Optimal design of interval type-2 fuzzy tracking controllers of mobile robots using a metaheuristic algorithm

Felizardo Cuevas, *Oscar Castillo and Prometeo Cortes
Tijuana Institute of Technology, Tijuana BC México
*email: ocastillo@tectijuana.mx, Prometeo.cortes@tectijuana.edu.mx, felizardo.cuevas@tectijuana.edu.mx

***Abstract*— This paper describes a modification of a biological inspired algorithm based on shark behavior (SSO, shark smell optimization) for the optimization of the membership function's parameters for the fuzzy controllers of autonomous mobile robots. SSO is a metaheuristic technique based on the behavior presented by sharks in nature, which can be used for solving optimization problems. First, SSO is used to optimize benchmark control problems. Second, the traditional SSO is tested with the optimization of the membership function's parameters of type-1 fuzzy controllers. Third, tests are also performed with the Interval Type-2 Fuzzy Logic Controller. The comparison of results between the controller optimized with SSO and the controller optimized with WDO demonstrates that the proposed method shows better performance in the optimal design of fuzzy controllers.**

Keywords: Mobile Robot, T2FS (Type-2 Fuzzy Systems), T2FLC (Type-2 Fuzzy Logic Controller), T1FLC (Type-1 Fuzzy Logic Controller), SSO (Shark Smell Optimization), WDO (Wind Driven Optimization).

I. INTRODUCTION

The main goal of optimization is to find the best solution for a set of possible solutions to a particular problem. There are cases in which the problem solving space is too large and this may result in a time to find the solution becoming unaffordable. In other cases, there are different computational intelligence areas that provide a set of techniques to solve search and optimization problems [1], [2]. Such techniques can provide very competitive results, although not necessarily the best alternatives. Algorithms based on populations have become a new paradigm of collective intelligence and can find the best or near to best solutions to optimization problems with reasonable times and costs [3], [4].

Many scientists in the field of artificial intelligence have taken these algorithms as a research topic. Swarm intelligence, can be viewed as a set of metaheuristic techniques of artificial intelligence that are based on the analysis of systems with collective behavior [5] [6] [7]. These systems are present in nature, generally in a self-organized manner [8].

Many systems based on search and optimization algorithms have been applied to solve a wide range of problems.

In [9] an algorithm inspired by the movement of stars, galaxies, and superclusters of galaxies under the force of gravity is proposed. In this work, different fuzzy systems were designed for the dynamic adaptation of the c_3 and c_4 parameters to measure the performance of the algorithm with seven mathematical functions with a different number of dimensions. This method increases the efficiency of the algorithm by providing multiple cycles of exploration and exploitation, thus increasing the chances of accurately finding a global minimum.

In [10] a new method for dynamic parameter adaptation in particle swarm optimization (PSO) is proposed. PSO is a metaheuristic inspired on social behaviors, which is very useful in optimization problems. This paper proposes an improvement to the convergence and diversity of the swarm in PSO using fuzzy logic. Fuzzy rules were used to control the key parameters in PSO to achieve the best possible dynamic adaptation of these parameter values.

In [11] a modification to the bee colony optimization algorithm (BCO), with a fuzzy approach to dynamically change its parameters, was proposed. This is a metaheuristic technique inspired by the behavior presented by bees in nature, which can be used for solving optimization problems. First, the traditional BCO was tested with the optimization of fuzzy controllers. Second, a modification of the original method is presented by including fuzzy logic to dynamically change the main parameter values of the algorithm during execution. Third, the proposed modification of the BCO algorithm with the fuzzy approach is used to optimize benchmark control problems.

In [12] the BCO technique was used to find the optimal distribution of the membership functions in the design of fuzzy controllers. BCO is used specifically for tuning membership functions of the fuzzy controller for trajectory stability in an autonomous mobile robot. Two types of perturbations were added to the model for the Generalized Type-2 Fuzzy Logic System to better analyze its behavior under uncertainty and this shows better results when compared to the original BCO.

In [13]–[15] the Differential Evolution (DE) algorithm uses fuzzy logic for dynamic parameter adaptation of the mutation parameter (F). This modification of the algorithm is called Fuzzy Differential Evolution algorithm (FDE).

A comparison of the algorithm using type-1 fuzzy logic and interval type-2 fuzzy logic was performed for a set of Benchmark functions. Through evolutionary computing, a route planner was also designed by dynamically optimizing the membership functions of the fuzzy controller and motion controller.

In [16] an auto-tuned fuzzy load frequency controller (FLFC)-based artificial bee colony (ABC) algorithm was developed to smooth the deviations in the frequency and tie-line power due to load disturbances in an interconnected power system. Optimal tuning of membership functions (MFs) and fuzzy control rules is very important to improve the design performance and achieve a satisfactory level of robustness for a particular application. In this work, to reduce the fuzzy system design effort and consider large parametric uncertainties, a new systematic and simultaneous tuning method was developed for designing MFs and fuzzy rules. For this, the designing problem is restructured as an optimization problem and the ABC algorithm was applied to solve it.

The main contribution of this work is the implementation of a method to dynamically determine the optimal parameter values of a fuzzy controller using the shark smell optimization algorithm (SSO). There are previous works by the researchers that use the same SSO algorithm with the difference that in this paper, SSO is used for the dynamic adjustment of fuzzy controller parameters. In this case, the algorithm is only used to adjust the parameters of membership functions of a fuzzy controller, but not the rules. In this work, the method is applied to different benchmark control problems.

