGURE 25. Enhanced membership of the LV of the FG-AOBM.

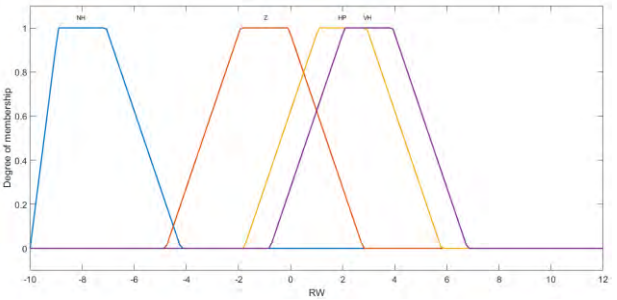

FIGURE 26. Enhanced membership of the RV of the FG-AOBM.

the NBM and AOBM to improve the performance of Power-Bot navigation during operation. The fuzzy-PSO module is illustrated in Figure 28.

1) FUZZY-PSO OF NAIVE NBM (FPSO-NBM)

The PSO algorithm is used with 100 generations, where each generation has a population size of 20. The best fitness was 17.13, which occurred at the 100th generation.

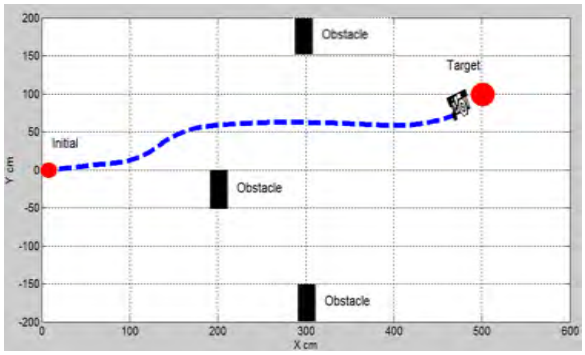

FIGURE 27. Scenario of the FG-AOBM.

FIGURE 28. Fuzzy-PSO Module.

The FPSO-NBM module is used to simulate the navigation behaviors of the mobile robot.

The FPSO-NBM and Naive NBM have similar inputs and outputs, as well as the same fuzzy rules. Figure 29 to Figure 32 illustrate the enhanced membership functions.

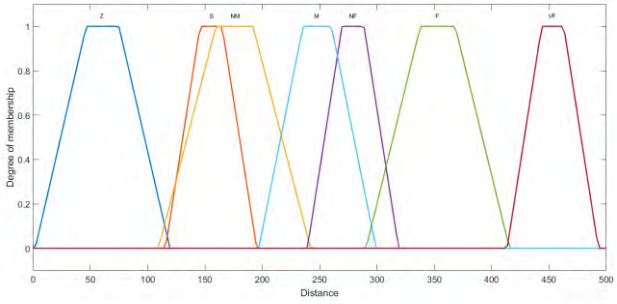

FIGURE 29. Enhanced membership of the distance error of FPSO-NBM.

The FP-NBM was applied to the PowerBot from the start point (0, 0) to the goal/target point (500 cm, 200 cm); the navigation result is shown in Figure 33.

2) FUZZY-PSO OF THE NAIVE AOBM (FPSO-ABOM)

In the FPSO-ABOM, the PSO was used to tune the membership functions of the AOBM with 100 generations; each generation had a population of 20. The best fitness that was obtained was 17.13. The FPSO-ABOM and Naive AOBM have similar inputs, outputs, and fuzzy rules. Figures 34 to 38 illustrate the enhanced membership functions.

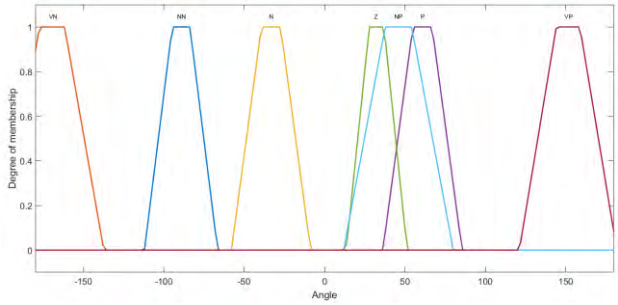

FIGURE 30. Enhanced membership of the angle error of FPSO-NBM.

