



Optimal Setting of Membership Functions for Interval Type-2 Fuzzy Tracking Controllers Using a Shark Smell Metaheuristic Algorithm

Felizardo Cuevas¹ · Oscar Castillo¹  · Prometeo Cortes¹

Received: 1 February 2021 / Revised: 18 April 2021 / Accepted: 7 June 2021 / Published online: 10 July 2021
© Taiwan Fuzzy Systems Association 2021

Abstract This article describes the application of a variant of the shark smell optimization (VSSO) biological inspired algorithm in the optimal design of a type-2 fuzzy controller. We show how the performance of VSSO is based on the frontal and rotational movement of the shark when navigating a dimensional search domain, which is based on the food-seeking behavior of sharks. The optimization of the design of a Mamdani interval type-2 fuzzy controller (IT2-FLC) applying VSSO is also described. The optimized controller is tested with the navigation of an autonomous mobile robot (AMR) in an unknown and changing environment. This work was developed as follows: first, the VSSO algorithm is improved by adjusting its main alpha and beta parameters with a fuzzy system, later the parameter values of the fuzzy controller input/output membership functions are optimized. Finally, a comparison is made between the results of type-1 (T1) and interval type-2 fuzzy controllers applying the proposed methodology. When comparing the T1 and IT2-FLC controllers, the application of the VSSO algorithm in T1-FLC shows good performance in robot navigation; however, IT2-FLC presents better performance due to its ability to handle higher levels of uncertainty. The performance evaluation of the proposed method and its application in different navigation problems has been carried out through computer simulations using Matlab-Simulink.

Keywords Autonomous mobile robots · Type-2 fuzzy systems · Type-2 fuzzy logic controller · Type-1 fuzzy logic controller · Variant shark smell optimization

1 Introduction

Nowadays, optimization theory has been receiving increasing attention, and its applications in recent decades have achieved a greater impact in real-world applications. Flexible computing techniques, such as fuzzy logic [1, 2], neuro-fuzzy systems [3], and nature inspired algorithms (Algorithm of bee colony optimization [4, 5], particle swarm optimization [6], ant colony algorithm [7], shark smell optimization algorithm [8], air-driven optimization [9], galactic swarm algorithm [10], differential evolution [11] and more), are some examples, where optimization techniques have been used in mobile robot navigation. In addition, it has been proven that techniques inspired by nature provide very good results, but it has also been seen that they are not the best solutions for all problems. The No-Free-Lunch (NFL) optimization theorem establishes that there is no algorithm of universal learning. On the other hand, fuzzy logic has been shown to be one of the most cited and used methods in the field of mobile robotics and control. Undoubtedly, the current use of metaheuristic algorithms to solve optimization problems is due to their ability to find the solutions closest to the optimum, reducing computational time and costs [12, 13]. These algorithms are currently being widely used in various applications, demonstrating that they have become a new paradigm of collective intelligence and a topic that has guided the work by many researchers in different fields, such as in fuzzy control and linguistic synthesis [14–16]. In nature, when looking closely at real phenomena, some

✉ Oscar Castillo
ocastillo@tectijuana.mx

Felizardo Cuevas
felizardo.cuevas@tectijuana.edu.mx

Prometeo Cortes
Prometeo.cortes@tectijuana.edu.mx

¹ Tijuana Institute of Technology, Tijuana, BC, Mexico

systems can be found to be of self-organizing [7] nature. This characteristic of systems in nature has been applied to many algorithm-based search and optimization methods in solving many problems. Here, we describe some examples of related metaheuristics that have been used for the optimization of fuzzy controllers.

In [6], an improved particle swarm optimization (PSO) algorithm with dynamic adaptation of parameters is presented. An improvement to diversity and convergence in the particle swarm algorithm was proposed by applying type-1 fuzzy logic. In this case, fuzzy rules are used to adjust the main parameters in PSO, thus achieving a better dynamic adaptation of the algorithm parameters.

In [10], an improvement to the galactic swarm (GSO) algorithm is proposed. At least three different designs of fuzzy systems for the dynamic adjustment in the main parameters in GSO are presented. This optimized fuzzy logic GSO method increases its efficiency as a metaheuristic by having multiple exploration and exploitation cycles, thus increasing the possibilities of improving its precision in finding the global minimum.

In [13–15], different improvements to the differential evolution (DE) algorithm, with a fuzzy approach to dynamically adapt the main mutation parameter (F), are presented. This new algorithm proposal was called the fuzzy differential evolution (FDE) algorithm. For a set of benchmark functions, a performance comparison between type-1 and interval type-2 fuzzy logic was carried out by applying the proposed algorithm. In addition, a path optimizer was designed by dynamically adjusting the fuzzy controller membership functions in a motion controller [17, 18].

In [19], an improvement to the BCO algorithm (bee colony optimization) was proposed by applying a fuzzy approach in the dynamic adjustment of the parameters in BCO. This work initially shows the traditional BCO in the optimization of type-1 fuzzy controllers. Second, a change is applied to the original method by making use of fuzzy logic in dynamically adjusting the values of the main parameters, during execution. Finally, the proposed modification to the method is used in the optimization of control problems. In addition, in [20] BCO was used to search for the optimal distribution of parameter values in the membership functions of the fuzzy controllers, seeking stability in the autonomous mobile robot (AMR) trajectory. In the model for the type-2 fuzzy system, two types of perturbation were used in the search for improving its behavior in the face of uncertainty, showing better results against its counterpart, the original BCO.

In [21], the artificial bee colony (ABC) algorithm was used for automatically tuning a fuzzy load frequency controller (FLFC), and it was designed to smooth out deviations in frequency and power, due to load

disturbances of the interconnected electrical system. The optimal adjustment in the membership functions (MFs) and the fuzzy control rules are taken into consideration, and this was of great importance to improve the performance of the method and achieve a level of robustness in the operation of the system.

There are also previous works that use the SSO algorithm [8, 22]. For example, the work oriented to Solar energy forecasting [23] or the optimal switching sequence of the reconfiguration of the electrical distribution network [24]. However, this work makes a difference by using a fuzzy variant that adjusts the main alpha and beta parameters of VSSO and is applied for the first time to optimize fuzzy logic controllers. The objective of this research is the dynamic search for better parameter values of the membership functions of the interval type-2 fuzzy logic controller (IT2FLC), having as objective function the minimization of errors (MSE) and the execution time in the process of following a trajectory. So, the main contribution of the paper is the proposal of the fuzzy variant of SSO and its application for the optimization of type-2 fuzzy control of AMR.

The following parts of this document are organized as follows: Sect. 2 presents an introduction and description of the used methodology, Sect. 3 describes the problems to be solved using the proposed methodology, Sect. 4 explains and shows the obtained experimental results, and finally, we close in Sect. 5 showing some future work suggestions and conclusions.

2 Overview of the Proposed Methodology, Type-1 and Type-2 Fuzzy Logic Systems

This section presents the methodology used in the development of our proposal, where their characteristics and behavior are detailed, and this is because they are relatively new and innovative. The particular SSO metaheuristic was selected from a set of extensive and recently used optimization algorithms that have been shown to give excellent results in solving optimization problems. For this reason, this work aims to show its potential and performance in problem optimization.

2.1 The Fuzzy Variant of Shark Smell Optimization (VSSO)

The original shark smell optimization (SSO) is a nature-based metaheuristic proposed by Oveis Abedinia et al. [8], based on the observation of how sharks' olfactory power is capable of detecting different types of smells from all directions. This metaheuristic uses a method that emulates the technique used by sharks when looking for prey

(moves, moves, and turns) and methods, such as using the blood concentration as a key factor in finding optimal routes to the food sources [22]. Recently, bio-inspired systems have caused a great boom, becoming a source of inspiration for the development of new ideas that currently model many artificial intelligence systems. This SSO metaheuristic presents several assumptions, described below in this section. The mathematical model is described based on the three main components that are shown below:

- (1) Initialization of food source (Population of solutions): The injured fish leaves a scattered trail of blood in the sea (search domain). It is considered at this point that the food (prey) is approximately fixed.
- (2) Shark search (exploitation phase, determination of food location): When blood is dispersed in the water, the flow of water distorts the scent particles of the prey. Therefore, the more intense the odor particles are, the closer the shark is to the prey. Following the effect of the shark’s scent intensity, the shark can reach the prey.
- (3) Scout sharks (exploitation Phase): this is a process of evaluating the quality of the solution in the search for a food source, in the shark search domain.

This metaheuristic is a highly efficient optimization algorithm, which can be extremely efficient in finding solutions based on the shark’s prey scent-picking abilities. Figure 1 illustrates the dynamics of shark movement in search of the most highly concentrated smell source. This new methodology is implemented during the iterations of the proposed algorithm (VSSO) in the search for the best solution for an optimization problem.

The detailed steps of the proposed algorithm (in a minimization problem) are briefly explained below:

2.1.1 Shark Position Initialization

First, we have to perform the shark position initialization using Eq. (1), which randomly generates initial solutions of a population for an optimization problem in the search domain (the sea, in this case). The food source (prey) is an optimal solution, so the set of solutions is characterized by

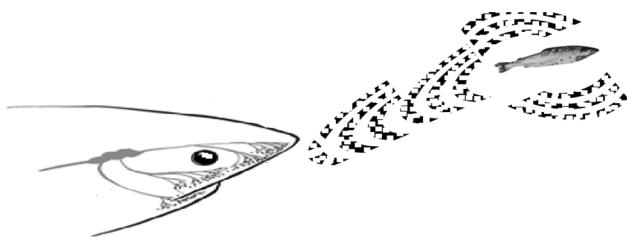


Fig. 1 Shark following the effect of the shark’s scent intensity, the shark reaches the prey

a source with an smell concentration and with a position. An initial solution is formulated in the following general way according to Eq. (1) [22]:

$$[X_1^1, X_2^1, X_3^1, \dots, X_{NP}^1] \quad NP = \text{Population size} \quad (1)$$

where superscript 1 represents the number of execution or iteration, there i_{th} initial point vector X_i^k , i.e., i_{th} possible initial solutions for the problem to be optimized is as follows Eq. (2):

$$X_{ij}^1 = [X_{i,1}^1, X_{i,2}^1, X_{i,3}^1, \dots, X_{i,NV}^1] \quad (2)$$

where X_{ij}^k belonging to J th dimension in the i th shark position; NV represents the number of decision variables in the optimization problem. A higher concentration on the current position of the shark will be an indication of how close the shark is to the prey (solution).

2.1.2 Forward Movement

Hypothetically, while there is dispersal in the water of the prey’s blood, the position and speed of the shark evolve towards the target, resulting in better solutions. So synchronously, the position vector and the velocity vector contain dimensional elements and decision variables represented by Eq. (3):

$$V_i^1 = [v_{i,1}^1, v_{i,2}^1, v_{i,3}^1, \dots, v_{i,NV}^1] \quad (3)$$

Limits must be established that should not be exceeded, when the intensity and concentration of the odor exponentially increase the speed of the shark. The dynamic inertia, when the shark swims, is given by the speed formula of each dimension and is given as follows in Eq. (4):

$$|v_{ij}^k| = \min \left[|\eta_k \cdot R_1 \cdot \nabla(OF)|_{x_{ij}^k} + |\alpha_k \cdot R_2 \cdot v_{ij}^{k-1}|, |\beta_k \cdot v_{ij}^{k-1}| \right] \\ i = 1, 2, 3, \dots, NP \quad j = 1, 2, 3, \dots, NV \quad k = 1, 2, 3, \dots, k_{max} \quad (4)$$

where in the original SSO: η_k : is a parameter in the interval [0 1], which is a gradient coefficient; $\nabla(OF)$: is the gradient of the objective function; R_1 and R_2 are random values with uniform distribution. The sign considered for the value of v_{ij}^k is based on the direction taken by the term chosen in the minimal operator; β_k : is a speed limiter factor for each stage k, it’s a random number in [0, 1] and α_k : is a measure of inertia and represents the weight coefficient in the shark’s swim, it is also a random number in [0, 1].

In our proposal (VSSO), two main parameters are adjusted with a fuzzy approach, taking the iteration as input with a range [0 1], and having as outputs: Alpha with a range [0 1] and finally Beta in a range [0 4]. In this form,

the coefficient of inertia or moment range ∞ and the speed limiting factor β , are dynamically adjusted in each stage k (number of stages or iterations involved in the frontal movement of the shark). For the calculation of the iteration value, we use Eq. (5) [7].

$$\text{Iteration} = \frac{\text{Current iteration}}{\text{Maximum of iterations}} \tag{5}$$

where the iteration variable performs the calculation of the percentage of elapsed iterations, the ‘‘current iteration’’ is described as the actual iteration and the ‘‘maximum of iterations’’ as the number of iterations established in the algorithm, this illustrated in Fig. 2a. Equation (5) is used as the input for the fuzzy system (called FSSO) for the adaptation of parameters, using this equation where the range of the input is from 0 to 1. The input variable is shown in Fig. 2b, where this variable has three triangular membership functions (Low, Medium, High), and their values are calculated using Eq. (5). The first output variable that is the α parameter is granulated into three triangular membership functions (Low, Medium, High) is shown in Fig. 3a, with a range from 0 to 1. This range was found by trial and error, in the search for the desired behavior of the SSO established by its original author. The second output variable β is shown in Fig. 3b with the range from 0 to 4, granulated into three membership functions (low, medium, high), which correspond to a factor to limit the speed of the shark.

The current velocity (velocity vector) considered as the previous position will give us a new position Y_i^{k+1} during the shark’s forward movement shown by Eq. (6):

$$Y_i^{k+1} = x_i^k + v_i^{k-1} \cdot \Delta t_k \tag{6}$$

where: Δt_k is time interval assumed to be 1 for every k_{th} iteration; Y_i^{k+1} is the new position; x_i^k is the actual position; and v_i^{k-1} is the velocity in the previous position.

2.1.3 Rotational Movement

The rotational displacement gives a shark the ability to find the highest odor particles as it moves forward, allowing it

to perform a local search at each iteration to find higher quality candidate solutions (positions). As shown in Fig. 4, the shark’s rotation is in a closed pattern, almost never in a circle. This is modeled by Eq. (7) below:

$$Z_i^{k+1,m} = Y_i^{k+1} + R_3 \cdot Y_i^{k+1}; \quad m = 1, 2, 3, \dots, M \tag{7}$$

where R_3 is a correlation value in the interval $[-1, 1]$. The previous procedure looks for the answer Y_i^{k+1} to complete the local search, and the best location $Z_i^{k+1,m}$ is determined. The maximum number of iterations to be performed by the VSSO algorithm is determined by M , and $m = (1, 2, 3, \dots, M)$ is the iteration (ubication points) searched at each stage around Y_i^{k+1} .