This paper is organized as follows. The basic concepts and operation of the SSO algorithm are described in Section II, the proposed methodology is explained in Section III, the case studies used in this work are described in Section IV, the experimental results with type-1 fuzzy logic, type-2 fuzzy logic and the statistical comparison between the two algorithms SSO (Shark Smell Optimization) and WDO (Wind Driven Optimization) respectively are described in Section V. Finally, in Section VI the conclusions are presented.

II. Shark smell optimization

This section describes the model proposed by Oveis Abedinia, Nima Amjady and Ali Ghasemi in 2004, based on the observation of intelligent behavior of sharks searching for food or prey [17][18][19]. Shark Smell optimization (SSO) is a meta-heuristic algorithm [17, 18] that belongs to the class of nature-inspired algorithms.

This method (SSO) uses a technique similar to the way the sharks look for food in nature and how they use their location and optimization methods to find optimal routes towards the source of food [18].

Natural systems have become currently an important source of ideas and models for the development of many artificial systems [19]. Various biological and natural processes have inspired these types of algorithms.

A. Behavior and structure of SSO

Natural systems have become currently an important source of ideas and models for the development of many artificial systems. The mathematical model of the shark search process consists of three main components shown below [17, 20]:

1) Initialization of food source (Population of solutions): The fish is injured and injects blood into the sea (search environment). Therefore, the speed of movement of the fish is low and negligible compared to the shark's velocity. Hence, the source of food (prey) is assumed to be approximately fixed.
2) Shark search (Determination of food location): The blood is injected into the sea and the effect of the water flows on distorting the odor particles. Thus, closer odor particles to the prey will be stronger. Consequently, by following the odor particles, the shark can hunt the prey.
3) Scout Sharks (exploration Phase): Evaluation of odor particles Information or quality of the solution in the constant search for a food source, a blood source, that is, an injured fish, in the shark search environment.

B. Behavior and structure of SSO

This method was proposed in [17]. The shark smell optimization algorithm (SSO) is based on the shark smelling abilities for localizing the source of food. In sharks' movement, the concentration of the smell is an important factor to guide the shark to the prey. In other words, the shark moves in the way with higher smell concentration. Fig. 1 presents the movement of shark to the odor source based on its concentration. This characteristic is used in the SSO algorithm to find the best solution to an optimization problem. In this algorithm, several assumptions are considered, which have been presented in [17]. The following steps briefly explain the algorithm (for a minimization problem):

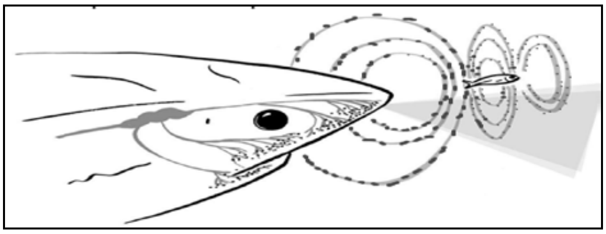

Fig. 1 Shark's movement to blood odor source and towards the prey.

a. Initialization

A population of initial solutions is randomly generated for an optimization problem in a feasible search area (sea). A source (prey) represents the optimal solution whilst the quality of the solution is represented by the smell strength at a particular position. According to [17],[20] the initial solution is given as follows:

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

where the i_{th} initial position vector X_i^k, i.e. i_{th} initial candidate solution for the optimization problem, is as follows:

$$X_{i,j}^1 = \left[X_{i,1}^1, X_{i,2}^1, X_{i,3}^1, \dots X_{i,ND}^1\right] \quad (2)$$
$$ND = Number\ of\ decision\ variables$$

where $X_{i,j}^k$ is the J^{th} dimension of the i^{th} shark position; ND is several decision variables in an optimization problem. Intensity of smell in each position indicates its closeness to the prey.

b. Forward Movement

As the blood disperses into the water, the shark will move towards the target with a velocity $'V'$, oriented by the smell of the stronger odor particles, hence leading to a high-quality solution. In correspondence with the position vector, each velocity vector has dimensional component elements:

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

By increasing the odor concentration, the velocity of the shark will increase. In each stage for magnitude of $V_{i\,J}^1$, is given as follows:

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

where

$\boldsymbol{\beta_k}$; *Is a velocity limiter ratio for stage k.*
$\boldsymbol{\eta_k}$; *is an element in* $[0\ 1]$.
$\propto_k$; *Is the inertia coefficient.*
$\boldsymbol{\nabla(OF)}$; *Is the gradient of the objective function.* $\frac{\partial(OF)}{\partial x_j}$

R_1 and R_2 are random values, which gives more randomness to the search when determining the velocity reached by the gradient function and to broaden the search of the algorithm. The rate of momentum $\propto_k$ becomes constant for stage k (number of stages for shark's forward movement) and the velocity is dependent from its former state.

The considered sign for the value of $v_{i\,j}^k$ depends on the direction of the selected term of the minimum operator.

The velocity vector will determine the new position during the forward movement of the shark given by:

$$\mathbf{Y_i^{k+1}} = x_i^k + v_i^{k-1} . \Delta t_k \quad (5)$$

where

$\boldsymbol{\Delta t_k}$; *Time interval is assumed to be* 1.
$\mathbf{Y_i^{k+1}}$; *Is new position.*
$\boldsymbol{x_i^k}$; *actual Position.*
$\boldsymbol{v_i^{k-1}}$; *Previous velocity.*

c. Rotational Movement

The rotational movement allows the shark to identify the stronger odor particles it when moves forward and this enables a local search within in the SSO algorithm. As can be noted from Fig. 2, the rotation of the shark is on a closed contour and not necessarily a circle. From optimization viewpoint, the shark implements a local search in each stage to find better candidate solutions. This is modelled by the equation below:

$$Z_i^{k+1,m} = Y_i^{k+1} + R_3 . Y_i^{k+1} \quad (6)$$
$$m = 1,2,\dots,M \quad i = 1,2,\dots,NP \quad k = 1,2,\dots,k_{max}$$

The rotational movement allows the shark to identify the stronger odor particles when it moves forward and this enables a local search with in the SSO algorithm.