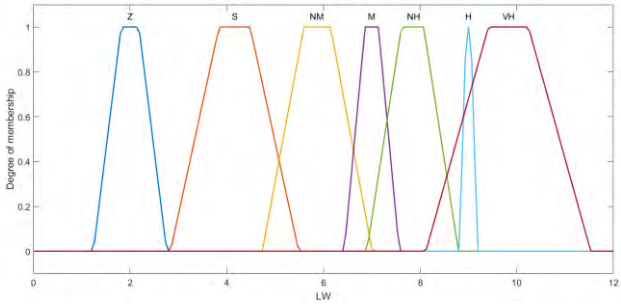

FIGURE 31. Enhanced membership of the LV of FPSO-NBM.

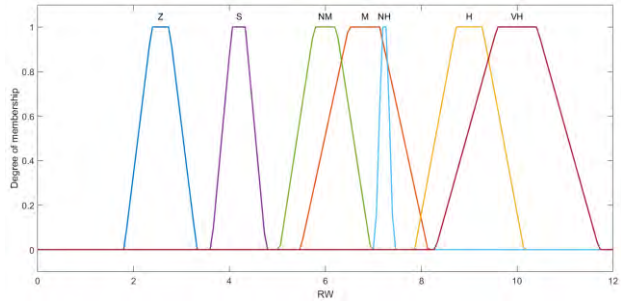

FIGURE 32. Enhanced membership of the RV of FPSO-NBM.

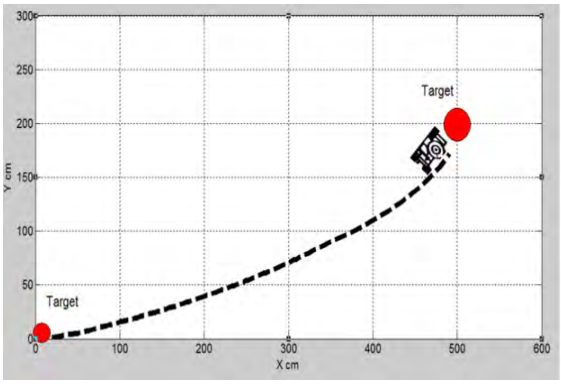

FIGURE 33. Path of the fuzzy-PSO NBM.

We applied the fuzzy-PSO module of the AOBM to the PowerBot mobile robot from the start point (0, 0) to the goal/target point (500 cm, 100 cm). Figure 39 illustrates the scenario.

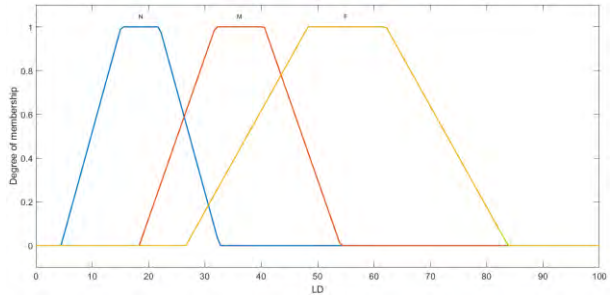

FIGURE 34. Enhanced membership of the left obstacle distance of the FPSO-ABOM module.

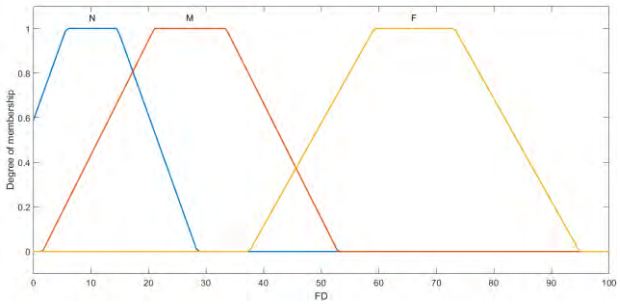

FIGURE 35. Enhanced membership of the front obstacle distance of the FPSO-ABOM module.