2.1.4 The Location Search of the Shark Position

When the shark encounters, during the rotational movement, a position with more intense smell, this will be the basis of its search from this position and continue its movement on this path, as shown in Fig. 4. Our goal during its execution is the minimization of the error, and we look for the smallest objective function value that is determined in the formula shown in Eq. (8):

$$X_i^{k+1} = \text{argmin}\{\text{OF}(Y_i^{k+1}), \text{OF}(Z_i^{k+1,1}), \dots, \text{OF}(Z_i^{k+1,M})\} \tag{8}$$

where X_i^{k+1} is the next shark position; Y_i^{k+1} is the forward movement and $Z_i^{k+1,M}$ is the shark’s rotational movement, similar to a local search of the SSO algorithm.

The goal is the minimization of the objective function OF. The iterations will stop until k reaches its minimum value (the best individual) or a stopping criterion is met in a particular search space, determined for the particular problem to be optimized.

The Variant Shark Smell Optimization Algorithm, like other methods based on a population, contains parameters in a similar way to other metaheuristic optimization methods, which include the parameters of number of stages k_{max} as well as α , β , and η of each stage. The following values were determined for VSSO: $\eta = 0.7$, $\alpha = \text{dynamic}$,

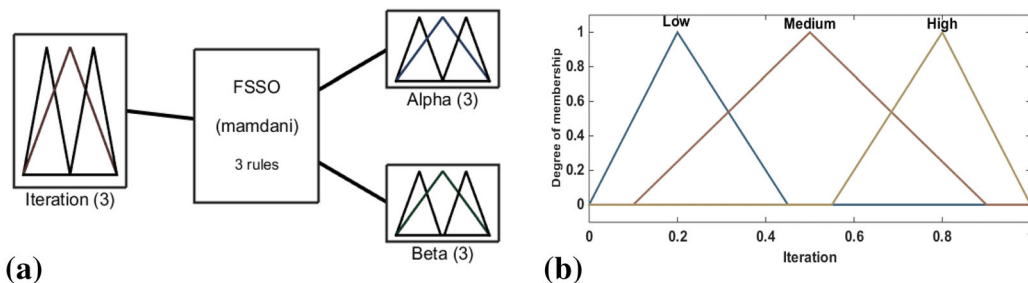


Fig. 2 a FSSO proposed fuzzy system and b the input iteration

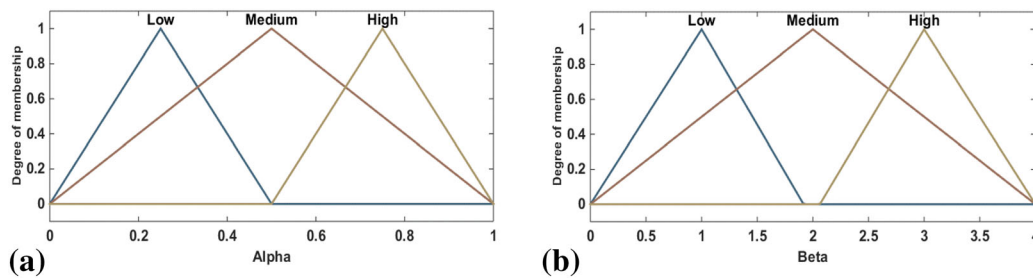


Fig. 3 FSSO output variables: a Alpha and b Beta

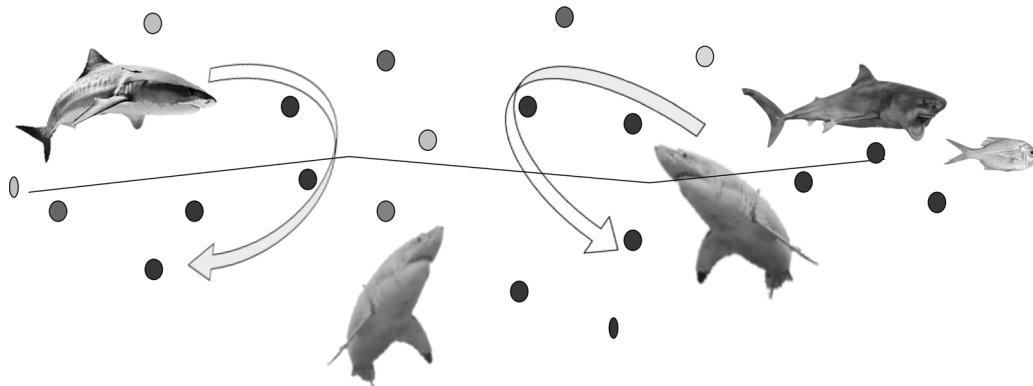


Fig. 4 Illustration of Shark's rotational movement

and $\beta = \text{dynamic}$, these parameters are adjusted for an optimization problem individually, and this is part of the proposal of our work using fuzzy logic to achieve this adaptation.

2.2 Wind-Driven Optimization (WDO) Background

Developed in [25], WDO is a new metaheuristic inspired by atmospheric wind circulation, today it is widely used to solve multidimensional problems [26]. A characteristic of this metaheuristic algorithm is how it represents its population by employing air packets, where the position and speed of their displacement are updated in each iteration.

$$u_{t+1}^i = (1 - \alpha)u_t^i - g x_t^i \left(RT \left| \frac{1}{r} - 1 \right| (x_{\text{opt}} - x_t^i) \right) + \left[\frac{cu_{t+1}^{\text{otherdim}}}{r} \right] \tag{9}$$

The new speed, from Eq. (9), describes the terms of the algorithm procedure, a maximum atmospheric pressure, α is the coefficient of friction, and u^i is the speed of the current air package. The second term is made up of g , which is equivalent to gravity, while x^i is the current position, RT is a universal constant that denotes the gas and temperature separately, r is the range of the air package with regards to their pressure, here the air particles are classified in descending order, x_{opt} denotes the best global position, and the Coriolis constant is represented by c .

Equation (10) represents the next position in the WDO.

$$x_{t+1}^i = x_t^i + u_{t+1}^i \tag{10}$$

2.3 Type-1 and Type-2 Fuzzy Logic Background

Zadeh originally proposed fuzzy logic in [27], and in [28] also describes recent applications of the method. Fuzzy logic unifies the incompatibility between precision and complexity because normal computational procedures are so precise that there is an intrinsic incompatibility with the complexity of human reasoning. Nowadays, fuzzy logic is a tool that makes life more comfortable, and human reasoning in its different ways of thinking can be modeled through the use of linguistic variables, with the result that when implemented on a machine, the logic of a human expert can be used for automating human reasoning in decision making.

Type-2 Fuzzy logic [29] was presented as an extension of classical fuzzy logic by assigning fuzzy values to memberships of linguistic variables and improves on the inference in type-1 fuzzy sets, thanks to the management of the existing uncertainties. In Figs. 3 and 4, Eqs. (11) and (12), the trapezoidal type-1 and type-2 membership functions are illustrated, as follows. As can be appreciated, when comparing the figures, a type-2 membership function (Fig. 6) uses an area to represent uncertainty (called footprint of uncertainty (FOU)) instead of having a simple curve like in a type-1 membership function (Figs. 5, 6).

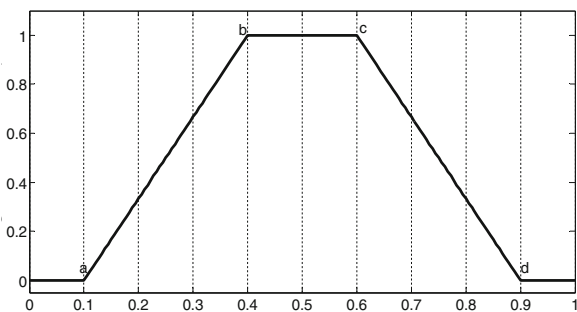


Fig. 5 Description of type-1 trapezoidal membership function

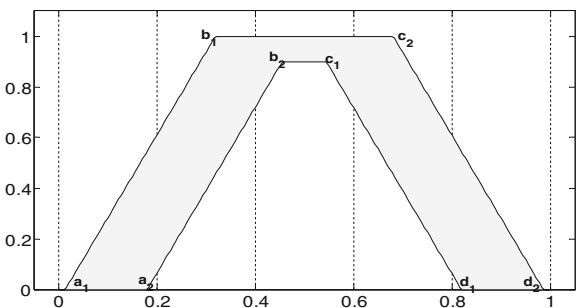


Fig. 6 Description of type-2 trapezoidal membership function IT2

$$\mu_{a,b,c,d}(x) = \begin{cases} 0 & \text{si, } x \leq a \\ \frac{x-a}{b-a} & \text{si, } a \leq x \leq b \\ 1 & \text{si, } b \leq x \leq c \\ \frac{d-x}{d-c} & \text{si } c \leq x \leq d \\ 0, & \text{si } d \leq x \end{cases} \quad \mu(x)$$

$$= \text{trapmf}(x, [abcd]) \tag{11}$$

$$\mu_{a,b,c,d}(x) = \begin{cases} 0 & \text{si, } x \leq a \\ \frac{x-a}{b-a} & \text{si, } a \leq x \leq b \\ 1 & \text{si, } b \leq x \leq c \\ \frac{d-x}{d-c} & \text{si } c \leq x \leq d \\ 0, & \text{si } d \leq x \end{cases} \quad \mu(x) = \text{itrapaty}(x, [a_1b_1c_1d_1a_2b_2c_2d_2])$$

$$\tilde{\mu}(x) = [\underline{\mu}(x), \bar{\mu}(x)] = \text{itrapaty}(x, [a_1b_1c_1d_1a_2b_2c_2d_2])$$

$$\mu_1(x) = \max\left(\min\left(\frac{x-a_1}{b_1-a_1}, 1, \frac{d_1-x}{d_1-c_1}\right), 0\right)$$

$$\mu_2(x) = \max\left(\min\left(\frac{x-a_2}{b_2-a_2}, 1, \frac{d_2-x}{d_2-c_2}\right), 0\right) \tag{12}$$

The combination of fuzzy logic with various AI techniques makes it possible to achieve good results in areas that previously seemed to be very difficult, like navigation of autonomous mobile robots. In the industry, robotics also plays a crucial role in the manufacturing processes. The updating and optimization of the software and hardware through these methodologies is of utmost importance since the efficiency and performance in the robot’s working environment will be improved [30].

We propose the design of a fuzzy controller that, when implemented, offers outputs with a soft response instead of a traditional hard logic response with oscillations. Recent work has been developed using other combinations based on metaheuristics and fuzzy logic in robotics, such as those described in [31], and [32]. In the last decade, other metaheuristics and combinations between them have been carried out, and they have also been hybridized with fuzzy logic in robotic applications, as in [33]. In addition, in [34, 35], the development of applications with other similar methods has been presented. In the same way, an optimization of the fuzzy logic controller with the VSSO metaheuristic is proposed in this work. In this case, it is a Mamdani type-2 fuzzy controller and its main function is to create a smooth response without oscillations on a mobile robotic platform, outperforming what can be done with type-1 fuzzy and traditional PID controllers [36].

3 Application of the Proposed Algorithm for Optimizing the Performance of the Mobile Robot

The proposed methodology for the optimization of parameter values in the fuzzy controller membership functions with the Variant Shark Smell Optimization Algorithm is the following: Evaluate the fuzzy controller performance using a differential robot plant and generate evaluation results concerning the desired path, these values will be taken by the VSSO algorithm, which will produce as a result the distance between the parameters of the membership functions, to then continue with the loop process. Next, we propose a fuzzy control architecture to achieve the dynamic variation of the parameters of the plant in the execution time, as shown in Fig. 7.

3.1 Type-1 Fuzzy Controller

A fuzzy controller is optimized for an AMR of differential type, using the proposed VSSO algorithm applied to this specific problem. The final design of the fuzzy controller is described in the following sections, and Fig. 8 describes the final methodology that is applied, and we describe how the VSSO dynamically searches for the optimal parameters of the membership functions in the fuzzy input and output sets. The plant for the differential mobile robot is described in [37]. The main objective of finding a better controller is to optimally follow a certain path, where the relationship between the linguistic variables is presented through an expert design of nine fuzzy rules. Figure 9 presents the desired path that will be the reference to be followed by the robot. Figure 9 illustrates the X-axis that shows the displacement in the X coordinate and the Y-axis represents the

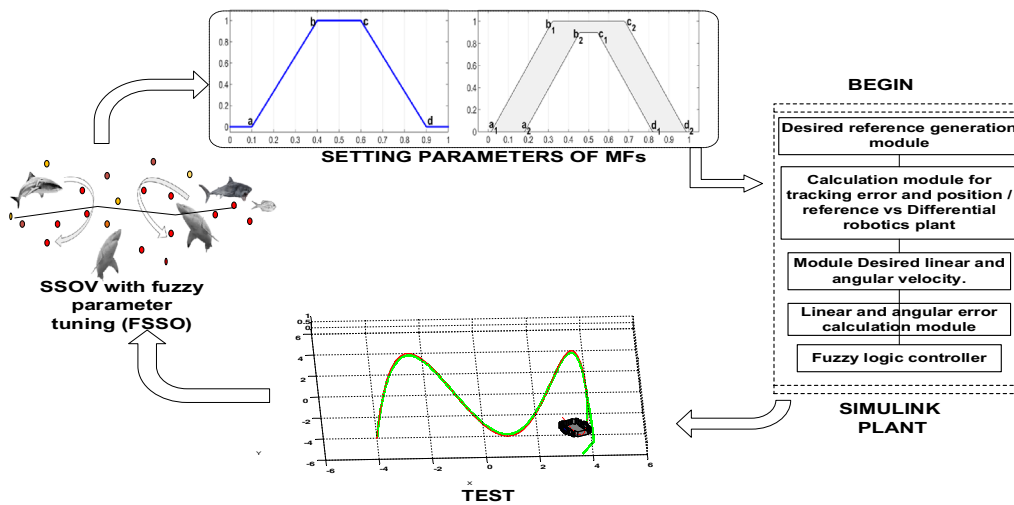


Fig. 7 Procedure for searching for optimal parameter values of membership functions

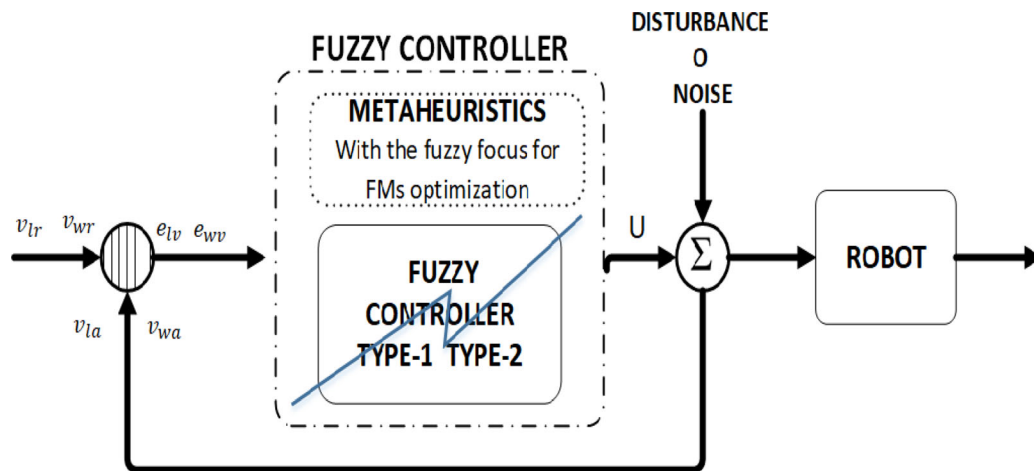


Fig. 8 Proposed methodology

displacement in the Y coordinate, which dynamically changes during the iterations of the algorithm in the adjustment of the values of the parameters that form the membership functions of the fuzzy controller, using error data interpretation.