R_3; *Random number with uniform distribution distribution in the range* $[1, -1]$
M; *Indicates number of points in the local search of each stage.*

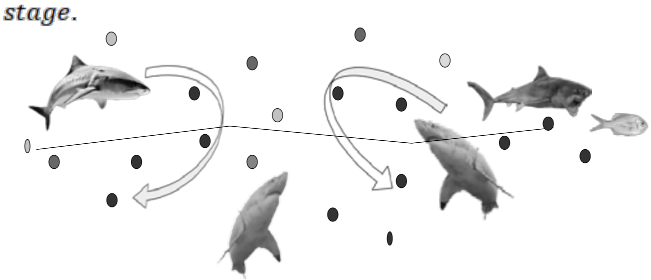

Fig. 2 Shark`s Rotational Movement.

d. Updating the Shark Position

If the shark finds a point with stronger smell during the rotational movement; it will follow this point and continue the search path from it as shown in Fig. 2. This characteristic is implemented in the SSO algorithm as follows:

$$\boldsymbol{X_i^{k+1}} = \boldsymbol{argmin}\{OF(Y_i^{k+1}), \boldsymbol{OF}(\boldsymbol{Z_i^{k+1,1}}), \dots, \boldsymbol{OF}(\boldsymbol{Z_i^{k+1,M}})\} \quad (7)$$
$$i = 1,2\dots,NP \quad k = 1,2,\dots,k_{max} \quad m = 1,2,\dots M$$

$\boldsymbol{X_i^{k+1}}$; *the next position of the shark.*
$\boldsymbol{Y_i^{k+1}}$; *Forward movement.*
$\boldsymbol{Z_i^{k+1,M}}$; *Rotational movement with range of random number generation.*

The objective function *OF* should be minimized. The cycle will continue until k reaches the minimum value (best individual) in the given population in a search space, which will be chosen for the optimization problem.

SSO has a number of defined parameters by us, like the other metaheuristic optimization methods, including population size NP and number of stages k_{max} as well $\propto$, β, and η of each stage. We have empirically seen that SSO works well with these values of $\eta = 0.75$, $\propto = 0.1$, and $\beta = 3$. However, these parameters can be fine-tuned for each optimization problem separately.

III. Dynamic Parameters Setting of the Controller with SSO Algorithm

This section is focused on describing the dynamic adjustment of the parameters of a fuzzy controller. Methodologies similar to this proposal are described in [21][22][23]. The main goal in choosing the SSO algorithm is because there are few research works published of this algorithm, there also exist other variants [24][25] and we consider a good idea to use it and analyze if good results can be achieved when compared with respect to other algorithms.

A. Dynamic adaptation of Fuzzy controller parameters with proposed methodology

For the development of the proposed method, we establish the following sequence of steps in the SSO algorithm:
1. 40 dimensions are needed to establish the position of the points to optimize in the MFs.
2. We set a lower and upper limit of -1 to 1 normalizing range in the membership functions in fuzzy input and output sets.
3. The objective function is based on the controller's fuzzy inference system.

The methodology to optimize the parameters of the membership functions with the shark smell algorithm, evaluating the fuzzy controller in the plant and the results obtained in the evaluation with respect to the desired path, is illustrated in Fig. 3.

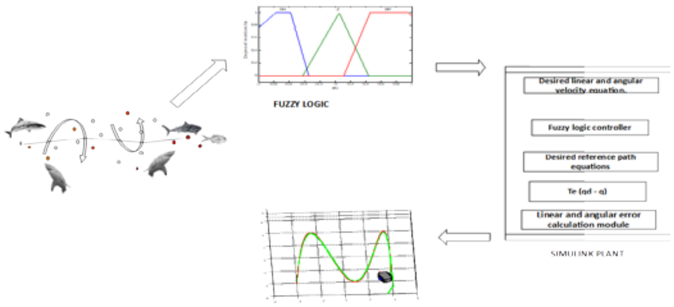

Fig. 3 Procedure for optimization of the parameters of the membership functions.

In Fig. 4, we graphically show the sequence of steps of the proposed algorithm.
The pseudocode of the proposed algorithm is presented below:
Step 1: Randomly generate the initial population of n blood particles for the MF parameters. To start, a set of random source odor particles is generated, where each row (vector) represents a blood particle. The particle has in memory a position, and that position is a possible solution to the problem, which in this case represents the values of the membership function parameters of the fuzzy controller.
The initial population must contain possible position candidate solutions that satisfy the constraints. Set k = 0, and evaluate the fitness value of the initial populations by (6).
Step 2: Select the best position for the neighborhood search. The selected odor particle memory contains the best position found so far and to define the fitness of each shark take position the equation of the mean square error is used, which is Eq. (7).
Step 3: Evaluate the fitness of each particle and identify each of the points of MFs using SSO.
Step 4: Represent the new value of MF from each shark position.
Step 5: Select the fittest positions of each shark.
Step 6: Move dynamically each of the MF points of the Inputs-Outputs.
Step 7: Check if the new fitness Function is better than the previous Fitness Function. If satisfied, update final position of each of the FIS MF points.
Step 8: Check the stopping criteria. If satisfied, terminate the search, else K = K + 1 up K_{max}, show and save the best values found and return to evaluate new solutions.

