CrossMark

Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm

Anish Pandey[*], Dayal R. Parhi

Mechanical Engineering Department, N.I.T., Rourkela, Orissa, India

ABSTRACT

This article introduces a singleton type-1 fuzzy logic system (T1-SFLS) controller and Fuzzy-WDO hybrid for the autonomous mobile robot navigation and collision avoidance in an unknown static and dynamic environment. The WDO (Wind Driven Optimization) algorithm is used to optimize and tune the input/output membership function parameters of the fuzzy controller. The WDO algorithm is working based on the atmospheric motion of infinitesimal small air parcels navigates over an N-dimensional search domain. The performance of this proposed technique has compared through many computer simulations and real-time experiments by using Khepera-III mobile robot. As compared to the T1-SFLS controller the Fuzzy-WDO algorithm is found good agreement for mobile robot navigation.

© 2017 Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

'Path planning and control' of an autonomous mobile robot in an unknown dynamic environment is one of the most challenging jobs. Fuzzy logic is a mimic of human behavior, which easily handles the system uncertainty. One of the most cited methods in the field of the mobile robot is the fuzzy logic. Soft computing techniques such as fuzzy logic [1], neural network [2], neuro-fuzzy [4] and nature-inspired algorithms (Genetic Algorithm [8], Particle Swarm Optimization [12,13], Ant Colony Algorithm [10,11], Simulated Annealing Algorithm [14,15], Bacterial Foraging Optimization [5]) are widely used for mobile robot navigation. However, each method (algorithm) has its strengths and weaknesses.

The motion control problem of an autonomous wheeled mobile robot has been widely investigated in past two decades. Abadi and Khooban [1] have introduced Mamdani-type fuzzy logic controller integrated with random inertia weight Particle Swarm Optimization (RNW-PSO) for optimal path tracking of wheeled mobile robots (WMRs). Algabri et al. [2] have combined the fuzzy logic with

other soft computing techniques such as Genetic Algorithm (GA), Neural Networks (NN), and Particle Swarm Optimization (PSO) for optimizing the membership function parameters of the fuzzy controller to improve the navigation performance of the mobile robot. A comparative study between two soft computing approaches, namely genetic-fuzzy and genetic-neural and the conventional potential field method have been designed and developed by Hui and Pratihar [3] for an adaptive navigation planning of a car-like mobile robot moving in the presence of some dynamic obstacles. Pothal and Parhi [4] have proposed the sensor based Adaptive Neuro Fuzzy Inference System (ANFIS) controller for navigation of single and multiple mobile robots in the highly cluttered environment.

Montiel et al. [5] and Hossain et al. [6] have explored the application of Bacterial Foraging Optimization (BFO) method in the field of mobile robot navigation to find out the shortest possible path within the minimum time to move from any start position to the goal position in an unknown environment between moving obstacles. Baturone et al. [7] have designed the low-cost embedded neuro-fuzzy controller for safe and collision-free navigation of an autonomous car-like robot among possible obstacles toward a goal configuration. Ming et al. [8] have designed a genetic algorithm to select the best membership functions from the fuzzy system to control a mobile robot in the partially unknown environment. Liang

* Corresponding author.
  E-mail addresses: anish06353@gmail.com (A. Pandey), dayalp3@yahoo.in (D.R. Parhi).

et al. [9] have presented the kinematic modeling of the two-wheeled differential drive mobile robot. Purian and Sadeghian [10] have explored the optimal path for a mobile robot in an unknown environment using Ant Colony Optimization (ACO) algorithm.

To prepare an optimal intelligent controller for an autonomous wheeled mobile robot, the Castillo et al. [11] have designed the hybridization of an ACO algorithm and the PSO algorithm to optimize the membership function of a fuzzy controller. Chung et al. [12] have developed PSO and fuzzy control algorithm to navigate the robot in the unknown environment. Allawi and Abdalla [13] have proposed the sensor based PSO-fuzzy model for the navigation of multiple mobile robots. Where, the PSO is used to determine the optimal input/output membership functions and the optimal rules for the fuzzy type-2 controllers. Yanar and Akyurek [14] have proposed the use of simulated annealing metaheuristic algorithm for tuning the Mamdani type fuzzy models. Martinez-Alfaro and Gomez-Garcia [15] have developed the simulated annealing and fuzzy logic for generating an automatic path planning of the mobile robot. Mohanty and Parhi [20] have combined the cuckoo search algorithm with ANFIS for optimizing the navigation path length of mobile robots. Wong et al. [21] have used the PSO algorithm to tune the parameters of the membership function.

One major problem with the fuzzy logic is the difficulty of constructing and tuning the correct membership function grade [22]. Therefore, the authors have tried to attempt to solve this problem by using WDO algorithm. In this article, a Fuzzy-WDO hybrid algorithm has been presented for mobile robot navigation and collision avoidance in an unknown static and dynamic environment. The WDO is integrated with the fuzzy controller to adjust and optimize the antecedent and consequent parameters of the generalized bell-shaped membership function. The WDO [16–18] method is a population-based iterative heuristic global optimization algorithm for multi-dimensional and multi-model problems with the potential to implement constraints on the search domain. This algorithm works by simultaneously maintaining several infinitesimal small air parcels or potential solutions in the search domain. For each iteration of the algorithm, each air parcels are evaluated by the membership function parameters (objective function) being optimized based on the fitness function of that solution. The primary objective of this research is to optimize the membership function parameters of the fuzzy controller by using WDO algorithm.

This article is organized into seven sections. Section 1 presents the introduction and literature review. T1-SFLS controller for mobile robot navigation is proposed in Section 2. The hybrid fuzzy-WDO algorithm for mobile robot navigation is presented in Section 3. Section 4 demonstrates the simulation results of the mobile robot in different environments. Section 5 describes the simulation result comparison with previous works. Section 6 presents the experimental results and discussion for validating the proposed controller. Finally, Section 7 depicts the summary.

2. T1-SFLS controller for the mobile robot navigation

In this section, the T1-SFLS rule-based controller has been designed and implemented for mobile robot navigation and
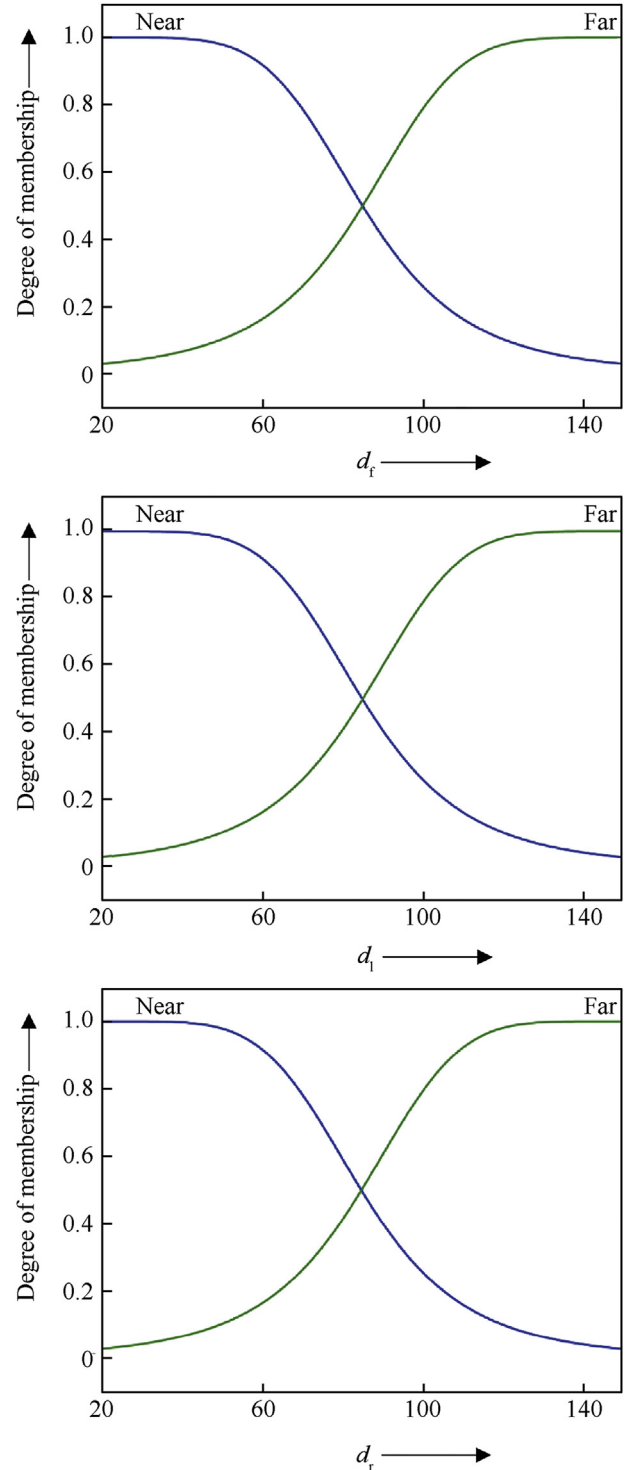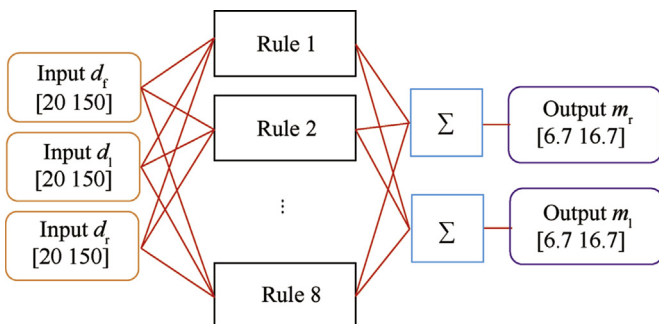

Fig. 1. The structure of a T1-SFLS controller for mobile robot navigation.