In Fig. 10, the best result of 35 executions is shown, without noise disturbances, where the continuous red line shows us the desired trajectory, and the blue dotted line denotes the real trajectory of the robot. At first glance, it highlights how the robot has good performance when

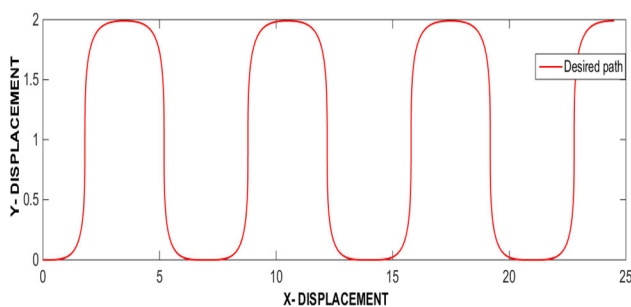


Fig. 9 Desired path

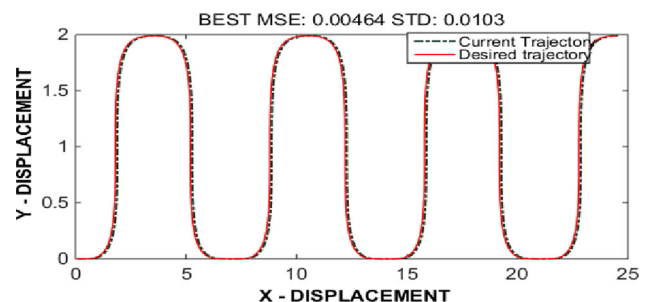


Fig. 10 Graph of the real and desired path using the mean square error (MSE) for comparison

optimally following a given trajectory, generating an MSE of 4.64×10^{-03} and a standard deviation of 1.03×10^{-02} .

Figures 11 and 12 show the design of the input variables, e_{lv} and e_{wv} , where each input is similarly granulated into three membership functions, trapezoidal at the ends and triangular in the middle. Standardization of the response values of the output variables was carried out, obtaining only values in the range of $[-1, 1]$. Both outputs are granulated into three membership functions of the triangular type, and the output variables are shown in Figs. 13 and 14, for PM_1 and PM_2 , respectively. The proposed controller performs control on the right and left motors with the parmotor or torque of the mobile robot using as input the error data. The structure of the fuzzy system is shown in Fig. 15.

Table 1 contains the fuzzy rule set of the proposed fuzzy controller (T1FLC and IT2FLC), where the relationship between the linguistic variables is described, e_{lv} represents the linear velocity error and e_{wv} is the angular velocity error. First of all, the designed T1FLC is used directly in the mobile robot plant for simulation tests, integrating a Mamdani type-1 fuzzy model as shown below:

Rule_n : if e_{v1} is $e_{v1(i)}$ and e_{wv} is e_{wv_k} then PM_1 is $PM_{1(i,k)}$ and PM_2 is $PM_{2(i,k)}$

$n = 1, 2, 3, \dots, 9(ninerules)$; $i = 1, 2, 3$; $k = 1, 2, 3$; because e_{v1} and e_{wv} have three membership functions each. The $e_{v1(i)}$ and e_{wv_k} correspond to fuzzy sets of the inputs e_{v1} and e_{wv} . Similarly, the $PM_{1(i,k)}$ and $PM_{2(i,k)}$ correspond to fuzzy sets of the outputs PM_1 and PM_2 , respectively.

3.2 Dynamic Parameter Setting of Membership Functions Using SSOV

This section mainly describes how the dynamic adjustment of the parameter values of the MFs is developed in the fuzzy controllers. We made use of the variant of the SSO algorithm for the selection and search of the values of the parameters that form the already mentioned membership

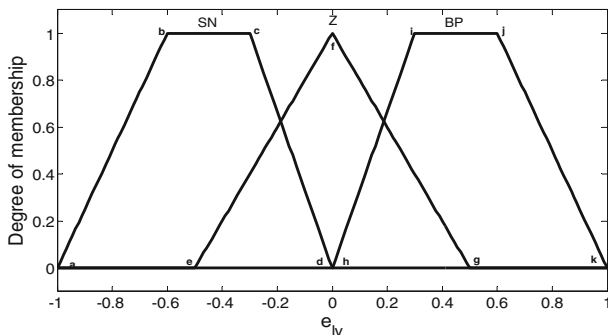


Fig. 11 Input 1; e_{lv}

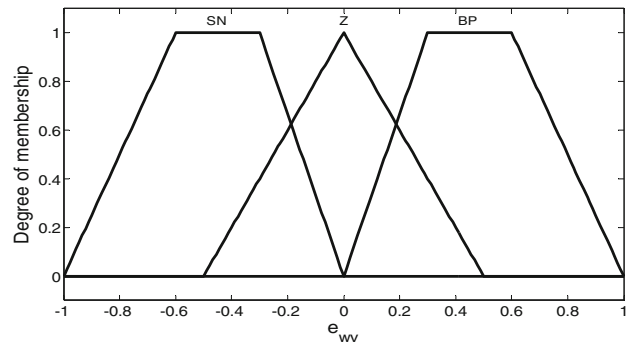


Fig. 12 Input 2; e_{wv}

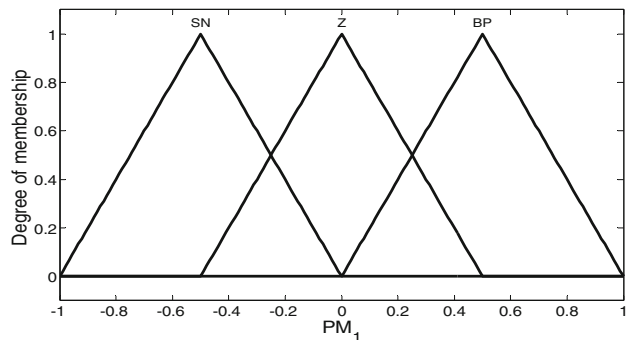


Fig. 13 Output 1; PM_1

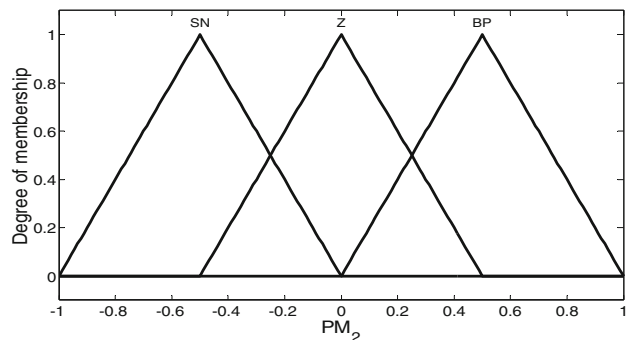


Fig. 14 Output 2; PM_2

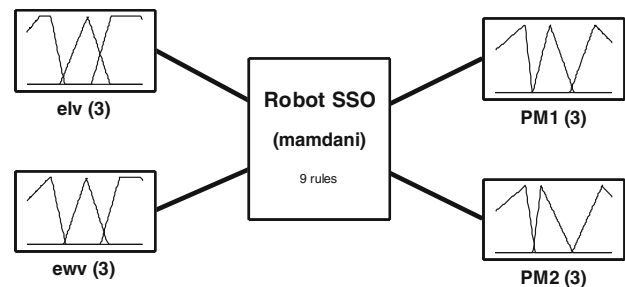


Fig. 15 Type-1 FLC design for the robot controller

Table 1 Fuzzy rules set

If (e_{v_l} is SN) and (E_{WV} is SN) then (PM ₁ is SN) and (PM ₂ is SN)
If (e_{v_l} is SN) and (E_{WV} is Z) then (PM ₁ is SN) and (PM ₂ is Z)
If (e_{v_l} is SN) and (E_{WV} is BP) then (PM ₁ is SN) and (PM ₂ is BP)
If (e_{v_l} is Z) and (E_{WV} is SN) then (PM ₁ is Z) and (PM ₂ is SN)
If (e_{v_l} is Z) and (E_{WV} is Z) then (PM ₁ is Z) and (PM ₂ is Z)
If (e_{v_l} is Z) and (E_{WV} is BP) then (PM ₁ is Z) and (PM ₂ is BP)
If (e_{v_l} is BP) and (E_{WV} is SN) then (PM ₁ is BP) and (PM ₂ is SN)
If (e_{v_l} is BP) and (E_{WV} is Z) then (PM ₁ is BP) and (PM ₂ is Z)
If (e_{v_l} is BP) and (E_{WV} is BP) then (PM ₁ is BP) and (PM ₂ is BP)

functions. Similar methodologies have recently been implemented in other works [38]. The main idea of choosing and creating a variant algorithm (VSSO) is that there is no published work on this algorithm using fuzzy logic in the optimization of membership functions, which is our proposal. There also exist other variants of SSO [39, 40], and we consider their implementation and analysis to be a good idea to demonstrate that good results can be achieved in comparison with other metaheuristics, but this can constitute possible future work.

3.2.1 The Location Search of the Shark Position

3.2.1.1 Autonomous Mobile Robot General Description

The control scheme for the AMR is established as follows: A differential robot model has been used, which consists of two independent driving wheels implemented on the same axle and as the support a free front wheel, [41, 42]. We define the problem to be solved as an autonomous robot with independent wheels with the ability to follow a given path in a fixed or dynamic environment. The robot’s structure is symmetrical on the perpendicular axis, and its center of mass has its location at the geometric center of the robot. Figure 16 graphically describes the differential robot model. For practicality in the robot model, we assume that we can ignore the dynamics of the freewheel, as shown in Eq. (13) [43].

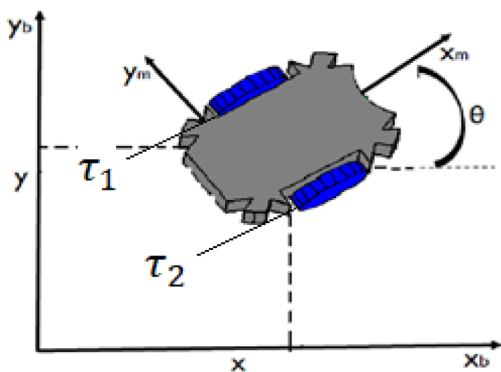


Fig. 16 Wheeled Mobile Robot Model

$$M_{(q)}\dot{v} + C_{(q,\dot{q})}v + Dv = \tau + P(t) \tag{13}$$

$$\dot{q} = \underbrace{\begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix}}_{J(q)} \underbrace{\begin{bmatrix} v \\ w \end{bmatrix}}_v \tag{14}$$

where $q = (x, y, \theta)^T$ describes a vector of the configuration coordinates of the model, $v = (v, w)^T$ contains the vector of system velocities, $\tau = (\tau_1 \tau_2)$ is the vector of torque response values or individual torque to the two wheels of the differential autonomous mobile robot, where τ_1 and τ_2 denote the intensity of the traction on the right or left wheels, $P \in R^2$ describes the uniformly bounded disturbance vector, $M_{(q)} \in R^{2 \times 2}$ describes the positive-definite inertia matrix, $C_{(q,\dot{q})}v$ is the vector of centripetal and Coriolis forces, and $D \in R^{2 \times 2}$ a diagonal positive-definite damping matrix.

According to [37], the kinematics of the system is described in Eq. (14), here (x, y) indicates the position in the $X - Y$ (world) reference frame; θ is the angular difference given between the direction to follow and the x -axis; v and w derivatives of the velocity vector are identified as the linear and angular velocities, respectively. Furthermore, Eqs. (13) and (14) have the following non-holonomic constraint (Eq. 15):

$$y \cos \theta - x \sin \theta = 0 \tag{15}$$

An ideal condition is displayed so that the wheel does not slip, avoiding skidding towards the sides of the robot [13]. So Eq. (14) does not satisfy the necessary Brockett condition for stabilization of the feedback in the system [39], being able to conclude that we do not have a continuous static state feedback controller that stabilizes the closed-loop system, so there is no equilibrium point.