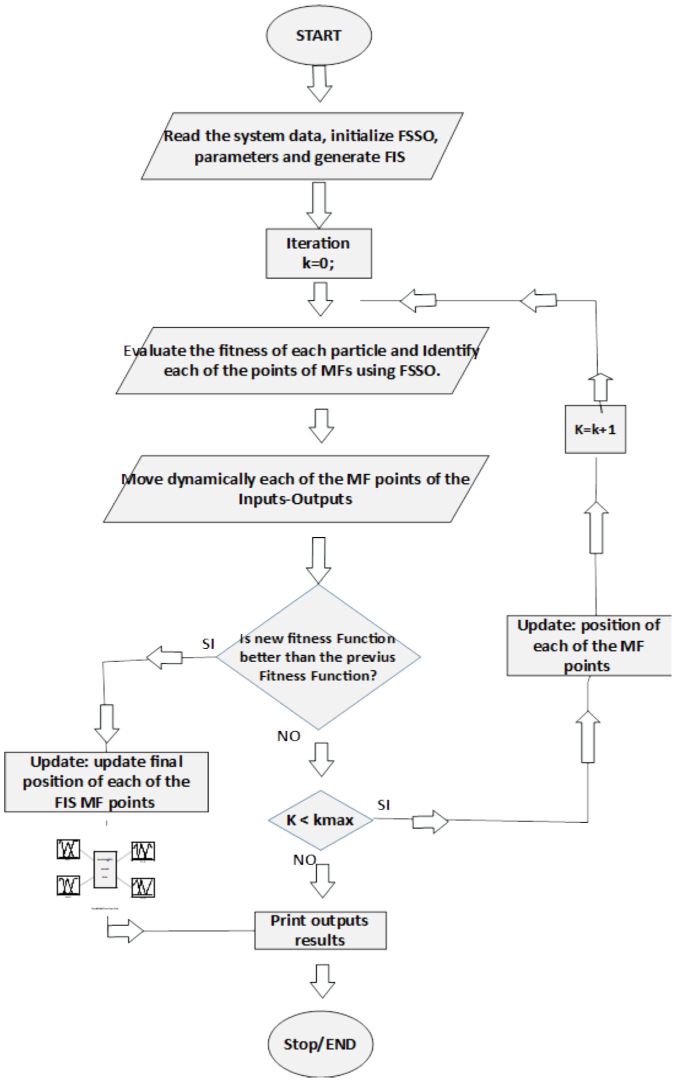

Fig. 4 Procedure to adjust fuzzy controller parameters using SSO.

We propose the following fuzzy control architecture to use dynamic variation of parameters at runtime, as is shown in Fig. 4.
The fuzzy logic controller design is of the Mamdani type and has 2 inputs and 2 outputs where the parameters of the Membership Functions are dynamically adjusted with 9 rules that are chosen with prior knowledge of the problem within the exploitation and exploration in search spaces.
To better explain the difference between the used method, with respect to one of the most used metaheuristics: particle swarm optimization (PSO) which was developed by Kennedy and Eberhart in [26][27], PSO has biological inspiration in nature, to be more specific in the behavior of birds, where each bird represents a particle. A comparison is made among PSO, WDO and SSO illustrated in Table I. Considering that the SSO algorithm is also based on PSO, the following comparison table is made, and a Clarification is made that PSO is not

currently used in this methodology, but rather only used in this table for comparison purposes.

TABLE I
COMPARISON OF METHOD CHARACTERISTICS. PSO: PARTICLE SWARM OPTIMIZATION; WDO: WIND-DRIVEN OPTIMIZATION; SSO: SHARK SMELL OPTIMIZATION

Characteristic	**PSO**	**WDO**	**SSO**
Population	Particle	Air package	Shark Smell
New speed	v_{k+1}^{i}	u_{t+1}^{i}	v_{ij}^{k}
Current speed	v_k^i	u_t^i	v_i^k
Actual position	x_i	x_t^i	x_i^k
Next position	x_{k+1}^{i}	x_{t+1}^{i}	x_{i+1}^{k}
Better experience	p_k^i	-	Y_i^{k+1}
Best group experience	p_k^g	-	$z_i^{k+1,M}$
Increase	k	T	k
Uniform random numbers between 0 and 1	r_1 , r_2	-	r_1, r_2, r_3
Cognitive parameter	c_1	-	-
Social parameter	c_2	-	-

B. *FUZZY LOGIC*

The term fuzzy logic was introduced with the 1965 proposal of fuzzy set theory by Lotfi Zadeh [28]. Fuzzy logic had however been studied since the 1920s, as multiple-valued logic—notably by Łukasiewicz and Tarski [29]. The use of linguistic variables can model human ways of thinking, and it is much easier to program systems based on the logic of human reasoning. By combining fuzzy logic with other artificial intelligence techniques, this mixture of techniques is able to achieve something that previously seemed impossible: the autonomy of mobile robots. So today, robots currently help with household chores, and even medical areas, in all areas of the industry, robots play a very important role in manufacturing processes. The good performance and efficiency in the work environment, creates the need for software and hardware to be updated and optimized regularly, as discussed in [30], [31].

We propose a fuzzy logic controller optimization; this fuzzy controller is of Mamdani type and is applied to create a smooth response in a mobile robotic platform, instead of a response with oscillations that could be produced with traditional hard logic. Other metaheuristics and combinations among them have also been used hybridizing with fuzzy logic in current robotics as in [32], [33][34]. Also Castillo and Melin et al. [5][35][36] describe the applications of the method.

Therefore, for designing the fuzzy system of Mamdani type, with dynamical adjustment of parameters of fuzzy controller, the two measures described as linear velocity error e_{lv} and angular velocity error e_{wv} were considered as inputs. The system has the fuzzy outputs; ParMotor1 and ParMotor2.