Fig. 2. Fuzzy membership functions for the inputs (d_f, d_l, and d_r).
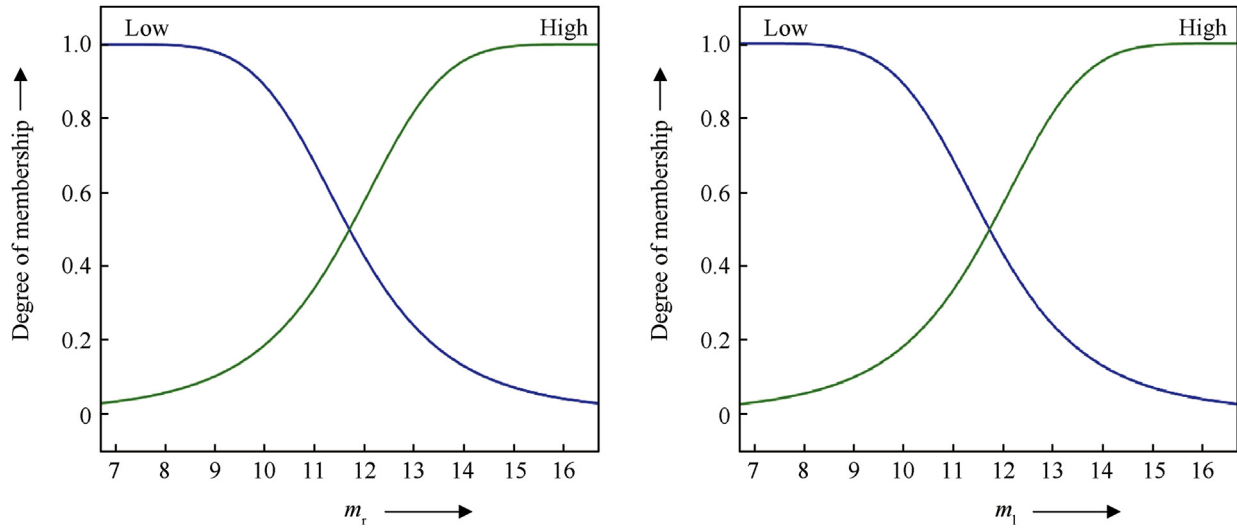
Fig. 3. Fuzzy membership functions for the outputs (m_r, and m_l).

Table 1
Fuzzy rules set.

If (d_f is Far) **and** (d_l is Far) **and** (d_r is Far) **then** (m_r is High) **and** (m_l is Low)
If (d_f is Near) **and** (d_l is Near) **and** (d_r is Near) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Far) **and** (d_l is Near) **and** (d_r is Far) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Far) **and** (d_l is Far) **and** (d_r is Near) **then** (m_r is High) **and** (m_l is Low)
If (d_f is Near) **and** (d_l is Far) **and** (d_r is Far) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Near) **and** (d_l is Near) **and** (d_r is Far) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Near) **and** (d_l is Far) **and** (d_r is Near) **then** (m_r is High) **and** (m_l is Low)
If (d_f is Far) **and** (d_l is Near) **and** (d_r is Near) **then** (m_r is Low) **and** (m_l is High)

Table 2
Adjusting parameters of the inputs before optimization.

Inputs	Membership function	a	b	c
d_f	Near	65	2.5	20
	Far	65	2.5	150
d_l	Near	65	2.5	20
	Far	65	2.5	150
d_r	Near	65	2.5	20
	Far	65	2.5	150

Table 3
Adjusting parameters of the outputs before optimization.

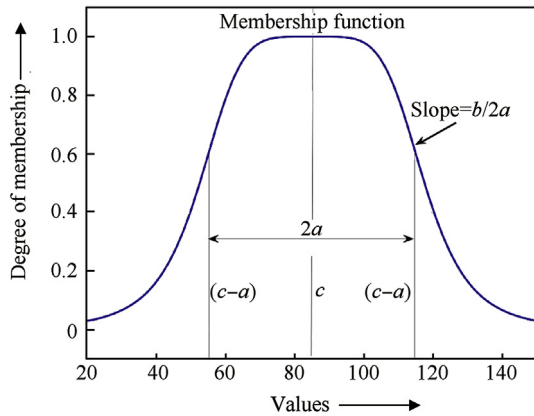| Outputs	Membership function	a	b	c
m_r	Low	5	2.5	6.7
	High	5	2.5	16.7
m_l	Low	5	2.5	6.7
	High	5	2.5	16.7

Fig. 4. The general structure of the generalized bell-shaped membership function.

collision avoidance in an unknown static and dynamic environment. The proposed controller controls the right motor velocity and left motor velocity of the mobile robot using sensory data interpretation. The T1-SFLS controller has three inputs: Forward Obstacle Distance (d_f), Left Forward Obstacle Distance (d_l) and Right Forward Obstacle Distance (d_r); and two outputs: Right Motor Velocity (m_r) and Left Motor Velocity (m_l), which are logically connected by eight rules (see Fig. 1). The T1-SFLS controller receives inputs (obstacle distances) from the front, left, and the right group of sensors of the robot, and output from T1-SFLS controller is right motor velocity and left motor velocity of the mobile robot. These sensors read the obstacle from 20 cm to 150 cm approximately. The input and output variables of the controller are

illustrated in Figs. 2 and 3, respectively. The fuzzy rule set of the T1-SFLS controller is described in Table 1. The two generalized bell-shaped (Gbell) membership functions are used for inputs and outputs. The range of inputs is divided into two linguistic variables: Near and Far. These inputs are located at 20 cm−150 cm. Similarly, the two Gbell membership functions (MFs) Low and High respectively have been used for the outputs, and it is located at 6.7 cm/s to 16.7 cm/s. The designed T1-SFLS controller is directly implemented in the mobile robot for simulations and experiments. The T1-SFLS controller is composed through Mamdani-type fuzzy model in the following form

$$Rule_n : \; If \; d_f \; is \; d_{f(i)}, \; d_l \; is \; d_{l(j)}, \; and \; d_r \; is \; d_{r(k)}$$

$$Then \; m_r \; is \; m_{r(ijk)} \; and \; m_l \; is \; m_{l(ijk)}$$

where $n = 1, 2, …, 8$ (eight rules), the $i = 1, 2, j = 1, 2$ and $k = 1, 2$ because d_f, d_l and d_r have two Gbell membership functions each. The $d_{f(i)}$, $d_{l(j)}$, and $d_{r(k)}$ are the fuzzy sets of the inputs d_f, d_l, and d_r respectively. Similarly, the $m_{r(ijk)}$, and $m_{l(ijk)}$ are the fuzzy sets of the outputs m_r, and m_l respectively. The fuzzy set (inputs and outputs) uses the following Gbell membership function.
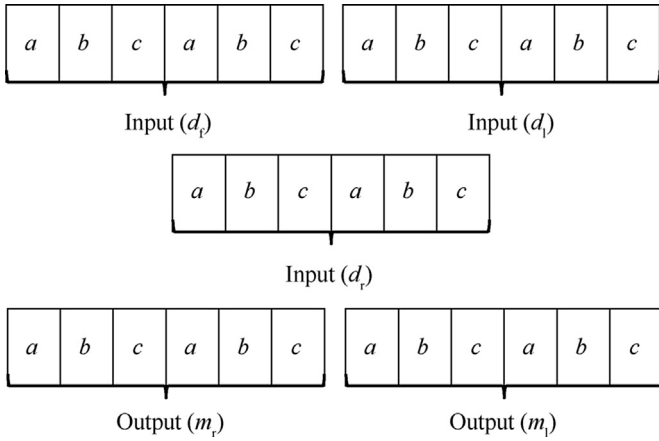
Fig. 5. Air parcels representation of the WDO algorithm.