The control objective is to design a fuzzy logic controller such that τ or PM ensures given a path $q_d(t)$ and the desired orientation. A fuzzy logic controller must be designed to apply an adequate Parmotor (PM), such that the measured positions $q(t)$ reach the reference trajectory $q_d(t)$ that is the following control goal (Eq. 16):

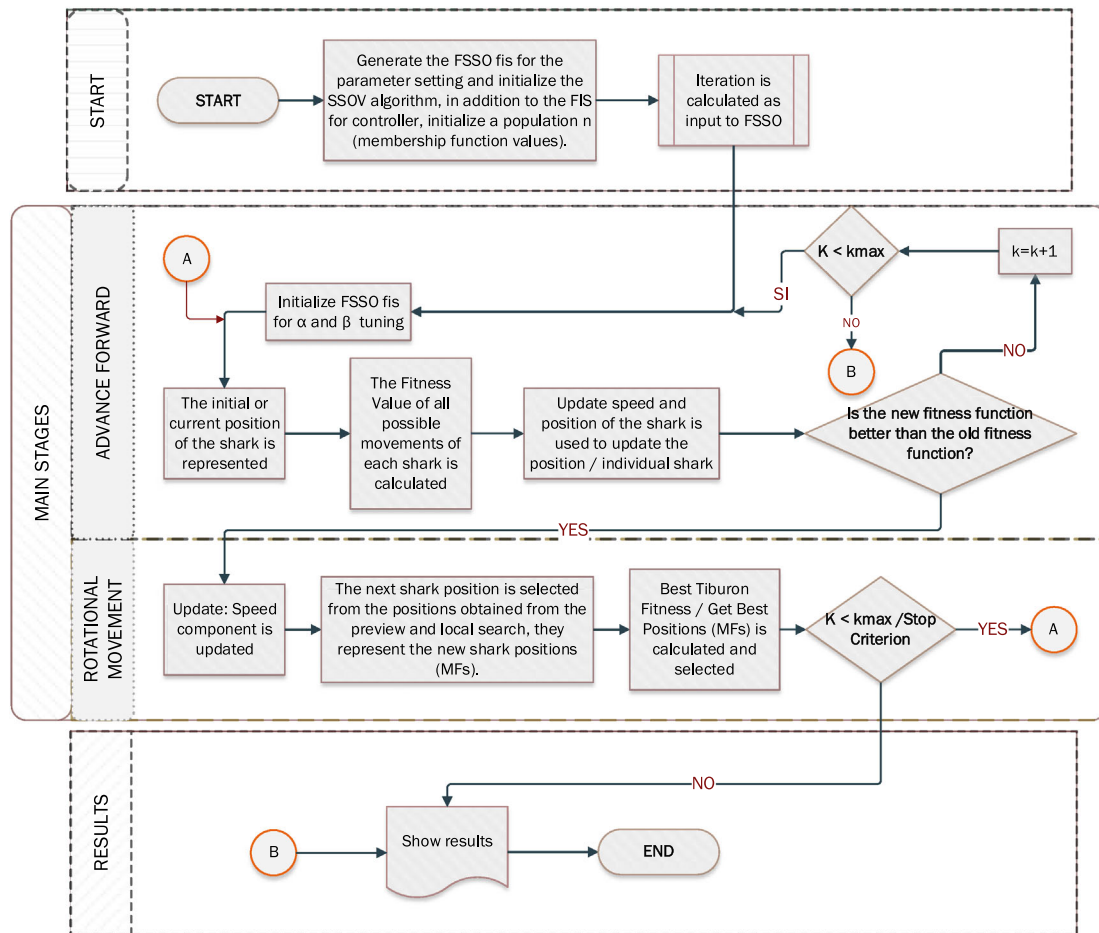


Fig. 17 Procedure to first adjust the parameters of the VSSO algorithm and finally apply it to the fuzzy controller

$$\lim_{t \rightarrow \infty} \|q_d(t) - q(t)\| = 0 \quad (16)$$

Below we can find the structure for achieving the stabilization of the differential robot that is following the desired path. The backward approach is used to design the stabilization control law in Eqs. (13) and (14), since the velocity vector v indirectly controls the kinematics subsystem. The overall controller design consists of two steps:

1. A reference velocity vector is applied $v_r = v$ such that the kinematics model (Eq. 14) be uniformly asymptotically stable.
2. Design a velocity controller τ or PM by using FLC that ensures:

$$v_r(t) - v(t) = 0, \forall t \geq t_s \quad (17)$$

where t_s is the accessibility time.

Equation (15) proposes that real AMRs have driving wheels, where the control input is PM, which must be utilized to stabilize the dynamics (Eq. 13), without destabilizing the system (Eq. (14)), by forcing $v \in R^2$ to reach the reference velocity vector $v_r \in R^2$ in a finite-time.

Roughly speaking, if (Eq. 15) is asymptotically satisfied (*i.e.*, $t_s = \infty$), then v along $t < \infty$. Thus, the mobile robot will be neither positioned nor oriented at the desired point. Figure 8 shows the feedback connection that implies the fuzzy controller in their process.

3.2.1.2 Optimal Membership Function Parameter Reconfiguration Using the Variant Algorithm (VSSO) The previous version of our fuzzy controller to adjust [42] was of the Takagi–Sugeno type, and for simplicity, we decided to modify it and make it of the Mamdani type, so that the input and output parameter values are represented by linguistic variables.

For simplification, fixed parameters (a, k) are established, which correspond to the initial-final parameters of the fuzzy sets, which is as a restriction on the input variables. By means of the VSSO metaheuristic, we find the optimal values of the remaining parameters, namely $b, c, d, e, f, g, h, i, j$, in a simple way (Fig. 11). Regarding the membership functions of the output variables, the algorithm will search for the optimal values of the parameters

Table 2 Comparison between methods: WDO: Wind-driven Optimization; VSSO: Shark Smell Optimization Variant

Characteristics Population	WDO Air package	SSOV Shark Smell V
New speed	u_{t+1}^i	v_{ij}^k
Current speed	u_t^i	v_i^k
Actual position	x_t^i	x_i^k
Next position	x_{t+1}^i	x_{i+1}^k
Better experience	–	y_i^{k+1}
Best group experience	–	$z_i^{k+1,M}$
Increase	T	k
Uniform random numbers between 0 and 1	–	r_1, r_2, r_3

(*b, c, d, e, f, g, h* except *a, i*, which remain fixed for simplicity) and the span of each one. The controller receives as input variables e_{lv} and e_{wv} (Figs. 11 and 12), that is, the difference between the predefined desired speed and the actual speed of the plant, and as output variables, the right PM_1 and left PM_2 parmotors of the above-mentioned robot (Figs. 13 and 14).

The next step is to define an appropriate objective function to evaluate the performance of the VSSO meta-heuristic with respect to the final fit of the membership functions of the controller output variables. Where the objective function represents the degree to which the

solution of the problem under consideration is solved. For this reason, Eq. (18) describes the root mean square error used to evaluate the suitability of the fuzzy system.

$$MSE = \frac{1}{N} \sum_{k=1}^N [y(k) - \tilde{y}(k)]^2 \tag{18}$$

where $y(k)$ is the reference value at instant k ; $\tilde{y}(k)$ is the computed output of the system at instant k ; N number of the samples considered. Since the system can control the linear (v) and angular (w) velocities of the plant, the total error is given by (Eqs. 19–21):

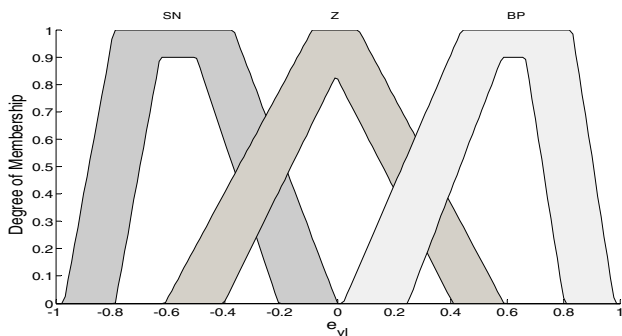


Fig. 18 Footprints of IT2 MFs for input e_{lv}

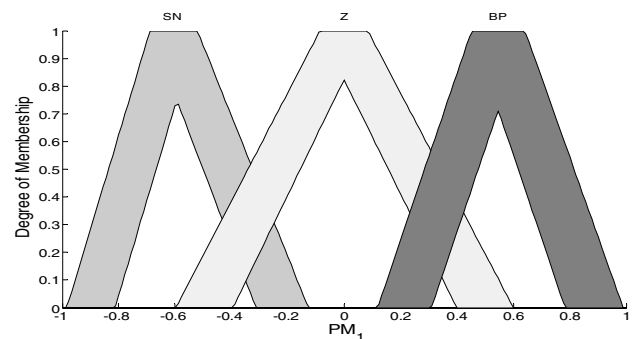


Fig. 20 Footprints of IT2 MFs for output PM_1

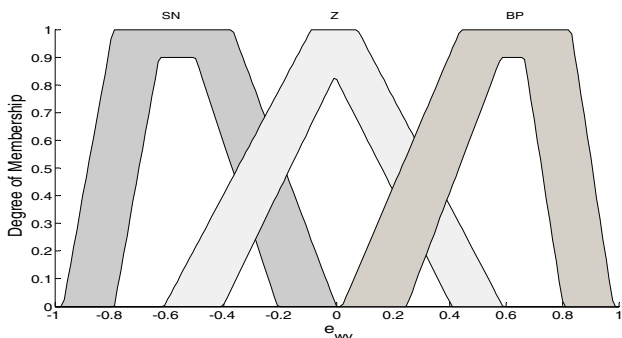


Fig. 19 Footprints of IT2 MFs for input e_{wv}

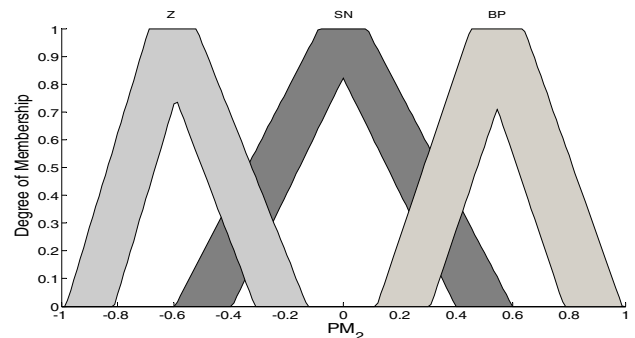


Fig. 21 Footprints of It2 for the output PM_2

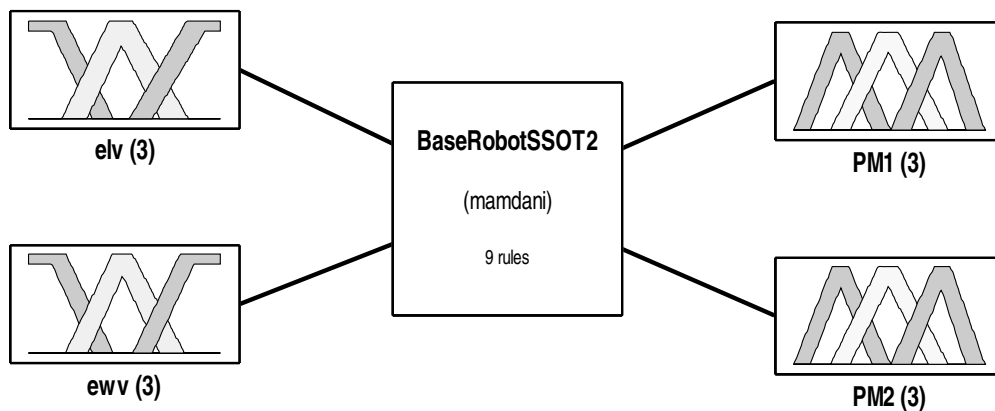


Fig. 22 IT2FLC architecture

$$MSE_v = \frac{1}{N} \sum_{k=1}^N [v(k) - \tilde{v}(k)]^2 \tag{19}$$

$$MSE_w = \frac{1}{N} \sum_{k=1}^N [w(k) - \tilde{w}(k)]^2 \tag{20}$$

$$E_{Total} = MSE_v + MSE_w \tag{21}$$

This is used to represent the total distance between the points on each MF plot of the controller’s inputs and outputs, generated by shark algorithm. Thus, for the development of the Mamdani type-1 and interval type-2 fuzzy systems, with the dynamic parameters adjustment, and the two measurements described as linear velocity error (e_{lv}) and angular velocity error (e_{wv}) were considered as the inputs. The system has the fuzzy outputs: Parmotor1 (PM₁) and Parmotor 2 (PM₂).

Regarding the input values to the fuzzy system, the linguistic error variables have a defined range of possible values that can go from -50 to 50 , with e_{lv} and e_{wv} , and the normalization of the values of these variables was formulated obtaining values in a range $[-1, 1]$. Equations (22) and (23) describe how the normalization of the linear velocity and angular velocity errors is performed, respectively.

$$e_{lv} = \frac{V_{Ref} - V_{Actual}}{50} \tag{22}$$

$$e_{wv} = \frac{W_{Ref} - W_{Actual}}{50} \tag{23}$$

We establish the following sequence of steps with the VSSO algorithm in the approach of the proposed method. Figure 17 graphically describes the order of the steps to be followed in the proposed algorithm.

We establish the following sequence of steps with the VSSO algorithm in the approach of the proposed method.

- Step 1 random initial population of point particles is generated for the MF values that satisfy the constraints. The particle has a position in its memory, and that position becomes a possible solution to the problem using Eq. (2), representing the values of the point parameters of the membership functions of the fuzzy controller. We generate the fuzzy system, this is for the parameter setting and initialization of the VSSO algorithm, in addition to the fuzzy controller. We establish a lower and upper limits from -1 to 1 , for normalizing the range of the membership functions in the input and output fuzzy sets.
- Step 2 For this case, dimensions of size 40 are needed for T1FLC and 80 for IT2FLC, where each point value of the membership function represents one dimension. Also, perform the iteration calculation using the current iteration (k) divided by the total iterations to be performed (Kmax) and set the calculation value, as the input for fis FSSO and evaluate the initial positions fit value using Eqs. (2) and (3).
- Step 3 The fitness of each particle is evaluated, and each one of the point parameters that form the MFs is identified using the metaheuristic variant (VSSO) and then new solutions are evaluated again using Eqs. (4) and (6). The new value of the MF parameters is represented by each shark position.
- Step 4 The most appropriate positions of each shark are taken at each stage, while dynamically adjusting each of the values that represent the parameter values of the MFs in the Fuzzy Inputs and Outputs.

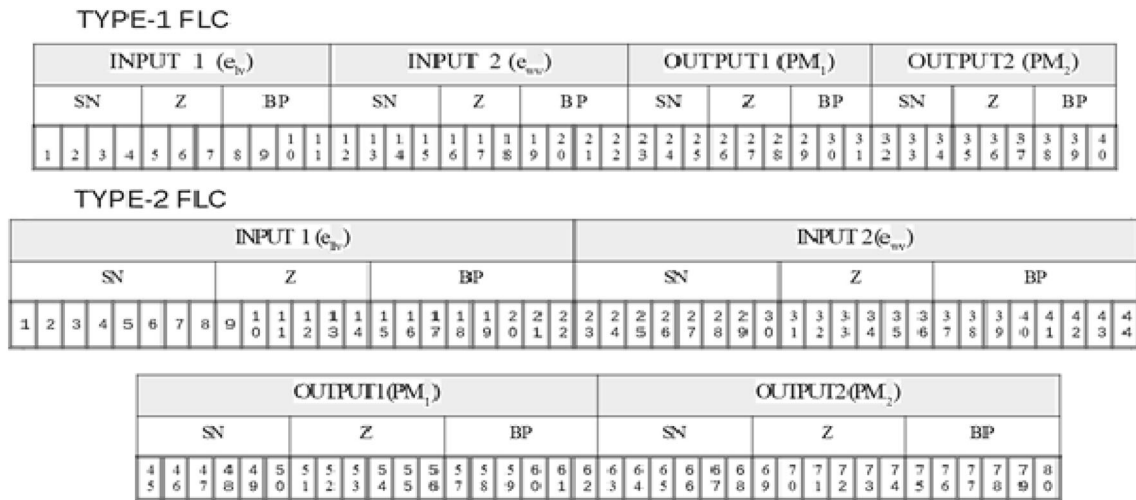


Fig. 23 Vector of smell particles to adjust the membership functions of fuzzy controllers of T1 and IT2

Table 3 Adjustable parameters of the type-1 and type-2 inputs before optimization

Type-1 FLS					Type-2 FLS								
Inputs	MFs	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₁	<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂	<i>d</i> ₂
<i>e_{vi}</i>	SN	-1	-	-0.3	0	-0.97	-0.80	-0.53	-	-0.79	-0.61	-0.37	-
	Z	-	0.6	0.5	-	-0.61	-0.09	0.40	0.27	-0.40	-0.07	0.59	-
	BP	0	0.3	0.6	1	0.02	0.44	0.65	0.80	0.24	0.62	0.83	0.98
<i>e_{wv}</i>	SN	-1	-	-0.3	0	-0.97	-0.80	-0.53	-	-0.79	-0.61	-0.37	-
	Z	-	0.6	0.5	-	-0.61	-0.09	0.40	0.27	-0.40	-0.07	0.59	-
	BP	0	0.3	0.6	1	0.02	0.44	0.65	0.80	0.24	0.62	0.83	0.98

- Step 5 The new value of the fitness function is evaluated using Eq. (7), to verify if it is better than the previous value of the fitness function. If the condition is satisfied, then update the final position to each of the points in the fuzzy system membership function.
- Step 6 Select the best position for local search using Eq. (8). The selected smell particle contains the best position found so far and to define the fitness of each shark position, which is made by the mean square error of Eq. (18). The error between the ideal values and the output of the fuzzy controller is the fundament for the calculation of the objective function (OF) is given by Eqs. (19) to (21).