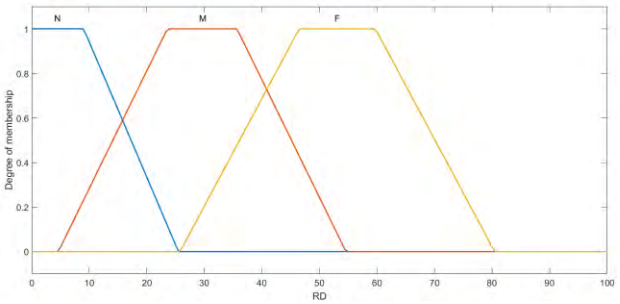

FIGURE 36. Enhanced membership of the right obstacle distance of the FPSO-ABOM module.

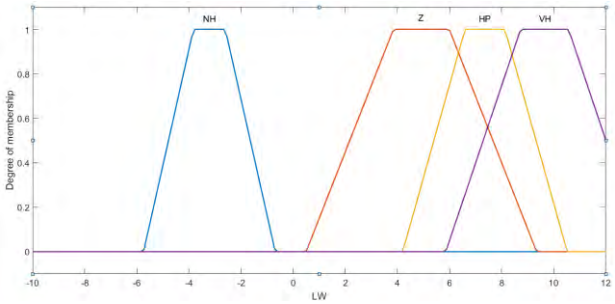

FIGURE 37. Enhanced membership of the LV of the AOBM fuzzy- PSO module.

Note that the shapes of the membership functions of the FPSO-NBM and AOBM are enhanced but still have many issues, such as overlap, gaps and a result that is not an optimal path.

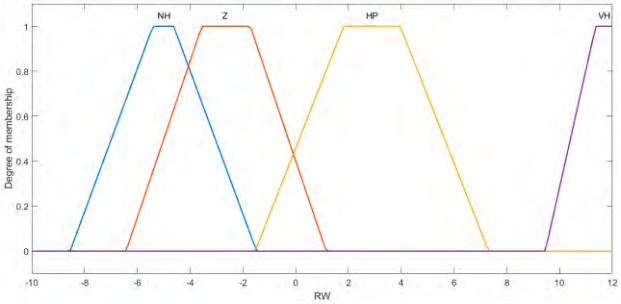

FIGURE 38. Enhanced membership of the RV of the AOBM fuzzy-PSO module.

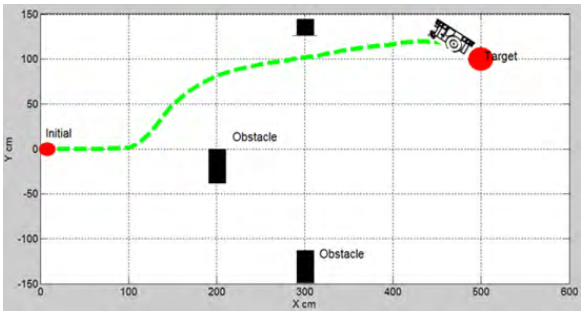

FIGURE 39. Path of the FPSO-AOBM.

D. EXPERT MODULES

In this part of the research, an expert was asked to tune the membership functions of the Naive AOBM and AOBM modules.

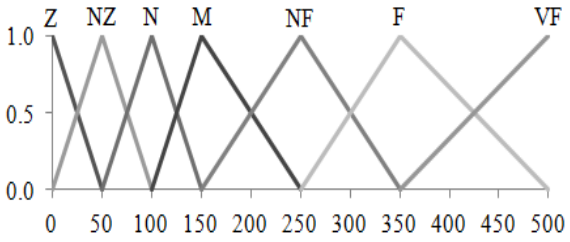

FIGURE 40. Enhanced membership of the distance of E-NBM.

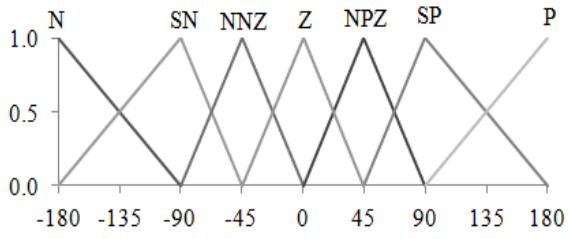

FIGURE 41. Enhanced membership of the angle of E-NBM.