Fig. 6. Fuzzy membership functions for the inputs (d_f, d_l, and d_r) after optimization.

Let d_f, d_l, and d_r are presented by x_1, x_2, and x_3 respectively. Similarly, m_r, and m_l are denoted by y_1, and y_2 respectively.

$$\mu_{n1}(x_1) = \frac{1}{1 + \left|\frac{x_1 - c_{n1}}{a_{n1}}\right|^{2b_{n1}}} \quad (2)$$

$$\mu_{n2}(x_2) = \frac{1}{1 + \left|\frac{x_2 - c_{n2}}{a_{n2}}\right|^{2b_{n2}}} \quad (3)$$

$$\mu_{n3}(x_3) = \frac{1}{1 + \left|\frac{x_3 - c_{n3}}{a_{n3}}\right|^{2b_{n3}}} \quad (4)$$

$$\mu_{n1}(y_1) = \frac{1}{1 + \left|\frac{y_1 - c_{n1}}{a_{n1}}\right|^{2b_{n1}}} \quad (5)$$

$$\mu_{n2}(y_2) = \frac{1}{1 + \left|\frac{y_2 - c_{n2}}{a_{n2}}\right|^{2b_{n2}}} \quad (6)$$

where a, b, and c are adjusting parameters of the membership function; called as the half width, slope control, and center respectively. The general structure of the generalized bell-shaped membership function is shown in Fig. 4.

The defuzzification of the outputs (y_1 and y_2) are accomplished by the weighted average method

$$y_1 = \frac{\sum_{n=1}^{8}(\mu_{n1}(x_1) \cdot \mu_{n2}(x_2) \cdot \mu_{n3}(x_3)) \cdot y_1}{\sum_{n=1}^{8}(\mu_{n1}(x_1) \cdot \mu_{n2}(x_2) \cdot \mu_{n3}(x_3))} \quad (7)$$

$$y_2 = \frac{\sum_{n=1}^{8}(\mu_{n1}(x_1) \cdot \mu_{n2}(x_2) \cdot \mu_{n3}(x_3)) \cdot y_2}{\sum_{n=1}^{8}(\mu_{n1}(x_1) \cdot \mu_{n2}(x_2) \cdot \mu_{n3}(x_3))} \quad (8)$$

The adjusting parameters a, b, and c of the inputs and outputs are listed in Table 2 and Table 3, respectively, which will be optimized through the WDO algorithm in Section 3 below.
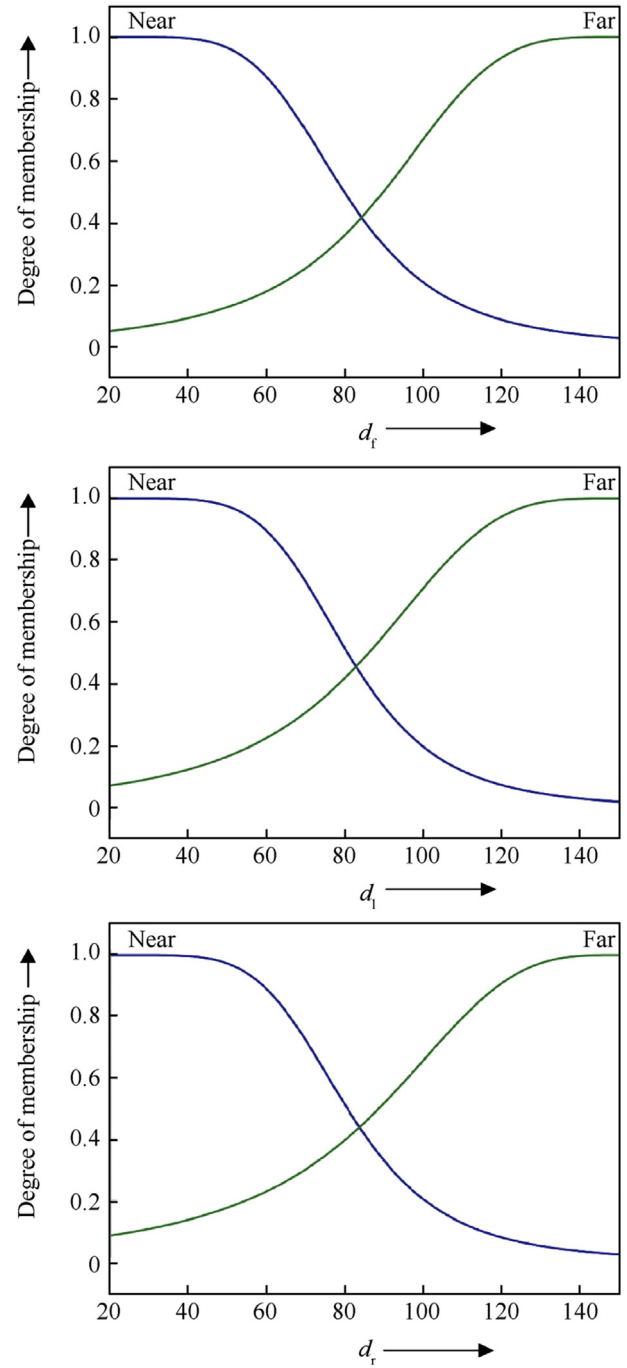
3. Fuzzy-WDO algorithm for the mobile robot navigation

WDO [16] algorithm is inspired by the earth's atmosphere, where the wind blows are trying to equalize the horizontal imbalance in the air pressure. WDO is a new type nature-inspired global optimization based on atmospheric motion developed by Bayraktar et al. [16] in 2013. This method is working on the population-based iterative heuristic global optimization algorithm for multi-dimensional and multi-modal problems with the potential to implement constraints on the search domain. WDO is similar to other nature-inspired optimization algorithms, in which population-based heuristic iterative process can be found for solving multi-dimensional optimization problems [18]. At its
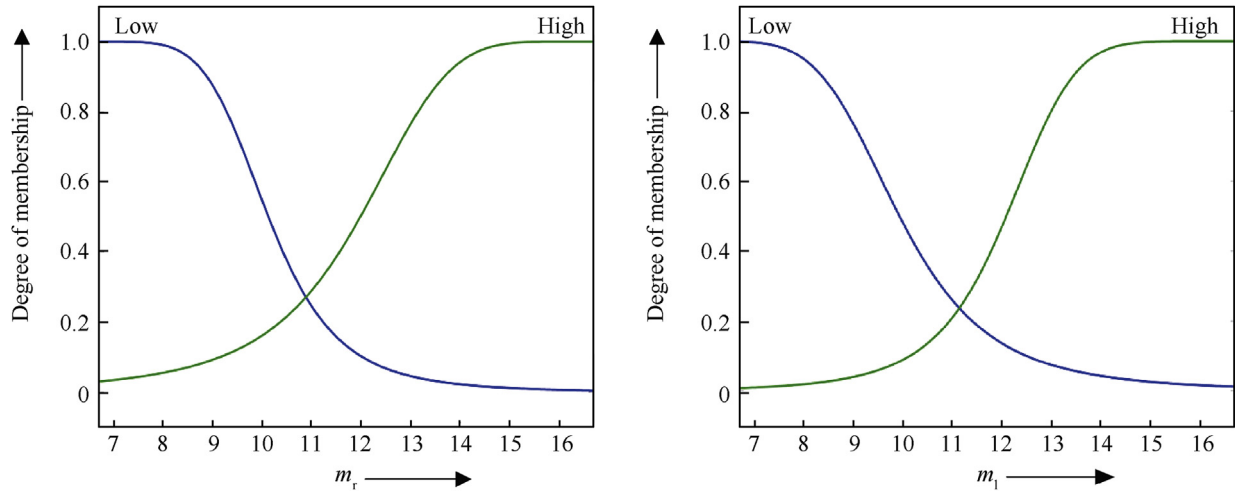
Fig. 7. Fuzzy membership functions for the outputs (m_r, and m_l) after optimization.

Table 4
Adjusting parameters of the inputs after optimization.

Inputs	Membership function	a	b	c
d_f	Near	55.11	2.14	25
	Far	59.6	1.88	149.4
d_l	Near	58.3	2.44	22.4
	Far	62.41	1.76	148.3
d_r	Near	57.42	2.33	23.1
	Far	60.29	1.55	148.9

Table 5
Adjusting parameters of the outputs after optimization.