- Step 7 Evaluate the stopping criteria. If it is fulfilled, the search ends, otherwise, increase the stage $K = K + 1$ to K_{max} and then display and save the optimal values found.

Both fuzzy controllers are of Mamdani form with type-1 and type-2 fuzzy logic, where the parameter values that make up the membership functions are dynamically changed with the VSSO metaheuristic and evaluated using nine fuzzy rules that are based on prior knowledge of the problem. The VSSO metaheuristic algorithm is based on PSO, which is also biologically inspired by nature [33, 44], clarifying that PSO is only referenced, but it is not used in the development of this methodology. Table 2 is used to make a comparison between the parameters of the metaheuristic algorithms, which are VSSO (proposed) and wind-driven optimization (WDO).

Table 4 Adjustable parameters of the type-1 and type-2 outputs before optimization

Type-1 FLS		Type-2 FLS								
Outputs	MFs	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁	<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>PM</i> ₁	SN	- 1	- 0.5	0	- 0.98	- 0.69	- 0.31	- 0.81	- 0.52	- 0.13
	Z	- 0.5	0	0.5	- 0.59	- 0.09	0.40	- 0.40	0.08	0.60
	BP	0	0.5	1	0.12	0.45	0.78	0.31	0.64	0.99
<i>PM</i> ₂	SN	- 1	- 0.5	0	- 0.98	- 0.69	- 0.31	- 0.81	- 0.52	- 0.13
	Z	- 0.5	0	0.5	- 0.59	- 0.09	0.40	- 0.40	0.08	0.60
	BP	0	0.5	1	0.12	0.45	0.78	0.31	0.64	0.99

3.3 Interval Type-2 Fuzzy Controller

The concepts of type-1 and type-2 fuzzy logic systems (T2FLS) were coined between 1965 and 1975 by Zadeh [28]. This section is aimed at providing a basic understanding of the operation of an IT2FLS through a precise description of its function. It is by granulating the input domains of a T1FLS baseline with a footprint uncertainty (FOU) that we obtain the T2FLS. The interval type-2 fuzzy logic system (IT2FS) input variables and their ranges are developed to follow the path. The membership functions of

Table 5 Parameters used VSSO and WDO method for fuzzy controller IT2FLC

Population	Iterations	Dimensions
30	1500	40

Table 6 Results of the experiments with the fuzzy controller IT2FLC, obtained with VSSO

Experiment	MSE	Experiment	MSE
1	1.2679 × 10 ⁻⁰⁶	16	6.3590 × 10 ⁻⁰⁴
2	1.3972 × 10 ⁻⁰⁴	17	1.3007 × 10 ⁻⁰⁵
3	6.7674 × 10 ⁻⁰⁴	18	4.7000 × 10 ⁻⁰⁴
4	1.1398 × 10 ⁻⁰⁵	19	8.4020 × 10 ⁻⁰³
5	3.7205 × 10 ⁻⁰⁴	20	4.7000 × 10 ⁻⁰³
6	2.4143 × 10 ⁻⁰⁵	21	1.4700 × 10 ⁻⁰⁴
7	2.9927 × 10 ⁻⁰⁶	22	5.1500 × 10 ⁻⁰³
8	5.0183 × 10 ⁻⁰⁴	23	7.0000 × 10 ⁻⁰³
9	2.9581 × 10 ⁻⁰⁵	24	3.7010 × 10 ⁻⁰³
10	5.6985 × 10 ⁻⁰⁶	25	5.9000 × 10 ⁻⁰³
11	6.5592 × 10 ⁻⁰⁶	26	2.7000 × 10 ⁻⁰³
12	4.0882 × 10 ⁻⁰⁵	27	8.2560 × 10 ⁻⁰⁴
13	1.8416 × 10 ⁻⁰⁶	28	6.5050 × 10 ⁻⁰⁴
14	1.1593 × 10 ⁻⁰⁵	29	2.8000 × 10 ⁻⁰³
15	3.6720 × 10 ⁻⁰⁵	30	5.4490 × 10 ⁻⁰⁴

Bold indicates the best values of the Tables, which are the lowest values in this case

the input variables are shown in Figs. 18 and 19, and for the outputs, in Figs. 20 and 21.

First, the overall performance of the robot we call it “behavior” is evaluated in the task of tracking and following a trajectory. The design of the controller that takes control of the actions of the mobile robot is presented in Fig. 22. In this case, the selection of the appropriate actions that are transmitted to the system is defined by the controller’s rule set. The control structure for the basic behaviors is based on a interval type-2 fuzzy logic controller supported by a set of IF–THEN fuzzy rules [45–47].

According to the uncertainty of the problem, the membership functions acquire distributed values. From the design, three MFs are used for the *e*_v input (SN: Small Negative, Z: Zero, BP: Big Positive), three MFs for the *e*_{wv} input (SN: Small Negative, Z: Zero, BP: Big Positive), three MFs for both the *PM*₁ and *PM*₂ outputs (SN: Small Negative, Z: Zero, BP: Big Positive), are built. Nine fuzzy rules were designed for the T2FLS for route tracking, and they are determined by expert knowledge on the control process. The fuzzy rules are used to evaluate the intensity of the torque of the right and left wheels and can generate changes in the direction to follow the trajectory when the robot wheels interact, and these rules are described in Table 1.

4 Results

This section presents the simulation results of the final adjustment of the fuzzy controller membership functions, to evaluate the performance of the proposed approach in the differential mobile robot model, defined by Eqs. (13) and (14).

The desired path for the robot is defined by the following expressions (Eq. 24):

$$v_d(t) = \begin{cases} v_d(t) = 0.4 * \cos(0.2(t)) \\ w_d(t) = 0.3 * \sin(0.2(t)) \end{cases} \tag{24}$$

The optimization approach generates a vector of values that are presented as the dynamically found values for the optimized membership functions, shown in Fig. 23.

Table 7 Adjusting parameters of the type-1 and IT2FLC inputs after optimization with VSSO

Type-1 FLS						Type 2 FLC							
Inputs	MFs	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁	<i>d</i> ₁	<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂	<i>d</i> ₂
<i>e_{vl}</i>	SN	- 1.12	- 0.85	- 0.67	- 0.35	- 1.01	- 1.01	- 0.74	- 0.27	- 1	- 1	- 0.72	- 0.05
	Z	- 0.49	0.03	0.70	-	- 0.65	- 0.02	0.41	-	- 0.46	- 0.01	0.79	-
	BP	- 0.11	0.41	0.93	1.19	0.05	0.65	0.96	1.57	0.16	0.90	1.29	1.9
<i>e_{wv}</i>	SN	- 1.20	- 0.88	- 0.70	- 0.28	- 1.01	- 1.01	- 0.78	- 0.09	- 1	- 1	- 0.69	- 0.06
	Z	- 0.34	- 0.03	0.42	-	- 0.65	- 0.04	0.35	-	- 0.45	- 0.01	0.78	-
	BP	0.26	0.60	0.97	1.24	- 0.08	0.55	1.05	1.05	0.27	0.57	1.06	1.06

Table 8 Adjustment obtained for type-1 and IT2FLC output parameters after optimization with VSSO

Type-1 FLS					Type-2 FLS					
Outputs	MFs	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁	<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₂
<i>PM</i> ₁	SN	- 1.2	- 0.63	- 0.37	- 1	- 0.60	- 0.32	- 0.83	- 0.57	- 0.14
	Z	- 0.47	- 0.10	0.36	- 0.61	0.06	0.41	- 0.48	0.07	0.61
	BP	0.28	0.70	1.20	0.13	0.46	0.8	0.22	0.48	0.99
<i>PM</i> ₂	SN	- 1.2	- 0.50	- 0.28	- 0.99	- 0.60	- 0.33	- 0.76	- 0.55	- 0.15
	Z	- 0.40	- 0.18	0.29	- 0.46	0.02	0.41	- 0.37	0.02	0.72
	BP	0.24	0.80	1.21	0.01	0.63	0.8	0.15	0.64	1

The FLC design for the AMR controller has trapezoidal and triangular membership functions at the inputs with four and three parameter values, respectively (see Figs. 11, 12, 15). Three triangular MFs in each output with three values for each triangular membership function (see Figs. 13, 14, 15), obtaining a total of 40 values for the vector of the Type-1 FLC. On the other hand, when the IT2FLC is used, each trapezoidal MF needs eight values and each triangular MF needs six values (see Figs. 18, 19, 20, 21), obtaining 80 values in the vector for the design of an IT2FLC (Fig. 22). The adjustable parameters: *a*, *b*, *c* and *d*, of the type-1 and type-2 inputs and outputs are listed in Tables 3 and 4, respectively, which are adjusted in the next subsection 4.1 using the proposed VSSO algorithm.

4.1 Results of Adjustment of Fuzzy Controllers Parameters

The parameters used in the VSSO and WDO metaheuristics are described in Table 5. Tables 6, 7, 8 and 9 are presenting the details of the 30 experiments carried out in the search for the best optimized fuzzy system, using the MSE (The objective function) as the metric. Tables 7, 8, and 10, 11 show the best parameters for the controller, obtained from the results after optimization. Figures 24, 25, 26, 27 show the inputs and outputs of the optimized controller with VSSO and Figs. 28, 29, 30, 31 optimized by WDO.

Figures 24, 25, 26, 27, 28, 29, 30 and 31 how the adjustment that was made to the values of the parameters that form the membership functions of the inputs and outputs in the fuzzy controller by the VSSO and WDO metaheuristics, respectively. Looking closely at the differences generated between each of them and how they perform, the improvement in the error generated by the proposed controller when simulating is notable. In addition, as has already been demonstrated in previous fuzzy control works, the cause for this is the overlap between the membership functions, which helps the robot to improve the tracking of the desired path, in contrast to the other method (WDO) that does not achieve this in all its membership functions. It can be noted in Tables 6 and 9, that there is less variation in the results obtained with the VSSO metaheuristic since it has the best MSE of 1.2679e⁻⁰⁶, but the worst is of 2.800e⁻⁰³, while the WDO metaheuristic has a greater variation in results; the best MSE with 1.90e⁻⁰⁵, a worst of 5.7094e⁻⁰². Based on the obtained results, the proposed VSSO metaheuristic can be viewed as the best one to optimize this specific problem. Table 12 describes the average results of the optimization methods without disturbance.

4.2 Statistical Comparison of Results

A dynamic tracking control was developed to adjust the membership functions with the VSSO and WDO

Table 9 Results of the experiments with the fuzzy controller IT2FLC, obtained with WDO

Experiment	MSE	Experiment	MSE
1	1.90 × 10 ⁻⁰⁵	16	1.64 × 10 ⁻⁰⁴
2	1.90 × 10 ⁻⁰⁵	17	1.64 × 10 ⁻⁰⁴
3	5.40 × 10 ⁻⁰⁵	18	2.60 × 10 ⁻⁰⁴
4	5.40 × 10 ⁻⁰⁵	19	2.93 × 10 ⁻⁰⁴
5	6.20 × 10 ⁻⁰⁵	20	4.33 × 10 ⁻⁰⁴
6	6.20 × 10 ⁻⁰⁵	21	1.62 × 10 ⁻⁰³
7	6.90 × 10 ⁻⁰⁵	22	1.65 × 10 ⁻⁰³
8	8.70 × 10 ⁻⁰⁵	23	6.21 × 10 ⁻⁰³
9	8.70 × 10 ⁻⁰⁵	24	6.35 × 10 ⁻⁰³
10	9.10 × 10 ⁻⁰⁵	25	7.73 × 10 ⁻⁰³
11	9.80 × 10 ⁻⁰⁵	26	1.67 × 10 ⁻⁰²
12	1.05 × 10 ⁻⁰⁴	27	1.67 × 10 ⁻⁰²
13	1.15 × 10 ⁻⁰⁴	28	2.65 × 10 ⁻⁰²
14	1.38 × 10 ⁻⁰⁴	29	2.92 × 10 ⁻⁰²
15	1.60 × 10 ⁻⁰⁴	30	5.70 × 10 ⁻⁰²

Bold indicates the best values of the Tables, which are the lowest values in this case

Table 10 Adjusting parameters of the input after optimization with T1FLC and WDO

Inputs	Membership functions	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
<i>e_{v1}</i>	SN	- 0.73	- 0.61	- 0.46	- 0.20
	Z	- 0.30	- 0.14	0.17	
	BP	- 0.08	0.23	0.64	1
<i>e_{wv}</i>	SN	- 0.77	- 0.56	- 0.47	- 0.21
	Z	- 0.42	0.01	0.40	
	BP	0.19	0.57	0.94	1

metaheuristics. Each of them was tested, competing with the same parameters, with 30 experiments as the limit. This was used to establish a comparison between the methods based on the statistical z test, detailing the statistical data in Table 13, whose parameters and results are summarized in Table 14. The statistical z-test, with a significance level of 0.05, was done, where the null hypothesis tells us that the average of the proposed VSSO (μ_1) is lower than or equal to the average of WDO (μ_2). In this case, the rejection region for all values that fall below - 1.645.