In regards to the inputs of the fuzzy system, the error variables have by themselves a defined range of possible values which range from -50 to 50, but with the linear velocity error e_{lv} and angular velocity error e_{wv}, we perform a normalization of the values of these to have values between -1 and 1. Eq. 8 shows how the normalization of the linear velocity error is performed and Eq. 9 shows how the normalization of the angular velocity error is obtained.

$$e_{lv} = \frac{\text{Vel. } Ref. - \text{Vel. Actual}}{50} \tag{8}$$

$$e_{wv} = \frac{w\ Ref. - w\ Actual}{50} \tag{9}$$

The design of the input variables can be appreciated in Fig. 5 and 6, which show the inputs linear velocity error e_{lv} , and angular velocity error e_{wv} respectively, where each input is granulated into three membership functions, trapezoidal at the ends and triangular at the middle. For the output variables, as mentioned above, we perform a normalization of the values of these to have values between -1 and 1, so that the output variables were designed using this range of values. Each output is granulated into three triangular membership functions, the design of the output variables can be seen in Fig. 7 and 8, PM_1 and PM_2 respectively. The fuzzy system has the linear and angular velocity errors as inputs, as shown in Fig. 9.

To design the rules of each fuzzy system, it was decided that in

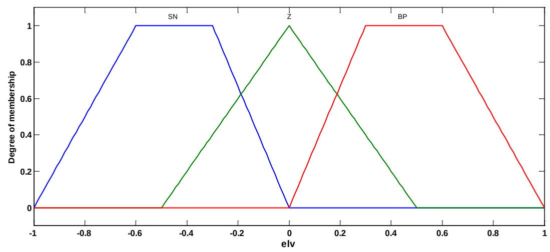

Fig. 5 Input 1: Linear velocity error e_{lv} .

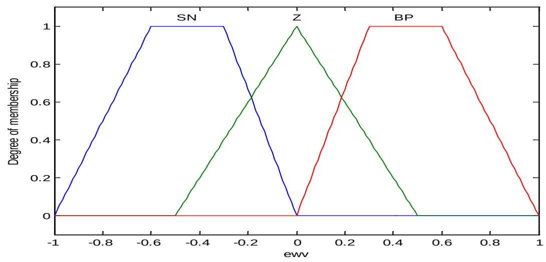

Fig. 6 Input 2: Angular velocity error e_{wv}.

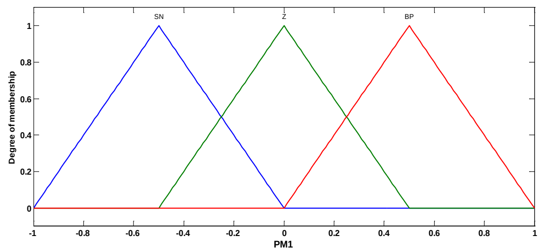

Fig. 7 Output 1: Parmotor 1 PM_1.

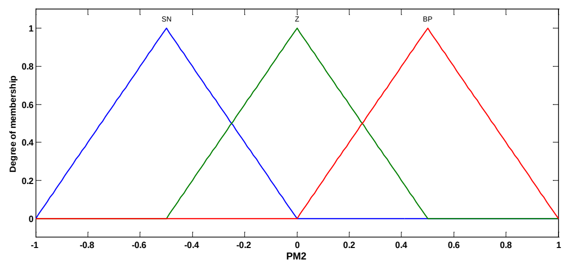

Fig. 8 Output 2: Parmotor 2 PM_2.

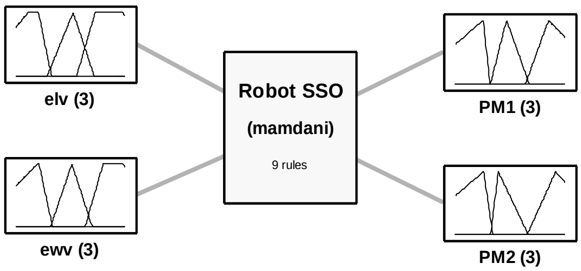

Fig. 9 Fuzzy system for robot controller.

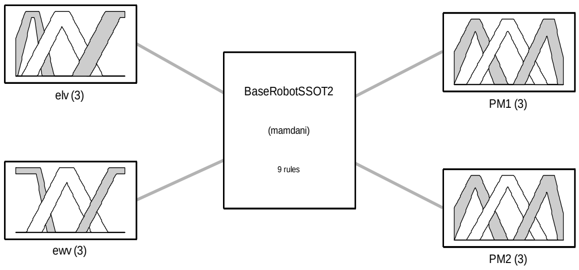

Fig. 10 It2flc Architecture.

early iterations the SSO algorithm must explore and eventually exploit. Taking into account other variables such as linear velocity error and angular velocity error for example, when e_{lv} is small negative (SN) and e_{wv} is zero (Z), that is, that the smell particles are close together, we must use exploitation, and when e_{lv} is big positive (BP) and e_{wv} is big positive (BP) we must use exploration, this in search of the candidate solutions. The fuzzy controller has 9 rules, which are shown below in Table II [37]:

TABLE II
FUZZY SYSTEM ROBOT CONTROLLER: IF-THEN FUZZY RULES

Rule	ANTECEDENT	CONSEQUENT
1.	If (elv is SN) and (ewv is SN)	then (PM1 is SN)(PM2 is SN) (1)
2.	If (elv is SN) and (ewv is Z)	then (PM1 is SN)(PM2 is Z) (1)
3.	If (elv is SN) and (ewv is BP)	then (PM1 is SN)(PM2 is BP) (1)
4.	If (elv is Z) and (ewv is SN)	then (PM1 is Z)(PM2 is SN) (1)
5.	If (elv is Z) and (ewv is Z)	then (PM1 is Z)(PM2 is Z) (1)
6.	If (elv is Z) and (ewv is BP)	then (PM1 is Z)(PM2 is BP) (1)
7.	If (elv is BP) and (ewv is SN)	then (PM1 is BP)(PM2 is SN) (1)
8.	If (elv is BP) and (ewv is Z)	then (PM1 is BP)(PM2 is Z) (1)
9.	If (elv is SN) and (ewv is SN)	then (PM1 is SN)(PM2 is SN) (1)

C. INTERVAL TYPE-2 FUZZY LOGIC

The concept of T1FS and T2FS fuzzy logic systems was developed by Zadeh between 1965 and 1975, approximately [38], [39]. This section presents a concise overview of T2FLS with the intention of providing basic knowledge of how IT2FLS works to achieve its objective.