Outputs	Membership function	a	b	c
m_r	Low	3.61	2.601	6.515
	High	4.22	2.14	16.2
m_l	Low	3.97	2.21	5.96
	High	4.32	2.96	16.4

center, a population of infinitesimally small air parcels navigates over an N-dimensional search space, employing Newton's second law of motion that is used to express the motion of air parcels inside the earth's atmosphere. As compared to other particle based optimization algorithm (e.g., PSO), the WDO algorithm has additional terms in the velocity update equation such as Gravitation and Coriolis forces, which provides robustness and extra degrees of freedom to the algorithm.

The WDO algorithm is working based on the atmospheric motion of infinitesimal small air parcels navigating over an N-dimensional search domain. The starting step of this algorithm is supported by the Newton's second law of motion, which provides accurate results when applied to the analysis of atmospheric motion. It states that the total force applied on an air parcel causes it to accelerate with an acceleration a in the same direction as the applied total force.

$$\rho \cdot a = \sum F_i \tag{9}$$

where ρ is the density of air for an infinitesimally small air parcel, and F_i represents all the individual forces acting on the air parcel. To relate the air pressure to the air parcel's density and temperature, the ideal gas law can be utilized and is given by
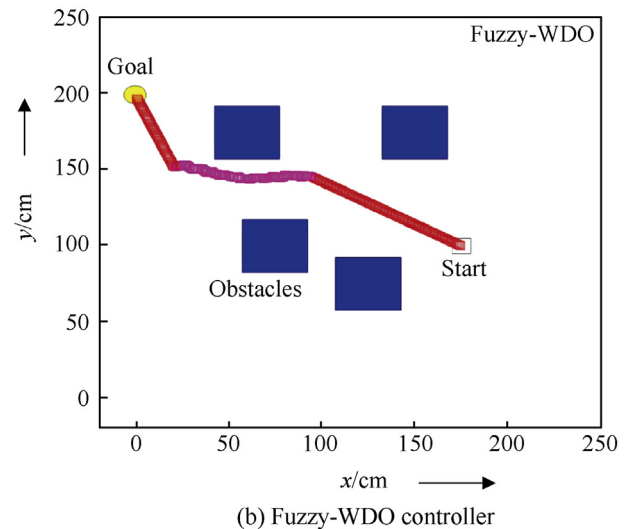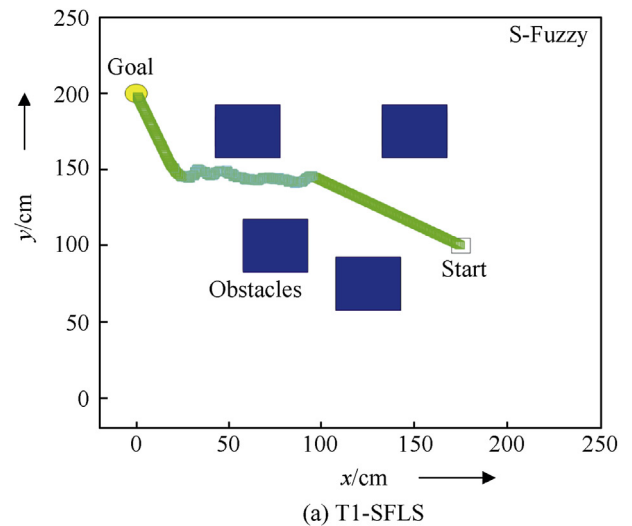

(a) T1-SFLS

(b) Fuzzy-WDO controller

Fig. 8. Mobile robot navigation between the obstacles using **(a)** T1-SFLS and **(b)** Fuzzy-WDO controller.

(a) T1-SFLS

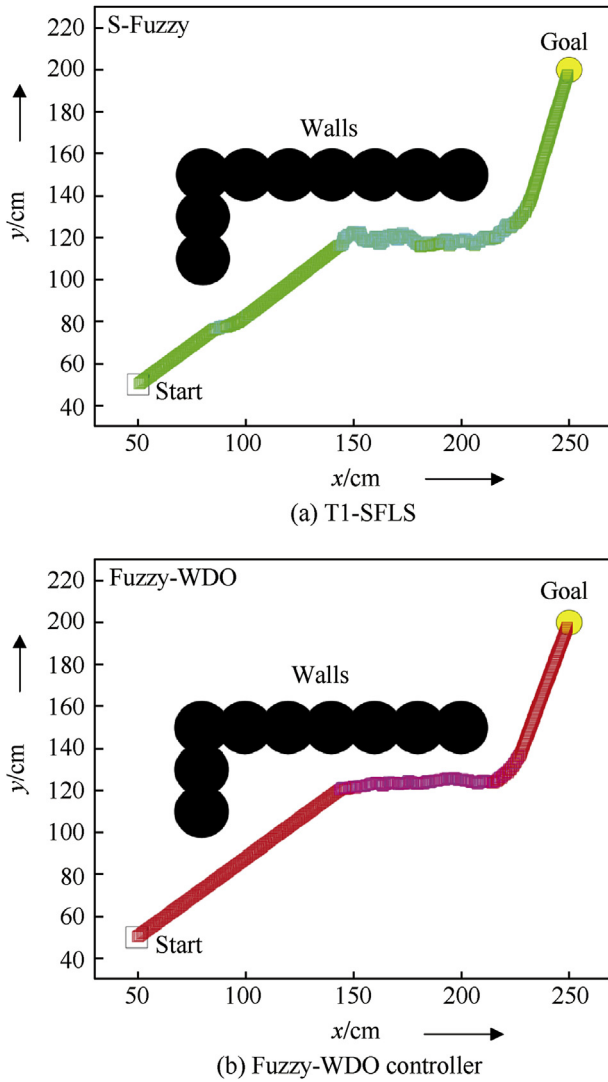(b) Fuzzy-WDO controller

Fig. 9. Mobile robot navigation between the walls using **(a)** T1-SFLS and **(b)** Fuzzy-WDO controller.

$$P = \rho RT \tag{10}$$

where P is the pressure, R is the universal gas constant, and T is the temperature.

Four major forces can be included in equation (9) that either cause the wind to move in a certain direction at a certain velocity or that deflect it from its existing path. The most observable force causing the air to move is the pressure gradient force F_{PG} defined in equation (11). Another force is the friction force F_F defined in equation (12), which simply acts to oppose the motion started by the pressure gradient force. In our three-dimensional physical atmosphere, the gravitational force F_G in equation (13) is a vertical force directed toward the earth's surface. The Coriolis force F_C in equation (14) is caused by due to the rotation of the earth and deflects the path of the wind from one dimension to another.

$$F_{PG} = -\nabla P \cdot \delta V \tag{11}$$

$$F_F = -\rho \cdot \alpha \cdot u \tag{12}$$

$$F_G = \rho \cdot \delta V \cdot g \tag{13}$$

$$F_C = -2 \cdot \Omega \cdot u \tag{14}$$

where, ∇P is the pressure gradient, δV represents the infinite air volume, Ω represents the rotation of the earth, g is the gravitational acceleration, α is the friction coefficient and u is the velocity vector of the wind.

The sum of all forces (gravitational force, pressure gradient force, friction force, and Coriolis force) described above can be entered on the right-hand side of Newton's second law of motion given in equation (9), which leads to

$$\rho \cdot \frac{\Delta u}{\Delta t} = (\rho \cdot \delta V \cdot g) + (-\nabla P \cdot \delta V) + (-\rho \cdot \alpha \cdot u) + (-2 \cdot \Omega \cdot u) \tag{15}$$

where the acceleration term in equation (9) is rewritten as $a = \Delta u/\Delta t$, and a time step $\Delta t = 1$ is assumed for simplicity. For an infinitesimally small, dimensionless air parcel, the volume is set as $\delta V = 1$, which simplifies equation (15) to

$$\rho \cdot \Delta u = (\rho \cdot g) + (-\nabla P) + (-\rho \cdot \alpha \cdot u) + (-2 \cdot \Omega \cdot u) \tag{16}$$

Putting the ideal gas law equation (10) in equation (16), the density ρ can be written in terms of the pressure, with temperature and the universal gas law constant

$$u_{new} = (1 - \alpha) \cdot u_{cur} - g \cdot x_{cur} + \left(RT \left| \frac{1}{i} - 1 \right| (x_{opt} - x_{cur}) \right)$$
$$+ \left(\frac{c \cdot u_{cur}^{other\ dim}}{i} \right) \tag{17}$$

where u_{new} is the velocity in the next iteration, u_{cur} is the velocity in current iteration, x_{cur} is the current location of the air parcel, x_{opt} is the optimum location of the air parcel, i represents the ranking between all air parcels, $u_{cur}^{other\ dim}$ is the velocity influence from another randomly chosen dimension of the same air parcel, and all other coefficients are combined into a single term c (i.e., $c = -2 \cdot \Omega \cdot RT$). Equation (17) represents the final form of the velocity update utilized in WDO [16,19]. The following function updates the position of the air parcel

$$x_{new} = x_{cur} + (u_{new} \cdot \Delta t) \tag{18}$$

where x_{new} is the new position of the air parcel in the next iteration. If the new velocity u_{new} exceeds the initialize maximum velocity ($u_{max} = 0.3$) in any dimension, then the velocity in that dimension is limited according to the following condition

$$u_{new}^* = \begin{cases} u_{max} & \text{if } u_{new} > u_{max} \\ -u_{max} & \text{if } u_{new} < -u_{max} \end{cases} \tag{19}$$

where the direction of motion is preserved but the magnitude is limited to be no more than $|u_{max}|$ at any dimension and u_{new}^* represents the adjusted velocity after it is limited to the maximum speed.