1) EXPERT MODULE OF NAIVE NBM (E-NBM)

To enhance the NBM, we used an expert to tune the membership functions of the NBM. Both E-NBM and Naive NBM have similar inputs, outputs, and fuzzy rules. Figures 40 to 42 illustrate the enhanced membership functions of both the inputs and outputs.

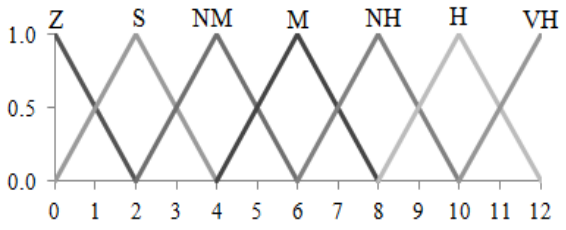

FIGURE 42. Enhanced membership of the LV and RV of E-NBM.

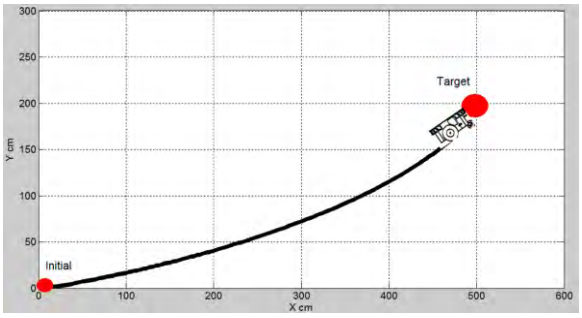

FIGURE 43. Path of the Expert NBM.

The E-NBM was applied to the PowerBot mobile robot from the initial point (0, 0) to the target point (500 cm, 200 cm); the result is illustrated in Figure 43.

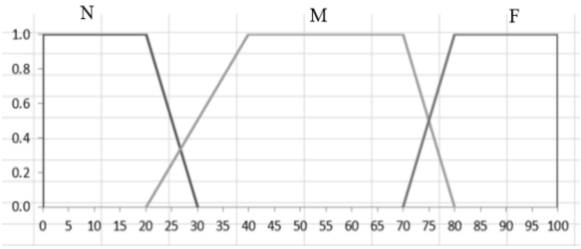

FIGURE 44. Enhanced membership of the left, front and right obstacle distance of E-AOBM.

2) EXPERT STRATEGY FOR THE NAIVE AOBM (E-AOBM)

The expert tuned the membership functions of the AOBM. Both E-AOBM and Naive AOBM have similar inputs, outputs, and fuzzy rules. Figures 44 and 45 illustrate the enhanced membership functions of the inputs and outputs.

III. PERFORMANCE INDEXES

We compare the naive FLC, GA, PSO, and FLC-Expert modules of both the NBM and AOBM through different scenarios as follows:

Each module was executed more than ten times for the same configuration and same points ((0 m, 0 m) to (5 m cm, 2 m).

The comparison included the execution time, the fitness value, the path used, and the traveled distance in meters. Plots of the results from the first experimental scenario are shown

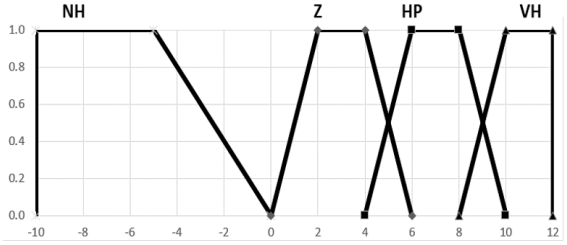

FIGURE 45. Enhanced membership of the LV and RV of the E-AOBM.

in Figure 46. Detailed performance indicators are shown in Table 3.