First, we break down the overall behavior of the robot that we call "behavior", which may have the task of following a line. In Fig. 10, a controller is designed to determine the control action of the mobile robot. These basic behaviors have as control structures a controller with type-2 fuzzy logic formed from a set of fuzzy IF-THEN rules [40], [41]. The actions are monitored by the controller's rule set to select the appropriate actions that are transmitted to the system.

We know that the T2FS is located in a region built by a main type-1 membership function (T1MF). T2FS is obtained by using fuzzy sets to partition the input domains of the base line T1FS with a footprint of uncertainty (FOU) as shown in Fig. 11 to Fig. 14. Consequently, the T1MF is extended to T2MF by adding FOU to represent uncertainty. The variability in each of the actions is modeled with interval type-2 fuzzy sets (IT2FS). The T2FS linguistic input variables and their ranges are used for path tracking, as shown in Fig. 11 and 12, with two outputs, which are the Parmotor1 (PM_1) and Parmotor2 (PM_2), as shown in Fig. 13 and 14.

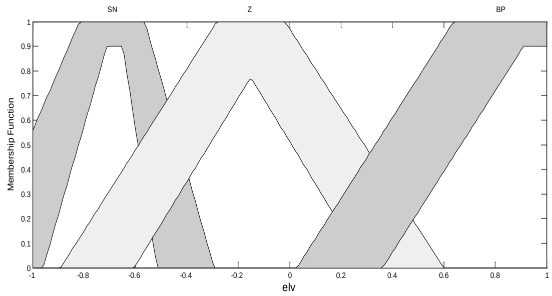

Fig. 11 Footprints of IT2 MFs for input e_{lv}

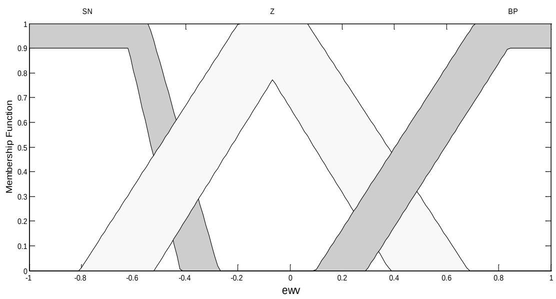

Fig. 12 Footprints of IT2 MFs for input e_{wv}

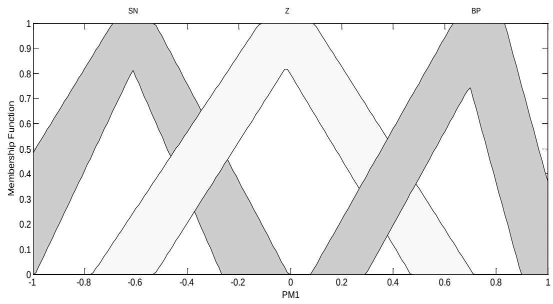

Fig. 13 Footprints of IT2 MFs for output Parmotor1PM_1 .

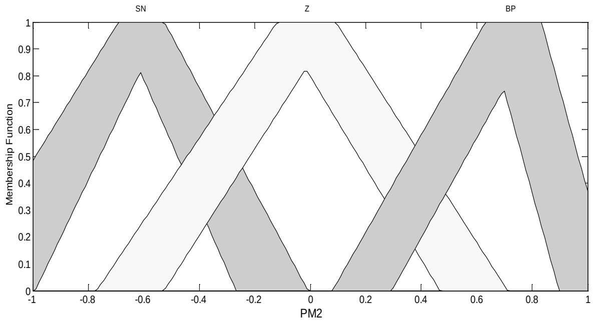

Fig. 14 Footprints of IT2 MFs for output Parmotor2 $\boldsymbol{PM_2}$.

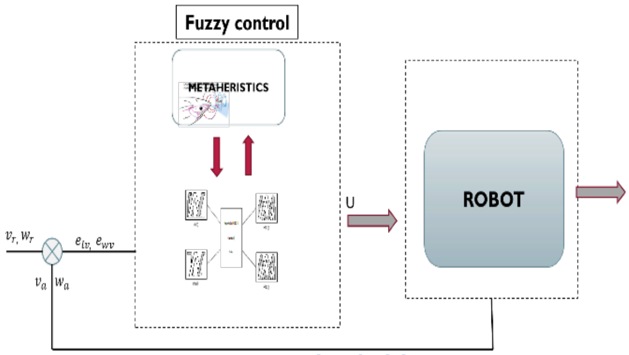

Fig. 16 proposed methodology.