The pseudo-code of the WDO algorithm can be summarized as follows:

Step 1. Start.
Step 2. Initialize the population size (i.e., group of air parcels), number of dimensions of the optimization problem, maximum number of iterations, coefficients (such as RT, g, α, c, u_{max}), pressure function (fitness function of the optimization problem), lower and upper boundaries of the optimization problem.
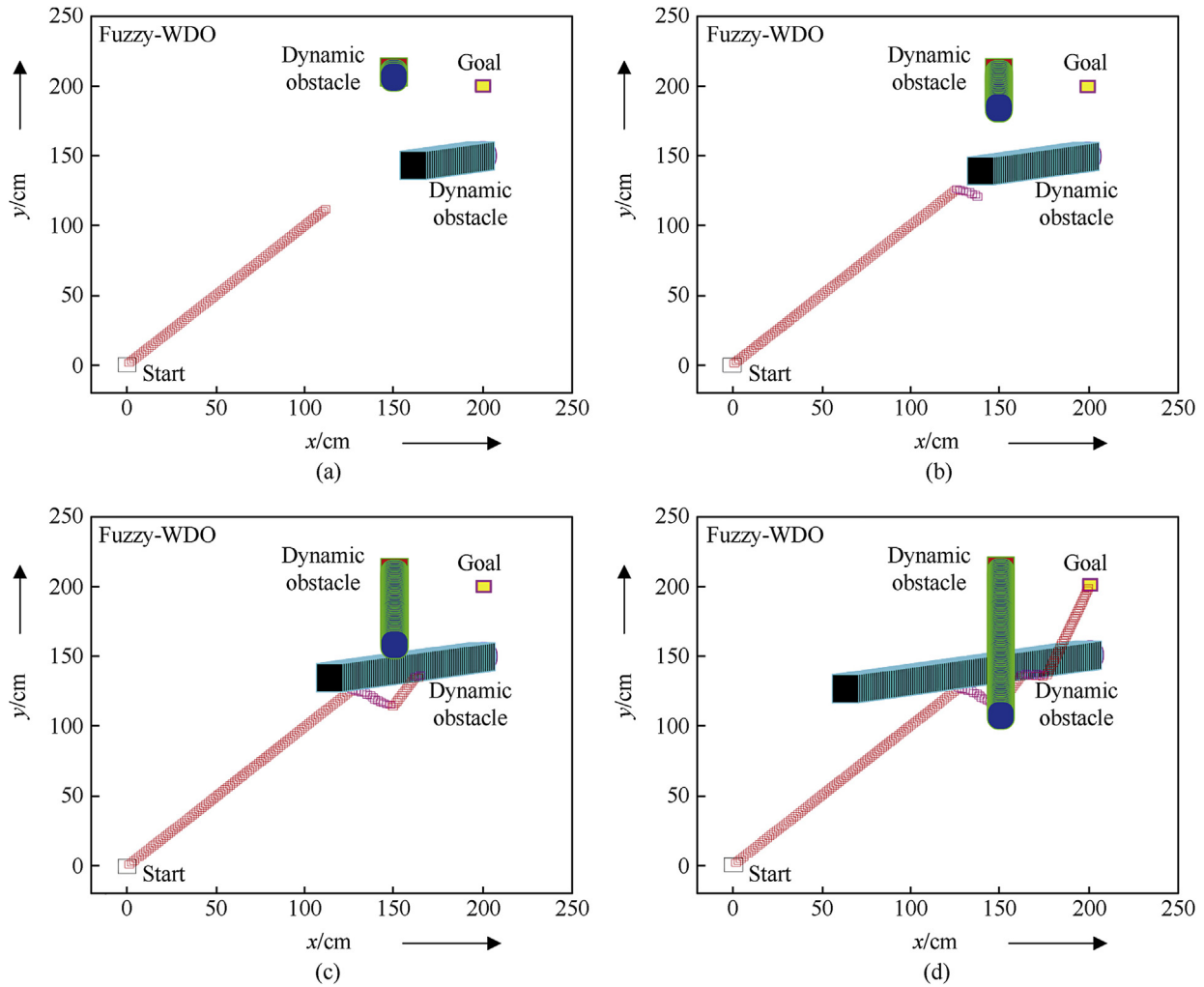
Fig. 10. Mobile robot navigation in the dynamic environment using Fuzzy-WDO controller.

Table 6
The simulation results of T1-SFLS and Fuzzy-WDO controllers.

Figure no.	Controller	Navigation path length /cm	Navigation time /s
Fig. 8	T1-SFLS	78.6	7.2
	Fuzzy-WDO	74.4	6.9
Fig. 9	T1-SFLS	103.7	9.1
	Fuzzy-WDO	98.2	8.7

Step 3. Assign random position and velocity of the air parcels.
Step 4. Evaluate the pressure (fitness) values of each air parcel at its current position.
Step 5. Once the pressure values have been evaluated, the population is ranked based on their pressure (ascending order), and the velocity updated according to equation (17) along with the restrictions are given in equation (19).
Step 6. Update the position of the air parcel for the next iteration according to equation (18) and also check the boundaries of the air parcel.
Step 7. Stop if a maximum number of iterations are achieved, else go to step 4.

When the maximum number of iterations is completed, the best pressure (fitness) value is achieved.

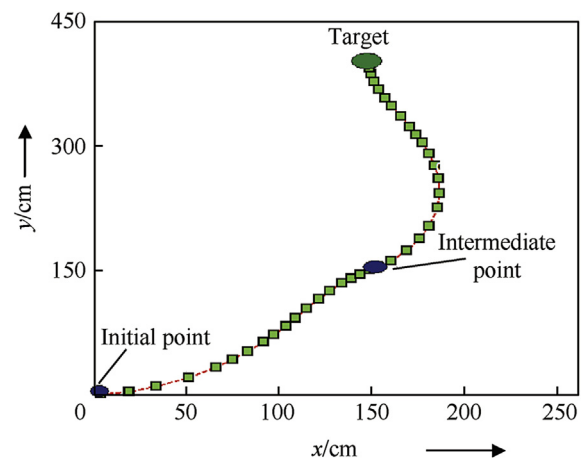This section describes the WDO algorithm used for the

Fig. 11. Mobile robot navigation in an environment without obstacle using fuzzy controller [23].

membership function parameter optimization of the T1-SFLS controller for the optimum navigation and collision avoidance in an unknown static and dynamic environment. One major problem with the fuzzy logic is the difficulty of constructing and tuning the
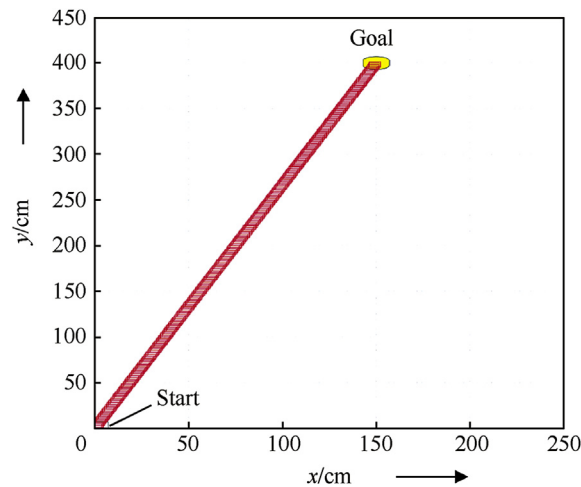
Fig. 12. Mobile robot navigation in an environment without obstacle using Fuzzy-WDO controller.
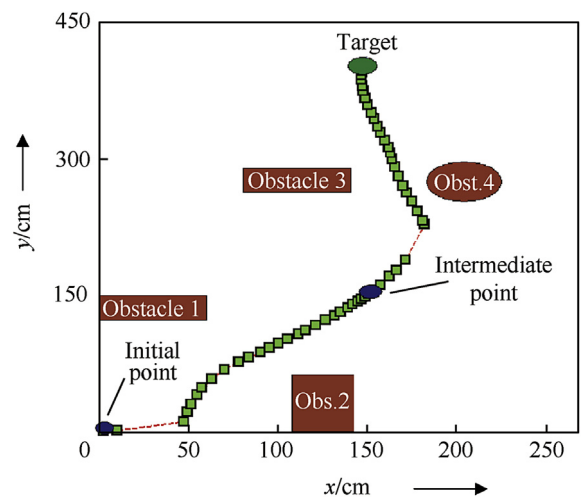