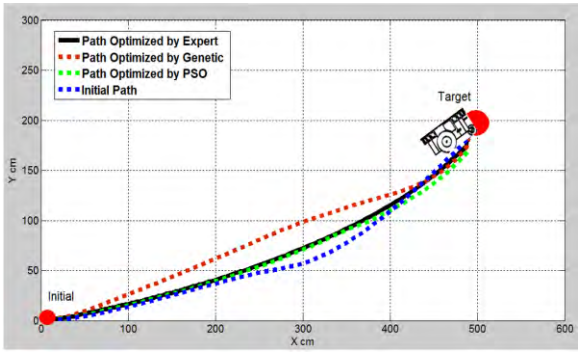

FIGURE 46. First scenario: Traveled paths of the Naive, FG, F-PSO and E-NBM.

TABLE 3. First scenario: Performance indexes of the Naive, FG, F-PSO, and Expert modules.

	Expert	Naive	FG	F-PSO
Execution time (s)	16.2	43.9	21.6	28.7
Fitness	-	-	16.8	17.1
Traveled distance (cm)	523	541	525	524

The statistical results of the NBM for the traveled distances and execution times of the first scenario for ten random selected experiments are listed in Table 4.

All the modules succeeded in navigating the first scenario with different execution times and travel distances.

In the second scenario, the working environment was changed to include objects to be avoided by the four modules of the NBM and AOBM. Each module was executed several times from the same startup point (0 cm, 0 cm) to the target point (500 cm, 100 cm). A plot of the results from one randomly selected experiment is shown in Figure 47, and the related performance indexes are listed in Table 5.

In the second scenario, three modules succeeded in the navigation phase, i.e., moving from the initial point to the

TABLE 4. Statistical data of all NBM modules.

	Attribute	Expert	Naive	FG	F-PSO
Execution time (s)	Min	16.2	42.8	21.1	28.7
	Max	18.6	44.3	22.9	29.7
	Mean±Std	18±0.7	43.7±0.5	21.9±0.59	29.1±0.3
Traveled distance (cm)	Min	522	541	522	524
	Max	525	549	527	534
	Mean±Std	523.4±0.8	544.5±2.5	524.1±1.7	529.1±3.7

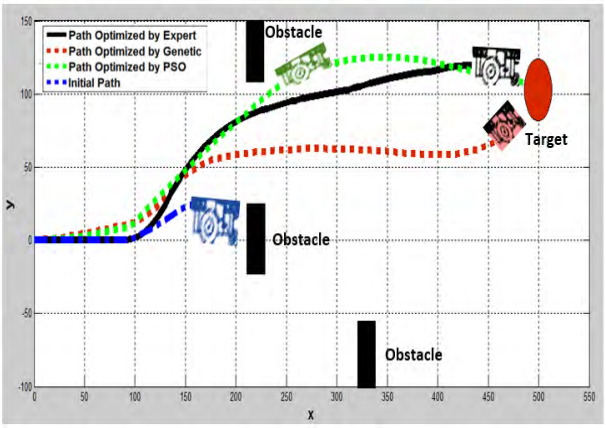

FIGURE 47. Second scenario: Traveled paths of the Naive, FG, F-PSO and Expert modules of NBM and AOBM.

TABLE 5. Second scenario: Performance indexes of the Naive, FG, F-PSO, and Expert modules.

	Expert	Naive	FG	F-PSO
Execution time (sec)	17	25	27.1	25.4
Fitness	-	-	24.7	17.9
Traveled distance (cm)	528	Failed	531	542

target using the obstacle avoidance strategy. However, each one had a different execution time and a different travel distance. All the issues related to this scenario are discussed in the next section.

In addition to the previous simple scenarios, the four modules of NBM and AOBM were tested in a third, more complex scenario. To determine the performance of both the NBM and AOBM, each module was executed several times from the same starting point (0 cm, 0 cm) through the intermediate points (4.7 m, −2 m), (4.8 m, −4.5 m) to the target point (6.5 m, −4.8 cm), as shown in Figure 48. Detailed performance indexes are listed in Table 6.