Therefore, membership functions have values distributed according to the uncertainty. As well as those that belong to the antecedents also in the consequent parts. From the design, three MFs, the $\boldsymbol{e_{lv}}$ input (***SN***: *Small Negative*, ***Z***: *Zero*, ***BP***: *Big Positive*), three MFs for the $\boldsymbol{e_{wv}}$ input (***SN***: *Small Negative*, ***Z***: *Zero*, ***BP***: *Big Positive*), three MFs for the $\boldsymbol{PM_1}$ output (***SN***: *Small Negative*, ***Z***: *Zero*, ***BP***: *Big Positive*) and three MFs, the $\boldsymbol{PM_2}$ output (***SN***: *Negative Small*, ***Z***: *Zero*, ***BP***: *Positive Big*), are built.

In this work, nine rules were developed for the T2FLS path tracking. The number of rules (9) is determined by expert knowledge with T1FLS. Fuzzy rules are obtained from the combination of two inputs with three membership functions. The rules are used to control the parmotor of wheels right and left, some of these rules are as presented in Table II.

IV. Study Cases

A. Case 1: Benchmark Functions

Also for the comparison of the proposed method with respect to other methods, we considered benchmark mathematical functions, defined in [42], which are a total of 27 functions and and in each one it is tried to find values that give us the global minimum of each function. In Fig. 15 there is a sample of the functions that are used.

The problem that is considered (Benchmark mathematical functions) [21] is illustrated in Fig. 15, and the proposed methodology is as shown in Fig. 16, where we can notice that fuzzy parameters are adjusted by SSO, and in turn this “SSO” searches for the optimal parameters for the membership functions of the fuzzy input-output sets.

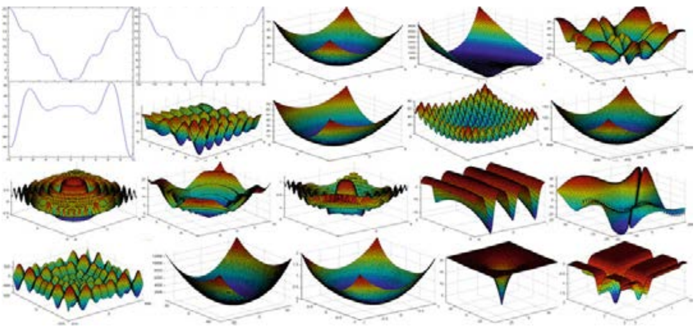

Fig. 15 Benchmark mathematical functions.

B. Case 1: Fuzzy Controller of a Mobile Robot

We propose the optimization for a fuzzy controller of an Autonomous Mobile Robot of differential type, using the proposed SSO for this specific problem; below, the fuzzy controller is explained in more detail.

The controller for the autonomous mobile robot is as used in [40],[43]. The main objective of the controller is the better search of a desired trajectory, and it has 9 rules that define the relationships between the linguistic variables of the fuzzy system. The linguistic variables are as follows: in the first input, it is the linear velocity error ($\boldsymbol{e_{lv}}$); the second input, it is the angular velocity error ($\boldsymbol{e_{wv}}$). The two inputs have the same linguistic values: big positive (BP), zero (Z), and small negative (SN), with the same membership functions in both fuzzy input sets. The outputs are Parmotor 1 ($\boldsymbol{PM_1}$) and Parmotor 2 ($\boldsymbol{PM_2}$) they represent the change of direction when the robot's wheels interact, and each of these outputs has three triangular functions. Fig. 17 represents the desired trajectory that the robot should follow. In this figure, the Y-axis shows the desired Y displacement, and the X-axis represents the X displacement. Fig. 18 describes the red line as the desired trajectory, and the real trajectory of the robot in blue, and as we can see, the robot is not lost, since it generates an error of 4.64 x 10^{-03} and a standard deviation 1.03 x 10^{-02}.

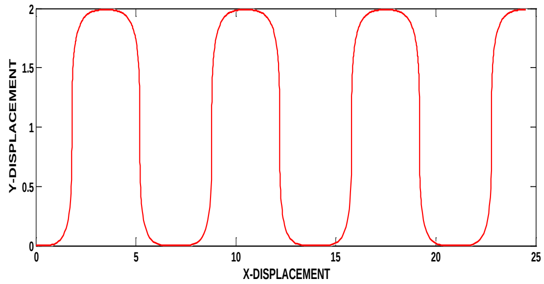

Fig. 17 Desired trajectory normalized axis

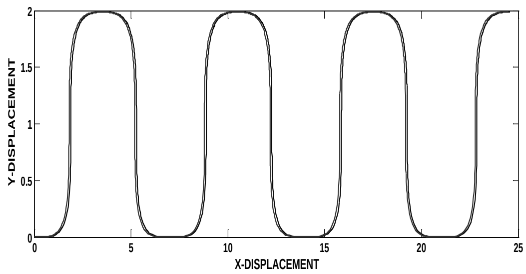

Fig. 18 Obtained mean square error (MSE).

V. SIMULATION RESULTS

In the case of the results obtained with the optimization of benchmark functions using the shark smell method, this is presented in the first part of this section. The results with the optimization of the membership functions of the fuzzy controller used with the autonomous mobile robot platform are presented in the second part of the section. This optimization generates a vector of values that are defined as the dynamic parameters of the optimized membership functions.

A. *Case 1 Results: Benchmark Functions*

Table III shows the parameters used in the optimization of Benchmark functions. Table IV column 1 shows the used functions; column 2 represents the minimum of the functions; column 3 shows the obtained average; column 4 represent the standard deviations respectively obtained with the SSO; To conclude columns 5 and 6 show the results of the SSO method. This was evaluated 30 times for each of these functions with the same parameters to obtain the averages and standard deviations. Table IV above shows the optimization results of benchmark functions where the minimum of functions f_{15}, f_{16}, f_{17}, and f_{18} is different from zero. With the evaluation with different minima, the performance of the optimization in the function with the SSO can be better observed.