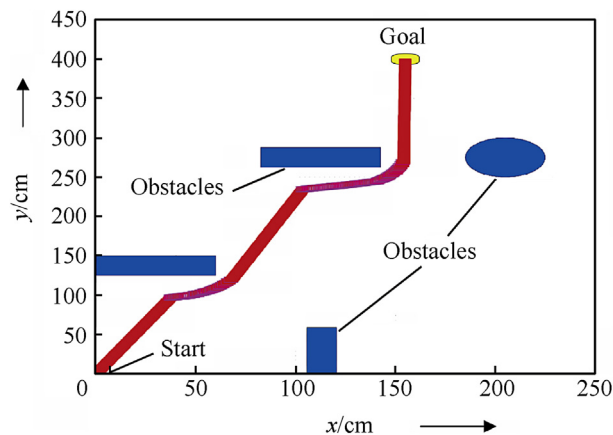Fig. 13. Mobile robot navigation in an environment with four obstacles using fuzzy controller [23].

Fig. 14. Mobile robot navigation in an environment with four obstacles using Fuzzy-WDO controller.

Table 7
The simulation result comparison between the fuzzy controller [23] and proposed Fuzzy-WDO controller.

Figure no.	Method	Navigation path length /cm
Fig. 11	Fuzzy controller [23]	181
Fig. 12	Fuzzy-WDO controller	165
Fig. 13	Fuzzy controller [23]	183
Fig. 14	Fuzzy-WDO controller	173

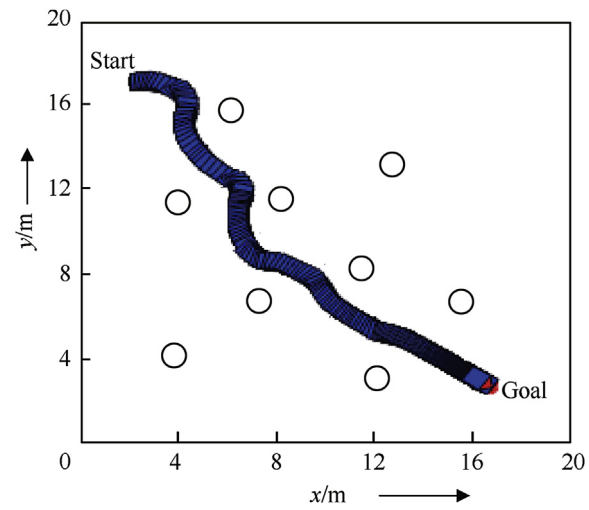Fig. 15. Mobile robot navigation between many obstacles using fuzzy model [24].
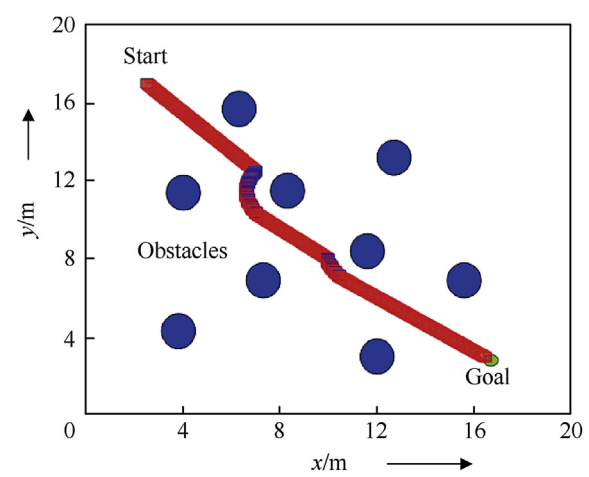

Fig. 16. Mobile robot navigation between many obstacles using Fuzzy-WDO controller.

Table 8
Comparison of simulation results between fuzzy model [24] over proposed Fuzzy-WDO controller.

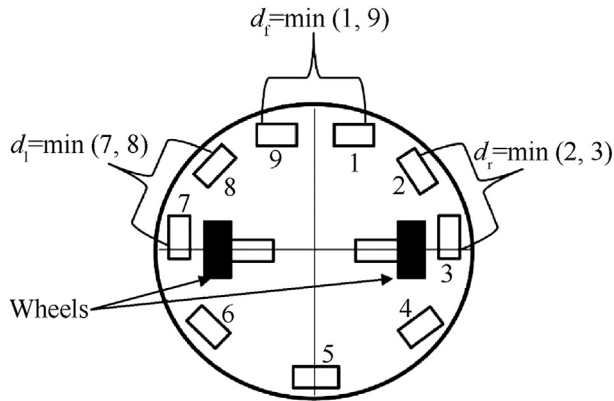| Figure no.	Method	Navigation path length /cm
Fig. 15	Fuzzy model [24]	91
Fig. 16	Fuzzy-WDO controller	84

Fig. 17. Infrared proximity sensor distribution of Khepera-III mobile robot.

correct membership function grade [22]. Because of this problem, the WDO algorithm is used to tune the adjusting parameters of the inputs and outputs. From Section 2, two Gbell membership function are considered for the inputs (d_f, d_l, and d_r) and outputs (m_r, and m_l). Each Gbell membership function has three adjusting parameters (a, b, and c). Therefore, each input has six adjusting parameters. Similarly, each output has six adjusting parameters. So the total number of adjusting parameters is to be thirty {5 (3 inputs + 2 outputs) × 2 (membership function) × 3 (adjusting parameters a, b, and c) = 30}.

The ranges of adjusting parameters are defined as [a_{min}, a_{max}] [b_{min}, b_{max}] and [c_{min}, c_{max}] respectively, for lower and the upper boundary of the WDO algorithm. The a_{min} and a_{max} are 30 and 65 for the membership function of the inputs. The b_{min} and b_{max} are 1 and 3.5 for the membership function of the inputs. The parameters c_{min} and c_{max} are 20 and 150 for the membership function of inputs respectively. Similarly, the a_{min} and a_{max} are 2 and 5 for the membership function of outputs. The b_{min} and b_{max} are 1 and 3.5 for the membership function of the outputs. The parameters c_{min} and c_{max} are located at 6.7 and 16.7 for the membership function of outputs respectively. Fig. 5 shows the air parcels representation of
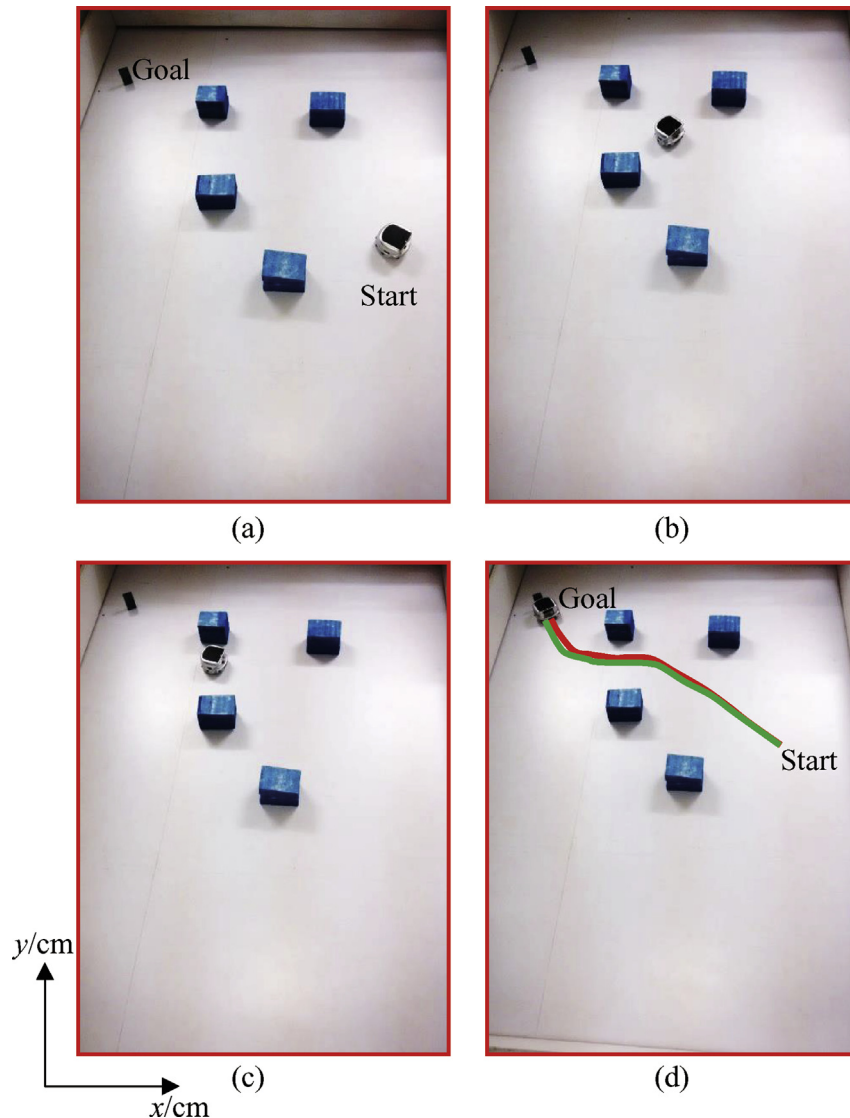