We apply the statistical Z test as defined by Eq. (25):

$$Z = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sigma_{\bar{x}_1 - \bar{x}_2}} \tag{25}$$

If the Z value falls outside the rejection zone, we fail to reject the null hypothesis. As a conclusion, then we can

Table 11 Adjusting parameters of the outputs after optimization with T1FLC and WDO

Inputs	Membership functions	<i>a</i>	<i>b</i>	<i>c</i>
PM ₁	SN	- 0.67	- 0.42	- 0.29
	Z	- 0.23	- 0.09	0.8
	BP	0	0.5	1
PM ₂	SN	- 0.53	- 0.31	- 0.09
	Z	- 0.21	- 0.05	0.7
	BP	- 0.02	0.59	1

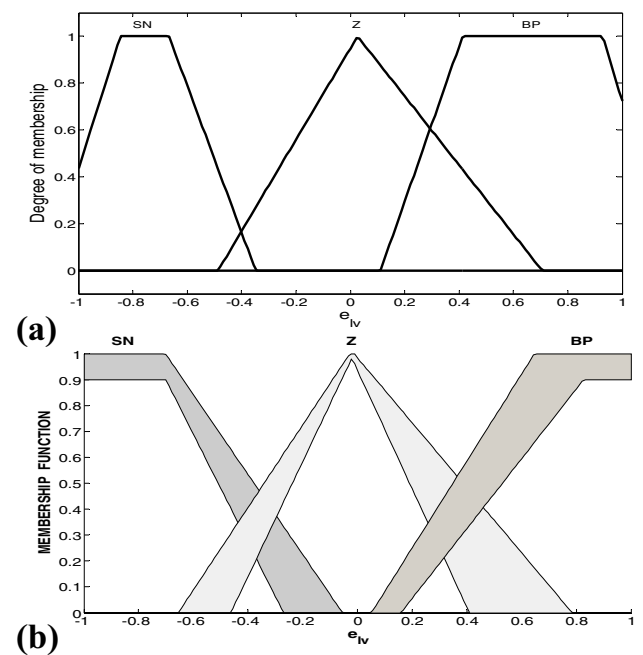


Fig. 24 Input 1 optimized by VSSO algorithm. **a** Type-1, **b** and Type-2

state that there is not enough statistical evidence at a significance level of 5% to support the claim that the mean of the VSSO is lower than WDO.

4.3 Experimental Results Between T1FLC and IT2FLC with Disturbance

An evaluation of the controllers was carried out to analyze their performance in the mobile robot, using IT2FLC in comparison with T1FLC, applying the proposed VSSO method and WDO, using the same number of rules. Disturbances are applied to the robot system by the band-limited white noise block, which generates normally distributed random numbers that are suitable for use in continuous or hybrid systems, with a degree of 0.1 and 0.2 of power level in white noise. This is shown in Figs. 38 and

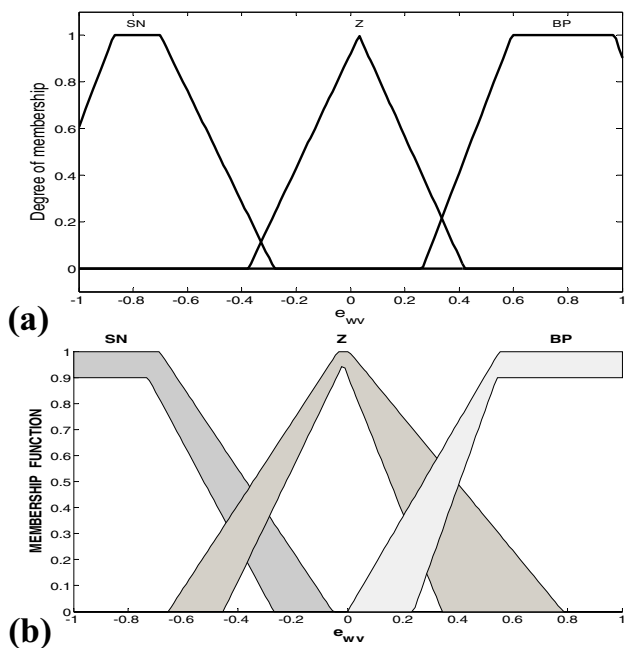


Fig. 25 Input 2 optimized by VSSO algorithm. **a** Type-1, **b** and Type-2

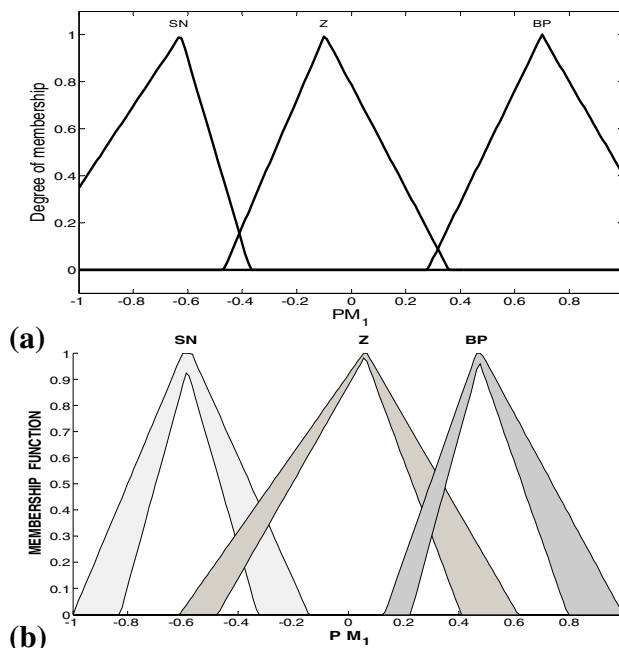


Fig. 26 Output 1 optimized by VSSO algorithm. **a** Type-1, **b** and Type-2

39 (Table 15). The sensitivity of the change in movement of the mobile robot towards the trajectory is evaluated using simulation, which is carried out through the differential platform in Simulink (Fig. 32). The tests were carried out with the same parameters in the different environments, as will be explained later. The collected information for processing will be taken from linear and angular velocity errors when simulated with the differential robot platform in Simulink.

The control surface of the T2FLS is described below, whose inputs are the linear velocity error and angular velocity error, and the output response is the left and right wheel torque.

Figures 33 and 34 show that the IT2FLS when compared to the T1FLS has a smoother surface, where each slope represents a gradual change in the torque of the wheel traction force.

In the search for a way to quantify the behavior of the fuzzy controllers with and without disturbance, performance metrics were established based on the error signal $e(t)$, which is the difference between the desired value of the reference variable of the tracking trajectory and its real value produced by the robot that follows it. The best known metrics are the so-called integral criteria presented in Eqs. (26) to (29):

Integral of absolute error.

$$IAE = \int_0^{\infty} |e(t)| dt \tag{26}$$

Integral of time by absolute error

$$ITAE = \int_0^{\infty} t|e(t)| dt \tag{27}$$

Integral of the quadratic error

$$ISE = \int_0^{\infty} e(t)^2 dt \tag{28}$$

The error is given by:

$$e(t) = r(t) - y(t) \tag{29}$$

The tracking performance of the IT2FLC and T1FLC controllers without noise is illustrated in Fig. 35, which also shows the amplified response of the controllers after application. The system with which this work is carried out is illustrated in Fig. 36. The performance comparison between the VSSO and WDO metaheuristics using white noise is measured by applying the IAE, ISE, and ITAE criteria for 60 s and are shown in Fig. 37. As it is obvious, tracking accuracy, controller error, and time response have better results with VSSO-IT2FLC in comparison with T1FLC (Figs. 38, 39), according to Tables 15 and 16.

In addition to the previous comment, the robustness of the IT2 FLC applying the proposed metaheuristic and WDO proposal is tested by performing several experiments with different levels of white noise and we describe the values of the three used performance metrics below. Table 15 shows the results of ITAE, ISE, IAE metrics of

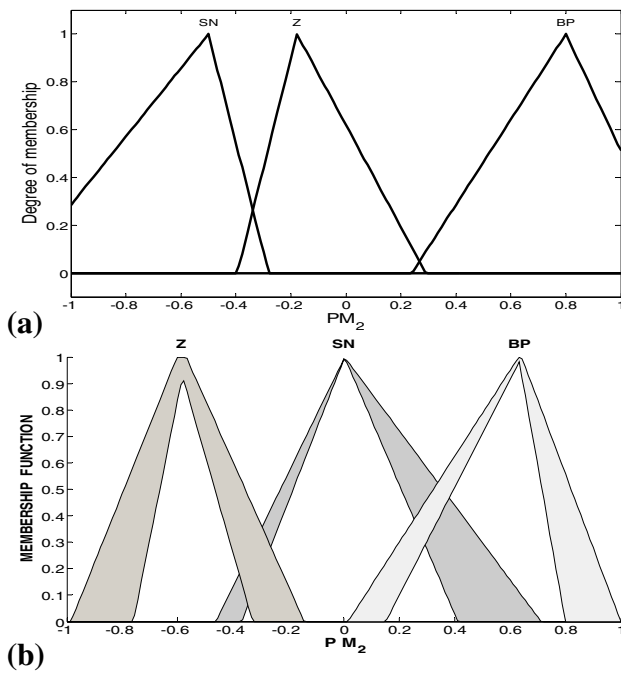


Fig. 27 Output 2 optimized by VSSO algorithm. **a** Type-1, **b** and Type-2

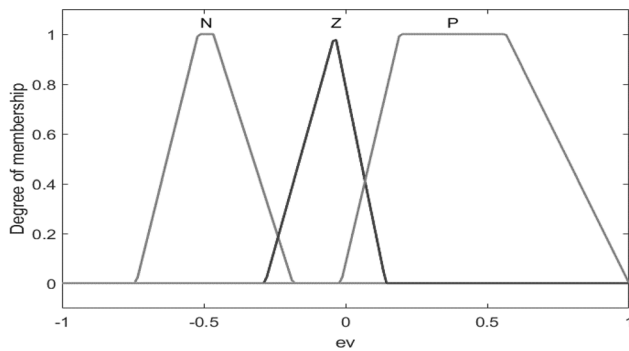


Fig. 28 Input 1 optimized by WDO algorithm

more than 1200 experiments with noise levels ranging from 0 (Zero noise), 0.1 (Low) and 0.2 (High). Table 16 shows a contrast between the response times.

From the linear and angular errors and their integral performance metrics, it follows that the type-2 fuzzy in VSSO has a much better adaptation for fitting the values of the parameters of the FLC. Therefore, we can infer that employing type 2 fuzzy logic leads to a better system response and therefore the fuzzy parameters have been adjusted quickly, while looking for the best vector of solutions for the FLC.

The proposed methodology exhibits a better performance using the VSSO metaheuristic with a fuzzy

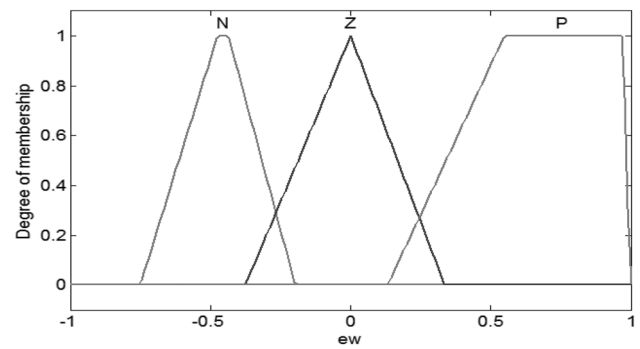


Fig. 29 Input 2 optimized by WDO algorithm

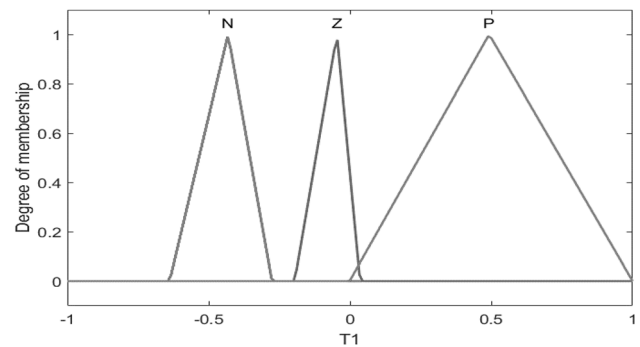


Fig. 30 Output 1 optimized by WDO

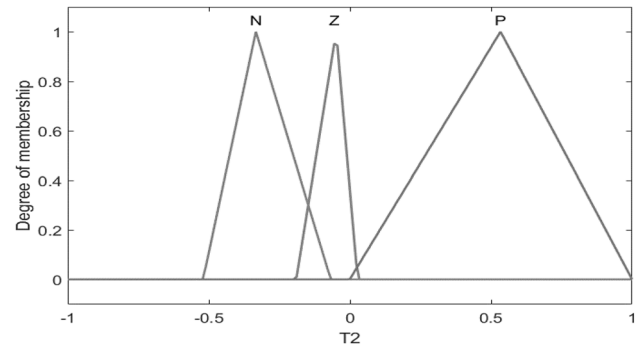


Fig. 31 Output 2 optimized by WDO

approach, when adjusting the parameter values that form the membership functions. The data vector that was obtained was able to optimize the fuzzy controller in such a way that the results were very close to the trajectory that it was given, unlike its counterpart (WDO) which was not as close to the desired trajectory. The details of the implementation can be checked at the following link: <https://drive.google.com/drive/folders/1-bivaFmcjSRURHRC-YzqpVISKCXc58>

Table 12 Results (MSE) of the optimization methods

	VSSO	WDO
Average	1.52×10^{-03}	6.00×10^{-03}
Standard deviation	2.39×10^{-03}	1.26×10^{-02}

Table 13 Statistical data

Variable	Number of samples	Mean	Standard deviation
VSSO	30	0.0015	0.0024
WDO	30	0.0060	0.0126

Table 14 Statistical test

Parameters	Value
z (Observed value)	- 1.494
z (Critica value)	- 1.645
Valor-p (one-tailed)	0.067
Alpha	0.05

5 Conclusion

The fuzzy methodology proposed for the adjustment of the alpha and beta main parameters of the variant of the shark smell optimization algorithm (VSSO), and its application for the optimization of the membership functions of the fuzzy controllers, is the main contribution of this article. Results demonstrate that this is the best method for this task, among other methods considered in this work. The results show that the proposed method is efficient in terms of controller performance and the rejection of noise disturbances. Furthermore, the proposed method, with interval type-2 fuzzy sets, can achieve the parameter adaptation of the fuzzy sets considering the complexities of the model and the uncertainties inherent in the input data. Also, it is

worth mentioning that in the path following by the robot, with T1FLC and T2FLC, it is possible to observe how the values of the parmotors are changing when analyzing the results with the integral parameters applied and the times of traversing the path.