In the third scenario, only the expert module succeeded in avoiding all obstacles and passing through the door toward

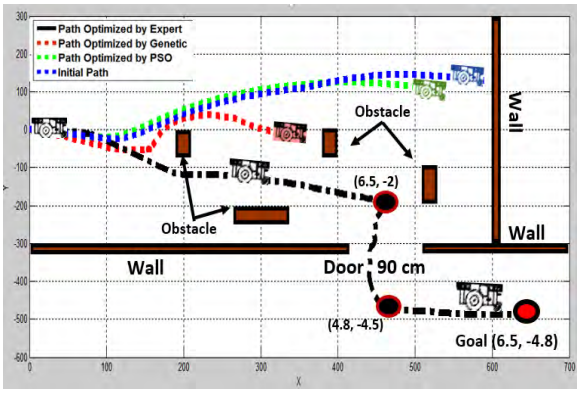

FIGURE 48. Third scenario of the Naive, FG, F-PSO and FLC-Expert modules of the NBM and AOBM.

TABLE 6. Third scenario: Performance indexes of the Naive, FG, F-PSO, and FLC-Expert modules.

	Expert	Naive	FG	F-PSO
Execution time (s)	45	25	32	21
Fitness	-	-	16.8	17.1
Traveled distance (cm)	1025	612	422	576

the target. Related discussions will be covered in the analysis section.

IV. Analysis

Based on the experimental results obtained for the integration of FL, GA, PSO, and the expert for navigation and object avoidance, we have focused our research direction on four performance indexes: the membership tuning time, the distance traveled, object avoidance scenarios, and the time to reach the target.

A. MEMBERSHIP TUNING TIME

The tuning phase using FG and PSO required two to four hours. This is 91.7% slower than the intelligent algorithms, but faster when compared to the expert, which required two to four days. Note that the expert is costly and not available at all times.

B. TRAVELED DISTANCE

In the first scenario, which is illustrated in Figure 46 and Table 3, all of the modules succeeded in the navigation phase, i.e., moving from the initial point to the target. It was concluded that all of the modules are competitive with regards to navigation, but in terms of execution time and traveled distance, the expert prevailed in the competition. Recall that the first scenario was executed more than fifty times; the best average time was 18 sec and the best distance traveled was 523.4 cm, as shown in Table 4.

In the second scenario, the shortest distance traveled was attained using the human expert. This is because the expert has a global view of the map and performs fine tuning based

on minimizing the length of the path through the membership functions.

In the third scenario, only the expert could reach the target because the membership functions were tuned based on the local and global errors; each tuning required many experiments. This tuning improved the behavior of the robot because the results can be reused for any other complex scenario with different initial and target points.

C. SIMPLE OBJECT AVOIDANCE

In the second scenario shown in Figure 47 and Table 5, all of the modules succeeded in avoiding obstacles, except the naive module. However, each method had a different execution time and travel distance, as listed in Table 5.

D. COMPLEX OBJECT AVOIDANCE

The complex scenario was evaluated to determine if all the modules could optimize their path by avoiding obstacles and still focus on the task of reaching the target point. Unfortunately, only the FLC-Expert could complete the task; the other methods (FL, F-PSO) diverged after the first obstacle, as seen in Figure 48. This is mainly due to the overlaps and gaps within the membership functions, which mislead the robot controller in complex situations.

E. TIME TO ARRIVAL TO TARGET

Distance is linearly related to time for a fixed speed, but both motors of the Scout-II do not have the same speed and are not synchronized with a preset velocity. Thus, every module had full control of each motor.

In all scenarios, the FLC-Expert had the best time for arriving at the target because the speed was better tuned, and both motors had an optimal control strategy.

The results of the analysis are summarized in Table 7.

TABLE 7. Comparison between the Naive, FG, F-PSO and FLC-Expert modules.