Fig. 18. Real-time navigation of Khepera-III mobile robot between the obstacles using T1-SFLS and Fuzzy-WDO controller.

the WDO algorithm. The optimized membership functions of the inputs (d_f, d_l, and d_r) and the outputs (m_r, and m_l) are shown in Figs. 6 and 7, respectively. The results of the adjusting parameters (a, b, and c) of the inputs and outputs after optimization are listed in Table 4 and Table 5, respectively.

The most important step in applying the WDO algorithm is to select the fitness function, which is used to evaluate the optimum pressure of the air parcels. In during the optimization process, the combined root mean square errors (CRMSE) are used to evaluate the fitness of the fuzzy controller

$$\text{RMSE}_{m_r} = \sqrt{\frac{1}{z}\sum_{p=1}^{z}\left(m_r^{\text{Actual}} - m_r^{\text{FW}}\right)^2} \tag{20}$$

$$\text{RMSE}_{m_l} = \sqrt{\frac{1}{z}\sum_{p=1}^{z}\left(m_l^{\text{Actual}} - m_l^{\text{FW}}\right)^2} \tag{21}$$

$$\text{CRMSE} = \text{RMSE}_{m_r} + \text{RMSE}_{m_l} \tag{22}$$

where m_r^{actual} and m_l^{actual} are the actual value of right and left motor velocity respectively. The m_r^{FW}, and m_l^{FW} are calculated value of the right and left motor velocity respectively through Fuzzy-WDO. The RMSE_{m_r}, and RMSE_{m_l} are the root mean square error of the right and left motor velocity respectively; and z is the iteration number.

4. Simulation results

This section describes the successful simulation results using T1-SFLS and Fuzzy-WDO controllers in the various unknown static and dynamic environments. The simulations are conducted using the MATLAB software on the HP 3.40 GHz processor. Figs. 8 and 9 show the navigation result of the mobile robot between the obstacles and walls respectively, using the T1-SFLS and Fuzzy-WDO controller in the unknown environments. Similarly, Fig. 10 demonstrates the navigation of a mobile robot in an unknown environment with the presence of two dynamic obstacles using Fuzzy-WDO controller. It is assumed that the position of the start point and goal point are known. But the positions of all the obstacles in the environment are unknown for the robot. In the simulation results, the green and red color trajectory indicates the path generated by the T1-SFLS and Fuzzy-WDO controllers respectively. Simulation results show the Fuzzy-WDO controller gives smooth and optimal path compared to the T1-SFLS controller. Table 6 shows the navigation path length and time taken by the robot using the T1-SFLS and Fuzzy-WDO controller in the various unknown environments.

5. Comparison with previous works

This section describes the computer simulation result comparison between the previous model [23] and proposed Fuzzy-WDO controller in the same path planning problems. In the article [23], the authors have used two simple fuzzy controllers such as tracking fuzzy logic control (TFLC) and obstacle avoidance fuzzy logic control (OAFLC) without adjusting its membership function for mobile robot navigation. Figs. 11 and 12 show the mobile robot navigation in the same environment without obstacle using fuzzy controller [23] and proposed Fuzzy-WDO controller, respectively. Similarly, Figs. 13 and 14 present the path covered by the robot in the same environment with the four obstacles using fuzzy controllers [23] and proposed Fuzzy-WDO controller, respectively. From the simulation figures, it can be seen that the proposed Fuzzy-WDO

controller covers shorter distance to reach the goal as compared to previous model [23] because WDO algorithm adjusts the membership function of the fuzzy controller, which provides better result compared to the standalone fuzzy model. Besides, the proposed Fuzzy-WDO controller also helps the mobile robot to reach the goal without taking any intermediate point. And due to this, it takes less time to reach the goal as compared to previous model [23]. Table 7 illustrates the path traced (in cm) by the robot to reach the goal using proposed controller and previous model [23].

In an article [24], the Cherroun and Boumehraz have designed the behaviour based Takagi-Sugeno type fuzzy model, which autonomously navigates the mobile robot in the crowded environment. Figs. 15 and 16 show the mobile robot navigation result comparison between the fuzzy model [24] and proposed Fuzzy-WDO controller, respectively. From the simulation results, it can be seen that the proposed controller provides the better trajectory in terms of path length and smoothness as compared to previous model. Table 8 shows the path covered (in cm) by the robot to reach the goal using proposed controller and previous model [24]. The centimetre measurements are taken on the proportional basis.

6. Experimental results

6.1. Khepera-III mobile robot description

The experiments are conducted using the Khepera-III mobile robot in unknown environments. The Khepera-III mobile robot has two wheels controlled by two DC servo motors and one caster wheel. The diameter and height of the robot are 13 cm and 7 cm respectively. The Khepera-III mobile robot is equipped with nine infrared proximity sensors and five ultrasonic sensors, as shown in Fig. 17. The Infrared proximity sensor reads obstacles up to 30 cm, and the ultrasonic sensor reads obstacles from 20 cm to 4 m approximately. In this study, we have set the minimum and maximum velocity of Khepera-III mobile robot between the 6.7–16.7 cm/s.