In addition, the gains with IT2FLC are smaller than those of the T1FLC baseline and they gradually decrease as the FOU increases. Therefore, smaller IT2 gains result in even smaller control signals, so an IT2FLC tends to be more resistant to disturbances and its response is slower [48]. The details that the results provide an explanation of the experimental observations that show that an IT2 fuzzy controller has the ability to handle uncertainty and eliminate disturbances present in the system.

For future work, our objective is to test our method in other types of problems that contain higher levels of uncertainty. In addition, performing more experimentation with other optimization methods in adjusting the parameters of the VSSO metaheuristic. In the same way, find other forms of efficiently adjusting the parameter values of the MFs of the fuzzy input and output sets of the fuzzy controller, testing the true efficiency of type-2 fuzzy systems through their application. Also, we are thinking about considering other VSSO parameters that could be dynamically adjusted that may improve results even more.

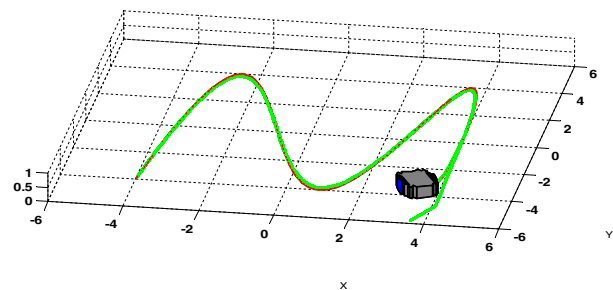


Fig. 32 Robot following the path

Table 15 Values of integral performance metrics for the test plant with 0 to 0.2 white noise power, (tracking for 60 s)

White noise power	Controller type												
	VSSO Type-1 FLC TYPE-1 FLC			VSSO Type-2 FLC TYPE-2 FLC			WDO Type-1 FLC			WDO Type-2 FLC			
	IAE	ISE	ITAE	IAE	ISE	ITAE	IAE	ISE	ITAE	IAE	ISE	ITAE	
Zero	24.53	12.80	749.11	20.24	9.88	597.53	25.09	15.02	15.02	755.29	23.99	12.72	719.91
Low (0.1)	23.50	12.05	704.23	16.02	4.53	480.33	21.71	9.47	9.47	654.63	19.84	8.45	601.88
High (0.2)	11.70	5.10	236.98	3.82	1.86	23.31	20.38	9.36	9.36	609.02	8.22	1.81	229.02

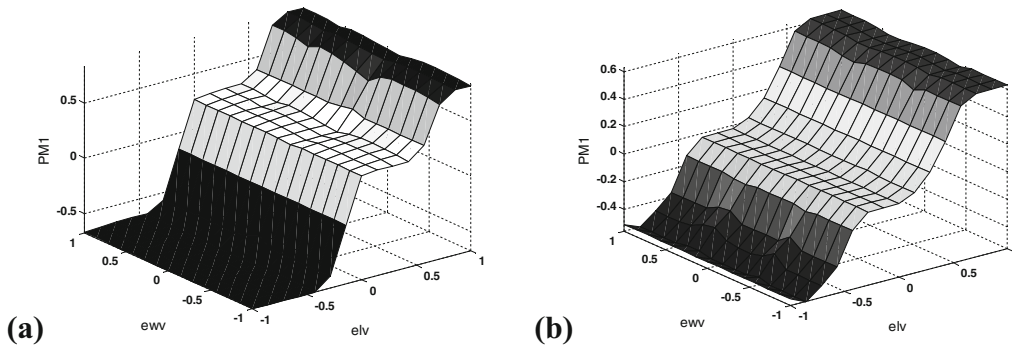


Fig. 33 a T1 and b T2 FLC surface for parmotor 1

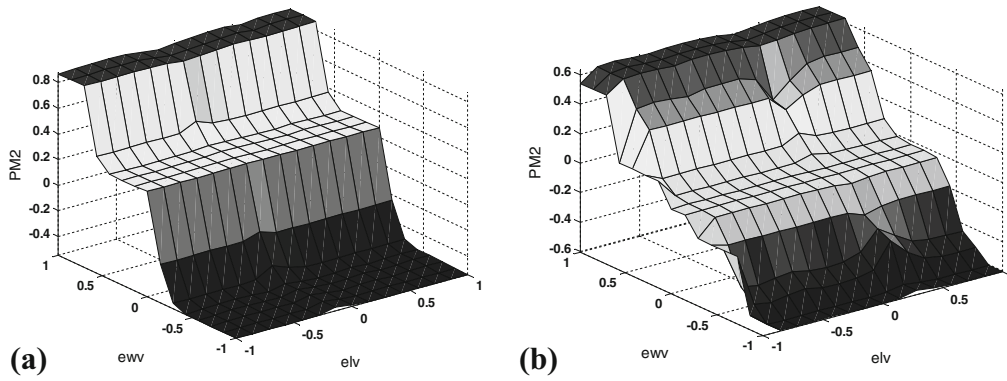


Fig. 34 a T1 and b T2 FLC surface for parmotor 2

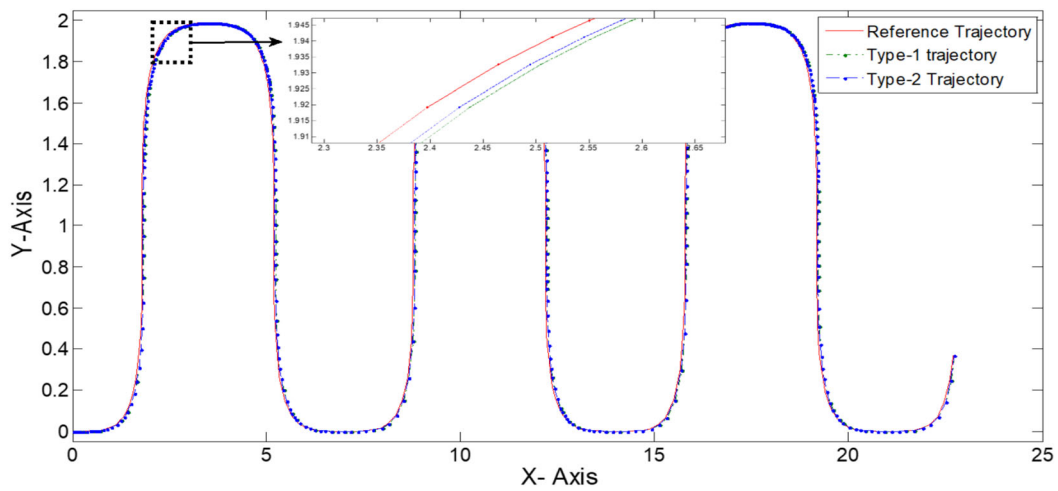


Fig. 35 Fuzzy controllers IT2 and T1 responses relative the reference. Shown Performance for the trajectory times and Observed differences

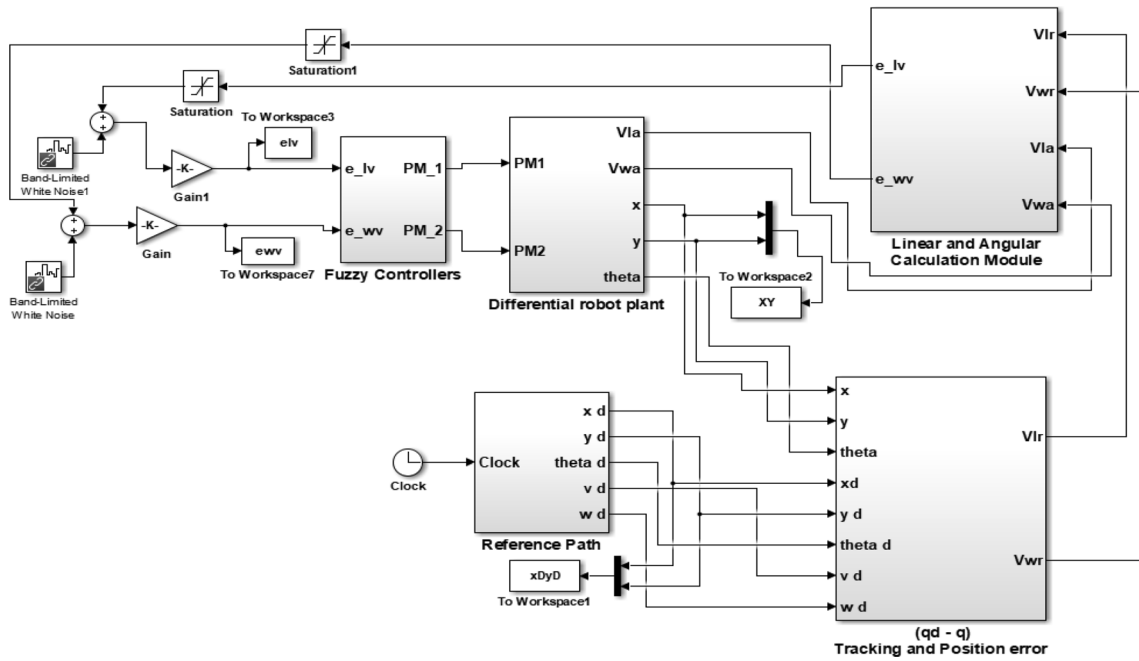


Fig. 36 Design of the mobile robot platform system

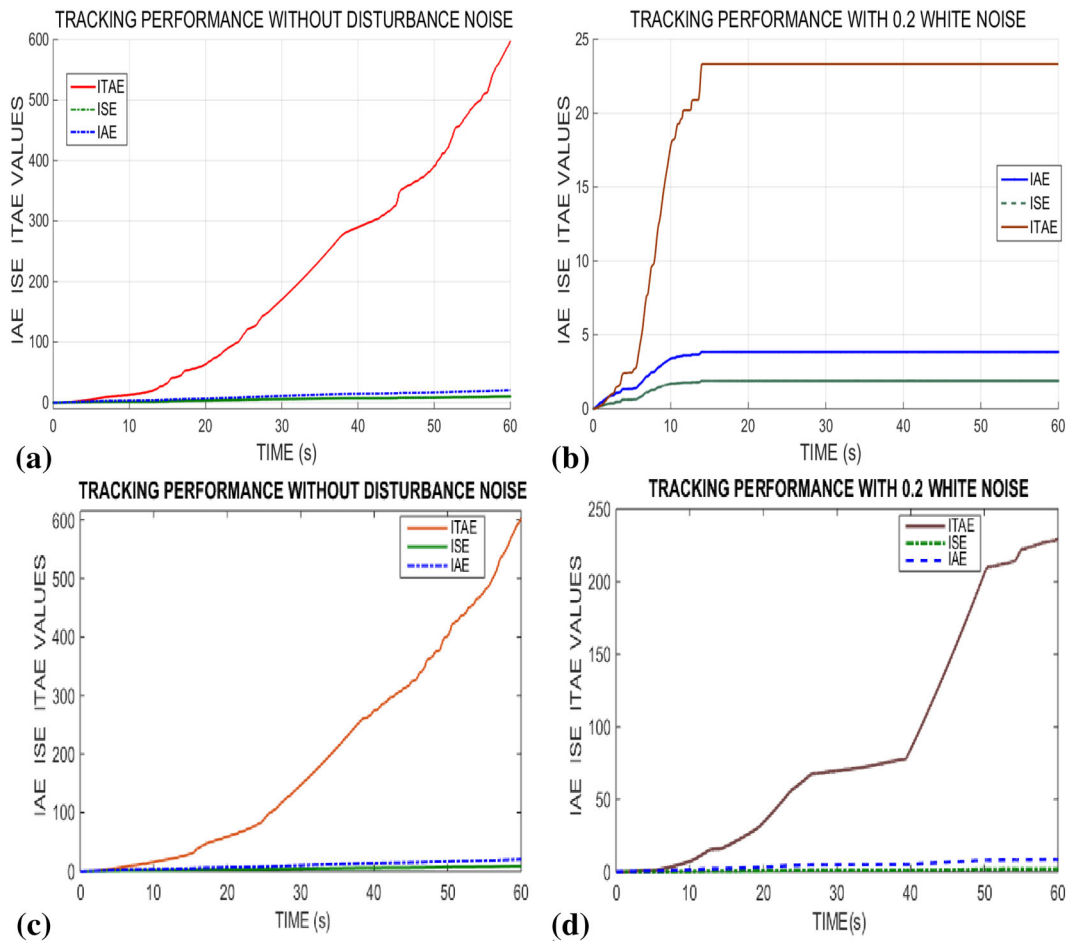


Fig. 37 Tracking performance a VSSO Without noise, b VSSO With High noise, c WDO Without noise, d WDO With High noise, for IT2FLS tuning

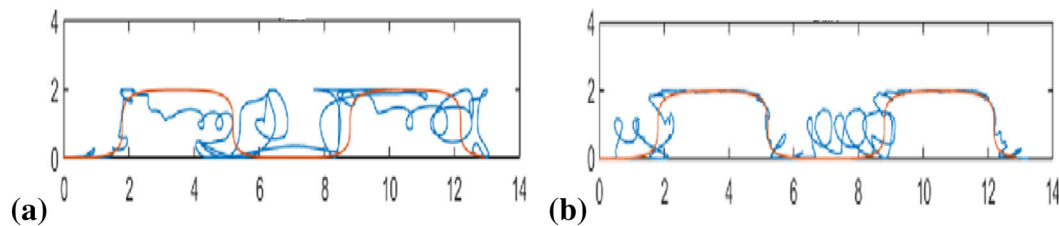


Fig. 38 VSSO results **a** T1FLC and **b** IT2FLC with 0.2 white noise power

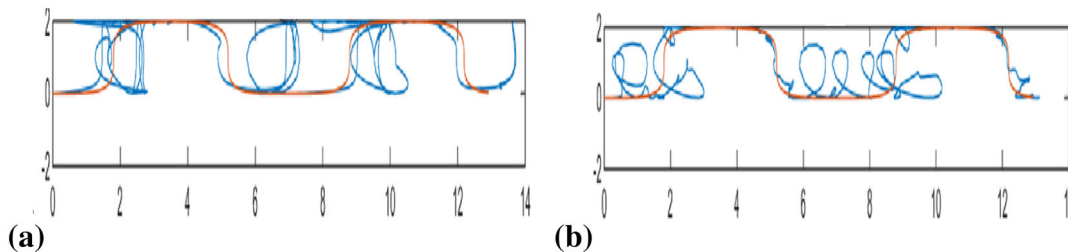


Fig. 39 WDO Results **a** T1FLC and **b** IT2FLC with 2% white noise power

Table 16 Time response comparison

Controller type	Method type			
	VSSO		WDO	
	T1FLC	IT2FLC	T1FLC	IT2FLC
Time response (min.)	00:03:24	00:39:21	00:06:43	01:13:27