	Naive	FG	F-PSO	Expert
Tuning time	1 Hour	2-4 Hours	2-4 Hours	2-4 Days
Availability	Available	Available	Available	Not available
NBM First scenario	Succeeded	Succeeded	Succeeded	Succeeded
Obstacle avoidance	Failed	Succeeded	Succeeded	Succeeded
Complex scenario	Failed	Failed	Failed	Succeeded

V. CONCLUSIONS

Mobile navigation through unknown and complex environments is still an open area of research, either in the controls field or in multi-sensor fusion. The approach taken in this paper targeted a combination of different algorithms, including fuzzy-logic, PSO, and genetic algorithms towards different decision making approaches. The goal was to identify the optimal navigation path, both with and without the constraint of obstacle avoidance, based on sensor data acquired from a laser rangefinder and an array of ultrasonic sensors on the PowerBot navigation robot platform.

The experimental results showed that the Naive, FG, F-PSO and Expert modules are competitive for navigation and simple obstacles avoidance. However, FG and F-PSO failed in complex obstacle avoidance scenarios. The alternative using an expert generated better results. Thus, designing a system with the aid of an expert can provide intelligent algorithms at the expense of availability and cost.

REFERENCES

[1] B. Graf, U. Reiser, M. Hägele, K. Mauz, and P. Klein, "Robotic home assistant care-O-bot 3-product vision and innovation platform," in *Proc. IEEE Workshop Adv. Robot. Soc. Impacts (ARSO)*, Nov. 2009, pp. 139–144.

[2] F. Matsuno, N. Sato, K. Kon, H. Igarashi, T. Kimura, and R. Murphy, "Utilization of robot systems in disaster sites of the great eastern Japan earthquake," in *Proc. 8th Int. Conf. Field and Service Robotics*, 2014, pp. 1–17.

[3] M. Gupta, S. Kumar, L. Behera, and V. K. Subramanian, "A novel vision-based tracking algorithm for a human-following mobile robot," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1415–1427, Jul. 2017.

[4] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: A review," *Int. Robot. Auto. J.*, vol. 2, p. 00022, 2017.

[5] X. Li, W. Pedrycz, and D. Ralescu, *Optimization Under Uncertainty: A Perspective of Soft Computing*. Amsterdam, The Netherlands: Elsevier, 2017.

[6] M. Cao and E. L. Hall, "Fuzzy logic control for an automated guided vehicle," *Proc. SPIE*, vol. 3522, pp. 303–312, Oct. 1998.

[7] R. Rashid, I. Elamvazuthi, M. Begam, and M. Arrofiq, "Differential drive wheeled mobile robot (WMR) control using fuzzy logic techniques," in *Proc. 4th Asia Int. Conf. Math./Anal. Modelling Comput. Simulation (AMS)*, 2010, pp. 51–55.

[8] M. Faisal, R. Hedjar, M. Al Sulaiman, and K. Al-Mutib, "Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 1, p. 37, 2013.

[9] V. Raudonis and R. Maskeliunas, "Trajectory based fuzzy controller for indoor navigation," in *Proc. IEEE 12th Int. Symp. Comput. Intell. Inf. (CINTI)*, Nov. 2011, pp. 69–72.

[10] T. H. S. Li, S.-J. Chang, and W. Tong, "Fuzzy target tracking control of autonomous mobile robots by using infrared sensors," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 491–501, Aug. 2004.

[11] G. Narvydas, R. Simutis, and V. Raudonis, "Autonomous mobile robot control using fuzzy logic and genetic algorithm," in *Proc. 4th IEEE Workshop Intelligent Data Acquisition Adv. Comput. Syst., Technol. Appl. (IDAACS)*, Nov. 2007, pp. 460–464.

[12] F. Cupertino, V. Giordano, D. Naso, and L. Delfine, "Fuzzy control of a mobile robot," *IEEE Robot. Autom. Mag.*, , vol. 13, no. 4, pp. 74–81, Dec. 2006.

[13] S. Wu, Q. Li, E. Zhu, J. Xie, and G. Zhichao, "Fuzzy controller of pipeline robot navigation optimized by genetic algorithm," in *Proc. Chin. Control Decision Conf. (CCDC)*, 2008, pp. 904–908.