6.2. Experiments

In the experiments, the controllers are implemented in the Khepera-III mobile robot using HP laptop. The width and height of the experimental platform are 250 cm and 250 cm, respectively. Fig. 18 and Fig. 19 shows the real-time navigation of the Khepera-III mobile robot in unknown environments with the obstacles and walls, respectively. In Fig. 18, the start position of the robot is (175, 100) cm, and the position of the goal is (0, 200) cm. The starting angle between the robot and the goal is 29.74°. Similarly, in Fig. 19, the start position of the robot is (50, 50) cm, and the goal position is (250, 200) cm. The starting angle between the robot and the goal is 36.87°. In the experiments, it is assumed that the position of the start point and goal point are known, but the positions of all the obstacles in the environment are unknown for the robot. The T1-SFLS and Fuzzy-WDO controller generate the motor velocity control command for obstacle avoidance using on-board sensor information. The successful experimental results in the various unknown environments are shown below to verify the effectiveness of the T1-SFLS and Fuzzy-WDO controllers. Table 9 shows the experimental path length and time taken by the Khepera-III mobile robot to reach target using the T1-SFLS and Fuzzy-WDO controllers in the various unknown environments. Tables 10 and 11 present the traveling path length and navigation time comparison between the simulation and experimental results. In the comparison study between the simulation and experiments, it is observed that some errors have been found, these happen due to slippage and friction during real time experiment.
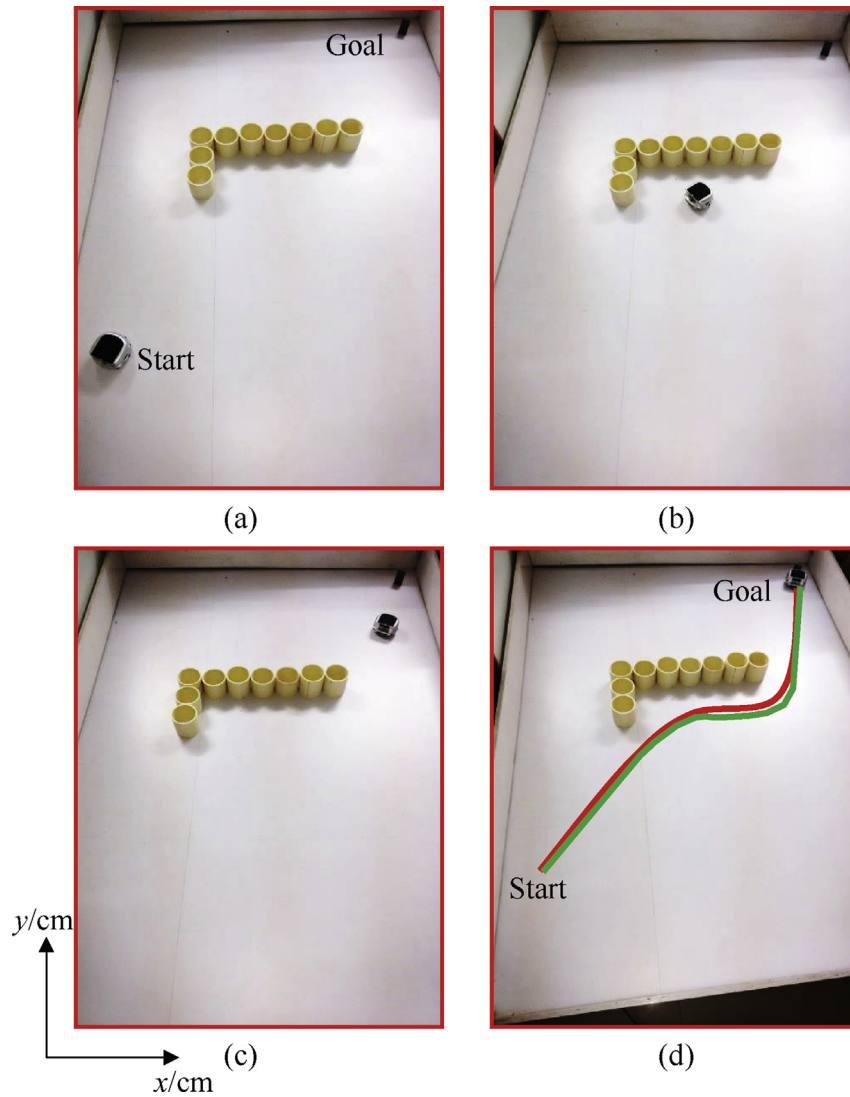
Fig. 19. Real-time navigation of Khepera-III mobile robot between the walls using T1-SFLS and Fuzzy-WDO controller.

7. Conclusion

In this article, the two methods T1-SFLS controller and the hybrid Fuzzy-WDO algorithm have been applied to the mobile robot navigation. A new population-based optimization algorithm, called Wind Driven Optimization (WDO) is used to optimize and set the antecedent and consequent parameters of the fuzzy controller. The proposed algorithms are successfully verified through simulations and real-time experiments in the different environments. Simulation and experimental results demonstrate the Fuzzy-WDO controller provide better performance as compared to the T1-SFLS controller.

Table 9
The experimental results of T1-SFLS and Fuzzy-WDO controllers.

Figure no.	Controller	Experimental path length /cm	Experimental time /s
Fig. 18	T1-SFLS	83.9	7.7
	Fuzzy-WDO	78.2	7.3
Fig. 19	T1-SFLS	111.3	9.7
	Fuzzy-WDO	103.4	9.2

Table 10
Traveling path lengths comparison between simulation and experimental results.

Figure no. (Simulation and experimental res.)	Controller	Traveling path length /cm		Error between simulation and experimental results %
		Simulation result	Experimental result	
Figs. 8 and 18	T1-SFLS	78.6	83.9	6.3
	Fuzzy-WDO	74.4	78.2	4.86
Figs. 9 and 19	T1-SFLS	103.7	111.3	6.82
	Fuzzy-WDO	98.2	103.4	5.1

Table 11
Navigation time comparison between simulation and experimental results.

Figure no. (Simulation and experimental res.)	Controller	Navigation time /s		Error between simulation and experimental results /%
		Simulation result	Experimental result	
Figs. 8 and 18	T1-SFLS	7.2	7.7	6.49
	Fuzzy-WDO	6.9	7.3	5.47
Figs. 9 and 19	T1-SFLS	9.1	9.7	6.18
	Fuzzy-WDO	8.7	9.2	5.43

References

[1] Abadi DNM, Khooban MH. Design of optimal mamdani-type fuzzy controller for nonholonomic wheeled mobile robots. J King Saud University-Engineering Sci 2015;27(1):92–100.
[2] Algabri M, Mathkour H, Ramdane H, Alsulaiman M. Comparative study of soft computing techniques for mobile robot navigation in an unknown environment. Comput Hum Behav 2015;50:42–56.
[3] Hui NB, Pratihar DK. A comparative study on some navigation schemes of a real robot tackling moving obstacles. Robotics Computer-Integrated Manuf 2009;25(4):810–28.
[4] Pothal JK, Parhi DR. Navigation of multiple mobile robots in a highly clutter terrains using adaptive neuro-fuzzy inference system. Robotics Aut Syst 2015;72:48–58.
[5] Montiel O, Orozco-Rosas U, Sepulveda R. Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles. Expert Syst Appl 2015;42(12):5177–91.
[6] Hossain MA, Ferdousand I. Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique. Robotics Aut Syst 2015;64:137–41.
[7] Baturone I, Gersnoviez A, Barriga A. Neuro-Fuzzy techniques to optimize an FPGA embedded controller for robot navigation. Appl Soft Comput 2014;21:95–106.
[8] Ming L, Zailin G, Shuzi Y. Mobile robot fuzzy control optimization using genetic algorithm. Artif Intell Eng 1996;10(4):293–8.
[9] Liang Y, Xu L, Wei R, Hu H. Adaptive fuzzy control for trajectory tracking of mobile robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010, pp. 4755–4760.
[10] Purian FK, Sadeghian E. Mobile robots path planning using ant colony optimization and fuzzy logic algorithms in unknown dynamic environments. In: IEEE International Conference on Control, Automation, Robotics and Embedded Systems (CARE), 2013, pp. 1–6.
[11] Castillo O, Martinez-Marroquin R, Melin P, Valdez F, Soria J. Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. Inf Sci 2012;192:19–38.
[12] Chung HY, Hou CC, Liu SC. Automatic navigation of a wheeled mobile robot using particle swarm optimization and fuzzy control. In: IEEE International Symposium on Industrial Electronics (ISIE), 2013, pp. 1–6.
[13] Allawi ZT, Abdalla TY. A PSO-optimized type-2 fuzzy logic controller for navigation of multiple mobile robots. In: IEEE International Conference on Methods and Models in Automation and Robotics (MMAR), 2014, pp. 33–39.
[14] Yanar TA, Akyurek Z. Fuzzy model tuning using simulated annealing. Expert Syst Appl 2011;38(7):8159–69.
[15] Martinez-Alfaro H, Gomez-Garcia S. Mobile robot path planning and tracking using simulated annealing and fuzzy logic control. Expert Syst Appl 1998;15(3):421–9.
[16] Bayraktar Z, Komurcu M, Bossard JA, Werner DH. The wind driven optimization technique and its application in electromagnetics. IEEE Trans Antennas Propag 2013;61(5):2745–57.
[17] Kuldeep B, Singh VK, Kumar A, Singh GK. Design of two-channel filter bank using nature inspired optimization based fractional derivative constraints. ISA Trans 2015;54:101–16.
[18] Bhandari AK, Singh VK, Kumar A, Singh GK. Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. Expert Syst Appl 2014;41(7):3538–60.
[19] Bayraktar Z, Turpin JP, Werner DH. Nature-inspired optimization of high-impedance metasurfaces with ultrasmall interwoven unit cells. IEEE Antennas Wirel Propag Lett 2011;10:1563–6.
[20] Mohanty PK, Parhi DR. A new hybrid optimization algorithm for multiple mobile robots navigation based on the CS-ANFIS approach. Memetic Comput 2015;7(4):255–73.
[21] Wong C, Wang H, Li S. PSO-based motion fuzzy controller design for mobile robots. Int J fuzzy Syst 2008;10(1):284–92.
[22] Tahmasebi P, Hezarkhani A. A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation. Comput Geosci 2014;42:18–27.
[23] Faisal M, Hedjar R, Al Sulaiman M, Al-Mutib K. Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment. Int J Adv Robotic Syst 2013;10(37):1–7.
[24] Cherroun L, Boumehraz M. Fuzzy behavior based navigation approach for mobile robot in unknown environment. J Electr Eng 2013;13(4):1–8.