References

1. Yildiz, A.R.: Comparison of evolutionary-based optimization algorithms for structural design optimization. *Eng. Appl. Artif. Intell.* **26**(1), 327–333 (2013). <https://doi.org/10.1016/j.engappai.2012.05.014>
2. Vesterström, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753) <https://doi.org/10.1109/cec.2004.1331139> (2004)
3. Al-Jarrah, R., Shahzad, A., Roth, H.: Path planning and motion coordination for multi-robots system using probabilistic neuro-fuzzy. *IFAC-PapersOnLine* **28**(10), 46–51 (2015). <https://doi.org/10.1016/j.ifacol.2015.08.106>
4. Caraveo, C., Valdez, F., Castillo, O.: Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation. *Appl. Soft Comput.* **43**, 131–142 (2016). <https://doi.org/10.1016/j.asoc.2016.02.033>
5. Amador-Angulo, L., Mendoza, O., Sensors, J.C., undefined 2016.: Fuzzy sets in dynamic adaptation of parameters of a bee colony optimization for controlling the trajectory of an autonomous mobile robot. *mdpi.com*. Accessed 19 May 2019. <https://www.mdpi.com/1424-8220/16/9/1458>
6. Olivas, F., Valdez, F., Castillo, O.: Fuzzy classification system design using PSO with dynamic parameter adaptation through fuzzy logic. *Stud. Comput. Intell.* **574**, 29–47 (2015). https://doi.org/10.1007/978-3-319-10960-2_2
7. Olivas, F., Valdez, F., Castillo, O., Gonzalez, C.I., Martinez, G., Melin, P.: Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems. *Appl. Soft Comput. J.* **53**, 74–87 (2017). <https://doi.org/10.1016/j.asoc.2016.12.015>
8. Abedinia, O., Amjady, N., Ghasemi, A.: A new metaheuristic algorithm based on shark smell optimization. *Complexity* (2016). <https://doi.org/10.1002/cplx.21634>
9. Bayraktar, Z., Komurcu, M., Werner, D.H.: Wind Driven Optimization (WDO): a novel nature-inspired optimization algorithm and its application to electromagnetics. In: 2010 IEEE Int. Symp. Antennas Propag. CNC-USNC/URSI Radio Sci. Meet.—Lead. Wave, AP-S/URSI 2010, no. 1, pp. 0–3, 2010. <https://doi.org/10.1109/APS.2010.5562213> (2010)
10. Bernal, E., Castillo, O., Soria, J., Valdez, F.: Galactic swarm optimization with adaptation of parameters using fuzzy logic for the optimization of mathematical functions. *Stud. Comput. Intell.* **749**, 131–140 (2018)
11. Ochoa, P., Castillo, O., Soria, J.: Differential evolution with dynamic adaptation of parameters for the optimization of fuzzy controllers, pp. 275–288 (2014)
12. Perez, J., Valdez, F., Castillo, O., Roeva, O.: Bat algorithm with parameter adaptation using Interval Type-2 fuzzy logic for benchmark mathematical functions. In: 2016 IEEE 8th International Conference on Intelligent Systems, IS 2016—Proceedings, pp. 120–127, <https://doi.org/10.1109/IS.2016.7737409> (2016)
13. Pérez, J., Valdez, F., Castillo, O.: Modification of the bat algorithm using type-2 fuzzy logic for dynamical parameter adaptation. *Stud. Comput. Intell.* **667**, 343–355 (2017)
14. Castillo, O., Melin, P.: Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic. *Stud. Fuzziness Soft Comput.* **223**, 121–132 (2008). https://doi.org/10.1007/978-3-540-76284-3_10
15. Mamdani, E.H., Assilian, S.: An experimental in linguistic synthesis with a fuzzy logic controller. *Int. J. Man. Mach. Stud.* **7**(1), 1–13 (1975). <https://doi.org/10.1006/ijhc.1973.0303>

16. Mendel, J.M.: On KM algorithms for solving type-2 fuzzy set problems. *IEEE Trans. Fuzzy Syst.* (2013). <https://doi.org/10.1109/TFUZZ.2012.2227488>
17. Ochoa, P., Castillo, O., Soria, J.: Differential evolution algorithm with interval type-2 fuzzy logic for the optimization of the mutation parameter. *Stud. Comput. Intell.* **749**, 55–65 (2018)
18. Ochoa, P., Castillo, O., Soria, J.: Fuzzy differential evolution method with dynamic parameter adaptation using type-2 fuzzy logic. In: 2016 IEEE 8th International Conference on Intelligent Systems, IS 2016—Proceedings, pp. 113–118, <https://doi.org/10.1109/IS.2016.7737408> (2016)
19. Caraveo, C., Valdez, F., Castillo, O.: Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation. *Appl. Soft Comput. J.* **43**, 131–142 (2016). <https://doi.org/10.1016/j.asoc.2016.02.033>
20. Amador-Angulo, L., Mendoza, O., Castro, J.R., Rodríguez-Díaz, A., Melin, P., Castillo, O.: Fuzzy sets in dynamic adaptation of parameters of a bee colony optimization for controlling the trajectory of an autonomous mobile robot. *Sensors* **16**(9), 1458 (2016). <https://doi.org/10.3390/s16091458>
21. Zamani, A.A., Bijami, E., Sheikholeslam, F., Jafrasteh, B.: Optimal fuzzy load frequency controller with simultaneous auto-tuned membership functions and fuzzy control rules. *Turkish J. Electr. Eng. Comput. Sci.* (2014). <https://doi.org/10.3906/elk-1203-3>
22. Oveis, A., Amjady, N., Ali, G.: A new metaheuristic algorithm based on shark smell optimization. *Complexity* (2014). <https://doi.org/10.1002/cplx>
23. Ehteram, M., Karami, H., Mousavi, S.F., El-Shafie, A., Amini, Z.: Optimizing dam and reservoirs operation based model utilizing shark algorithm approach. *Knowl.-Based Syst.* **122**, 26–38 (2017). <https://doi.org/10.1016/j.knosys.2017.01.026>
24. Juma, S.A., Muriithi, C.M., Ngoo, L.M.: Optimal switching sequence using a metaheuristic algorithm for feeder reconfiguration. *Int. J. Eng. Res. Technol.* **11**(8), 1329–1346 (2018)
25. Bayraktar, Z., Komurcu, M., Werner, D.H.: Wind Driven Optimization (WDO): a novel nature-inspired optimization algorithm and its application to electromagnetics. <https://doi.org/10.1109/APS.2010.5562213> (2010)
26. Pandey, A., Parhi, D.R.: Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm. *Def. Technol.* **13**(1), 47–58 (2017). <https://doi.org/10.1016/j.dt.2017.01.001>
27. Zadeh, L.A.: Fuzzy sets. *Inf. Control* (1965). [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
28. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst.* (1978). [https://doi.org/10.1016/0165-0114\(78\)90029-5](https://doi.org/10.1016/0165-0114(78)90029-5)
29. Wu, D., Mendel, J.M.: Enhanced Karnik-Mendel algorithms. *IEEE Trans. Fuzzy Syst.* (2009). <https://doi.org/10.1109/TFUZZ.2008.924329>
30. Ibrahim, M.T., Hanafi, D., Ghoni, R.: Autonomous navigation for a dynamical hexapod robot using fuzzy logic controller. *Procedia Eng.* **38**, 330–341 (2012). <https://doi.org/10.1016/j.proeng.2012.06.042>
31. Garcia, M.A.P., Montiel, O., Castillo, O., Sepúlveda, R., Melin, P.: Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Appl. Soft Comput. J.* **9**(3), 1102–1110 (2009). <https://doi.org/10.1016/j.asoc.2009.02.014>
32. El-Ferik, S., TariqNasir, M., Baroudi, U.: A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space. *Appl. Soft Comput. J.* **44**, 117–127 (2016). <https://doi.org/10.1016/j.asoc.2016.03.018>
33. Eberhart, R.C., Shi, Y.: Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation. <https://doi.org/10.1109/cec.2001.934374> (2001)
34. Castillo, O., Melin, P., Pedrycz, W.: Design of interval type-2 fuzzy models through optimal granularity allocation. *Appl. Soft Comput. J.* **11**(8), 5590–5601 (2011). <https://doi.org/10.1016/j.asoc.2011.04.005>
35. Castillo, O., Melin, P.: Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic. *Stud. Fuzziness Soft Comput.* (2008). https://doi.org/10.1007/978-3-540-76284-3_10
36. Sunisith, S., Joseph, L., Saritha, M.: Comparison of Fuzzy PID controller with conventional PID controller in controlling the speed of a brushless DC motor. *Int. Electr. Eng. J.* **5**(12), 1665–1672 (2014)
37. Pérez, J., Valdez, F., Castillo, O.: A new bat algorithm with fuzzy logic for dynamical parameter adaptation and its applicability to fuzzy control design. *Stud. Comput. Intell.* **574**, 65–79 (2015). https://doi.org/10.1007/978-3-319-10960-2_4
38. Bidar, M., Kanan, H.R.: Modified firefly algorithm using fuzzy tuned parameters. In: 2013 13th Iranian Conference on Fuzzy Systems (IFSC). <https://doi.org/10.1109/IFSC.2013.6675634> (2013)
39. Ahmadigorji, M., Amjady, N.: A multiyear DG-incorporated framework for expansion planning of distribution networks using binary chaotic shark smell optimization algorithm. *Energy* (2016). <https://doi.org/10.1016/j.energy.2016.02.088>
40. Bagheri, M., Sultanbek, A., Abedinia, O., Naderi, M.S., Naderi, M.S., Ghadimi, N.: Multi-objective shark smell optimization for solving the reactive power dispatch problem. In: 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe). <https://doi.org/10.1109/EEEIC.2018.8494502> (2018)
41. Astudillo, L., Melin, P., Castillo, O.: Optimization of type-2 and type-1 fuzzy tracking controllers for an autonomous mobile robot under perturbed torques by means of a chemical optimization paradigm. *Stud. Fuzziness Soft Comput.* **294**, 3–26 (2013). <https://doi.org/10.1007/978-3-642-35323-9-1>
42. Caraveo, C., Valdez, F., Castillo, O.: A new meta-heuristics of optimization with dynamic adaptation of parameters using type-2 fuzzy logic for trajectory control of a mobile robot. *Algorithms* **10**(3), 85 (2017). <https://doi.org/10.3390/a10030085>
43. Brockett, R.W.: Asymptotic stability and feedback stabilization. *Differ. Geom. Control Theory* **27**(1), 181–191 (1983)
44. Eberhart, R., Kennedy, J.: New optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. <https://doi.org/10.1109/mhs.1995.494215> (1995)
45. Oltean, S.E., Dulau, M., Puskas, R.: Position control of Robotino mobile robot using fuzzy logic. In: 2010 IEEE Int. Conf. Autom. Qual. Testing, Robot, pp. 1–6. <https://doi.org/10.1109/AQTR.2010.5520855> (2010)
46. Cuevas, F., Castillo, O., Cortes-Antonio, P.: Towards an adaptive control Strategy based on type-2 fuzzy logic for autonomous mobile robots. In: IEEE International Conference on Fuzzy Systems, vol. 2019. <https://doi.org/10.1109/FUZZ-IEEE.2019.8858801> (2019)
47. Cuevas, F., Castillo, O., Cortes, P.: Towards a control strategy based on type-2 fuzzy logic for an autonomous mobile robot, pp. 301–314 (2020)
48. Ontiveros-Robles, E., Melin, P., Castillo, O.: Comparative analysis of noise robustness of type 2 fuzzy logic controllers. *Kybernetika* **54**(1), 175–201 (2018). <https://doi.org/10.14736/kyb-2018-1-0175>



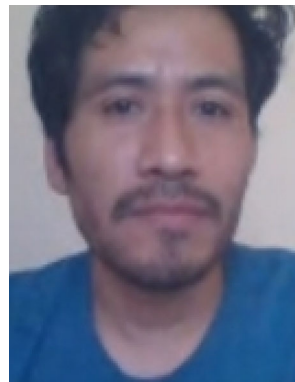
Felizardo Cuevas is a Ph.D. student in the Computer Science program of Tijuana Institute of Technology. He is working on a Ph.D. thesis in the area of type-2 fuzzy control of autonomous mobile robots. Currently, he has more than 10 papers published in fuzzy control, metaheuristics and robotics.



Oscar Castillo holds the Doctor in Science degree (Doctor Habilitatus) in Computer Science from the Polish Academy of Sciences (with the Dissertation “Soft Computing and Fractal Theory for Intelligent Manufacturing”). He is a Professor of Computer Science in the Graduate Division, Tijuana Institute of Technology, Tijuana, Mexico. In addition, he is serving as Research Director of Computer Science and head of the research group on Hybrid

Fuzzy Intelligent Systems. Currently, he is President of HAFSA (Hispanic American Fuzzy Systems Association) and Past President of IFSA (International Fuzzy Systems Association). Prof. Castillo is also Chair of the Mexican Chapter of the Computational Intelligence Society (IEEE). He also belongs to the Technical Committee on Fuzzy Systems of IEEE and to the Task Force on “Extensions to Type-1 Fuzzy Systems”. He is also a member of NAFIPS, IFSA and

IEEE. He belongs to the Mexican Research System (SNI Level 3). His research interests are in Type-2 Fuzzy Logic, Fuzzy Control, Neuro-Fuzzy and Genetic-Fuzzy hybrid approaches. He has published over 300 journal papers, 10 authored books, 50 edited books, 300 papers in conference proceedings, and more than 300 chapters in edited books, in total more than 970 publications with h index of 78 according to Google Scholar. He has been Guest Editor of several successful Special Issues in the past, like in the following journals: Applied Soft Computing, Intelligent Systems, Information Sciences, Soft Computing, Non-Linear Studies, Fuzzy Sets and Systems, JAMRIS and Engineering Letters. He is currently Associate Editor of the Information Sciences Journal, Journal of Engineering Applications on Artificial Intelligence, International Journal of Fuzzy Systems, Journal of Complex and Intelligent Systems, Granular Computing Journal and Intelligent Systems Journal (Wiley). He was Associate Editor of Journal of Applied Soft Computing and IEEE Transactions on Fuzzy Systems. He has been elected IFSA Fellow in 2015 and MICAIFellow in 2016. Finally, he recently received the Recognition as Highly Cited Researcher in 2017 and 2018 by Clarivate Analytics and Web of Science.



Prometeo Cortes holds the Ph.D. in Computer Science from National Polytechnic Institute in Mexico City. Currently, he is Professor in the Doctoral program in Computer Science from Tijuana Institute of Technology. His current interests are in fuzzy logic, fuzzy control, type-2 fuzzy systems and robotics. At the moment, he has published more than 30 papers, including papers in journals, chapters in edited books and conference

proceedings.