[14] C. Rekik, M. Jallouli, and N. Derbel, "Integrated genetic algorithms and fuzzy control approach for optimization mobile robot navigation," in *Proc. 6Th Int. Multi-Conf. Syst., Signals Devices (SSD*, Jun. 2009, pp. 1–8.

[15] O. Castillo, J. Soria, H. Arias, J. B. Morales, and M. Inzunza, "Intelligent control and planning of autonomous mobile robots using fuzzy logic and multiple objective genetic algorithms," in *Analysis and Design of Intelligent Systems using Soft Computing Techniques*. Berlin, Germany: Springer, 2007, pp. 799–807.

[16] A. Adriansyah and S. H. M. Amin, "Learning of fuzzy-behaviours using particle swarm optimisation in behaviour-based mobile robot," *Int. J. Intell. Syst. Technol. Appl.*, vol. 5, no. 2, pp. 49–67, 2008.

[17] C. Wong, H. Wang, and S. Li, "PSO-based motion fuzzy controller design for mobile robots," *Int. J. Fuzzy Syst.*, vol. 10, no. 1, p. 284–292, 2008.
[18] R. Martínez-Marroquín, O. Castillo, and J. Soria, "Particle swarm optimization applied to the design of type-1 and type-2 fuzzy controllers for an autonomous mobile robot," in *Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition*. Berlin, Germany: Springer, 2009, pp. 247–262.
[19] M. Gupta, L. Behera, and K. Venkatesh, "PSO based modeling of Takagi-Sugeno fuzzy motion controller for dynamic object tracking with mobile platform," in *Proc. Int. Multiconf. Comput. Sci. Inf. Technol. (IMCSIT)*, 2010, pp. 37–43.
[20] N. A. Shiltagh and L. D. Jalal, "A comparative study: Modified particle swarm optimization and modified genetic algorithm for global mobile robot navigation," *Int. J. Comput. Appl.*, vol. 89, no. 9, pp. 32–46, 2014.

MOHAMMED FAISAL (M'16) received the M.S. degree in security of wireless sensor networks and the Ph.D. degree in robotics and artificial intelligence from the King Saud University. His current research interests include swarm intelligence, stereovision, robot navigation, and the security of wireless sensor networks.

MOHAMMED ALGABRI received the B.Sc. degree in computer science from Umm Al-Qura University, and the master's degree from the Department of Computer Science, King Saud University, in 2015, where he is currently pursuing the Ph.D. degree with the Computer Science Department, College of Computer and Information Science. He is currently with the Center of Smart Robotics Research, King Saud University. He has authored and co-authored more than 15 publications in journals and conferences. His research interest includes soft computing techniques, robotics, deep learning, and speech recognition.

BENCHERIF MOHAMED ABDELKADER received the Engineer degree in control from INELEC, Boumerdès, Algeria, in 1992, the master's degree in signals and systems in 2005, and the Ph.D. degree in the classification of remote sensing images from Université Saad Dahlab Blida, Blida, Algeria, in 2015. He was involved in diverse industrial projects, mainly on project management. He is currently at the Center of Smart Robotics Research, King Saud University. His areas of interest are robotic design, speech classification, and pattern recognition.

HABIB DHAHRI was born in Sidi Bouzid, Tunisia, in 1975. He received the degree in computer science in 2001, and the Ph.D. degree in computer engineering from the National Engineering School of Sfax in 2013. He is currently an Assistant Professor of computer science at King Saud University. His research interest includes computational intelligence: neural network, swarm intelligence, differential evolution, and genetic algorithm.

MOHAMAD MAHMOUD AL RAHHAL (S'14–M'17) received the B.Sc. degree in computer engineering from the University of Aleppo, Aleppo, Syria, in 2002, the M.Sc. degree from Hamdard University, New Delhi, India, in 2005, and the Ph.D. degree in computer engineering from King Saud University, Riyadh, Saudi Arabia, in 2015. From 2006 to 2012, he was a Lecturer with Al Jouf University, Sakakah, Saudi Arabia. Since 2015, he has been an Assistant Professor in computer science with King Saud University. His research interests include signal/image medical analysis, remote sensing, and computer vision.