TABLE III
PARAMETERS USED IN THE EXPERIMENTS

Population	Iterations	Dimensions
100	500	30

TABLE IV
RESULTS FOR 40 DIMENSIONS

		W D O		S S O	
Funct	f_{min}	Average	Standard Deviation	Average	Standard Deviation
f_1	0	10^{-27}	10^{-27}	10^{-13}	6.97 x 10^{-14}
f_2	0	10^{-14}	10^{-14}	10^{-06}	3.05 x 10^{-07}
f_3	0	10^{-20}	10^{-20}	10^{-13}	3.12 x 10^{-14}
f_4	0	10^{-13}	10^{-13}	10^{-13}	4.17 x 10^{-14}
f_5	0	28.569	10^{-02}	10^{-13}	4.70 x 10^{-16}
f_6	0	10^{-02}	10^{-03}	10^{-05}	3.12 x 10^{-06}
f_7	0	10^{-02}	10^{-02}	10^{-12}	9.44 x 10^{-15}
f_9	0	-118.27	0	10^{-13}	1.05 x 10^{-14}
f_{10}	0	9.53 x 10^{-15}	1.23 x 10^{-14}	10^{-08}	3.65 x 10^{-09}
f_{11}	0	0	0	10^{-13}	3.36 x 10^{-29}
f_{15}	0.000	3.07 x 10^{-04}	2.18 x 10^{-07}	10^{-06}	4.70 x 10^{-07}
f_{16}	-1.031	1.0316	6.78 x 10^{-16}	1.07	3.44 x 10^{-16}
f_{17}	0.398	0.3979	6.78 x 10^{-05}	6.0 x 10^{-12}	3.73 x 10^{-15}
f_{18}	3	7.783	3.61 x 10^{-15}	10^{-01}	3.80 x 10^{-02}

B. *Case 2 Results: Dynamic Adjustment of Fuzzy Controller Parameters*

Table V shows the parameters used in the WDO, and SSO methods, Table VI and Table VII shows some the 30 experiments performed to obtain the best optimized fuzzy system, using as metric the MSE, where it can be observed that the best error found is of 0.00169, and in general, the values only varied from 10^{-02} to 10^{-03}.

In Fig. 20 to Fig. 21, the dynamic adjustment that was made in the parameters of the membership functions of the inputs of the fuzzy controller is observed, where the uncertainty that is generated between each of them can be observed, and with this, the error that shows the fuzzy controller in the simulation is considerably improved. Fig. 22 and Fig. 23 show the outputs of the controller optimized by the SSO method, respectively.

Fig. 19–23 show the optimized movement of the parameters of the membership functions of the inputs e_{vl}, e_{wl} and the outputs PM_1 and PM_2, respectively. In addition, as can be noted from Fig. 20 to Fig.23, the overlap between existing functions helps the robot to have a better tracking of the desired trajectory, in comparison with the other method used in this work.

TABLE V
PARAMETERS USED WDO AND SSO METHOD FOR FUZZY CONTROLLER

Population	Iterations	Dimensions
30	1500	30

TABLE VI
RESULTS OF THE EXPERIMENTS WITH THE FUZZY CONTROLLER OBTAINED WITH SSO

Experiment	MSE	Experiment	MSE
1	1.26 X 10^{-06}	16	6.35 X 10^{-04}
2	1.39 X 10^{-05}	17	1.30 X 10^{-03}
3	6.76 X 10^{-04}	18	4.70 X 10^{-03}
4	1.13 X 10^{-05}	19	8.40 X 10^{-03}
5	3.72 X 10^{-05}	20	4.70 X 10^{-03}
6	2.41 X 10^{-05}	21	1.47 X 10^{-02}
7	2.99 X 10^{-06}	22	5.15 X 10^{-02}
8	5.01 X 10^{-05}	23	7.00 X 10^{-03}
9	2.95 X 10^{-05}	24	3.70 X 10^{-03}
10	5.69 X 10^{-06}	25	5.90 X 10^{-03}
11	6.55 X 10^{-06}	26	2.70 X 10^{-02}
12	4.08 X 10^{-05}	27	8.25 X 10^{-04}
13	1.84 X 10^{-06}	28	6.50 X 10^{-04}
14	1.15 X 10^{-05}	29	2.80 X 10^{-03}
15	3.67 X 10^{-06}	30	5.44 X 10^{-04}

It can be noted in Tables VI and VII, that there is a variation in the results obtained with the SSO method, since it has the best MSE of 0.00000126, but the worst is 0.05150. Unlike WDO where there is a big variation in the results; the best MSE with

0.000019, the worst of 0.057094. For this reason, on average, this method can be the best at optimizing this specific problem. Table VIII summarized the results of the optimization methods.

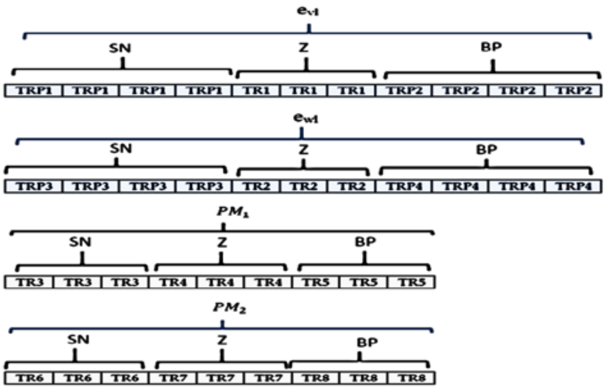

Fig. 19 Particles vector for optimizing the membership functions of the fuzzy controller.

TABLE VII
RESULTS OF THE EXPERIMENTS WITH THE FUZZY CONTROLLER. OBTAINED WITH WDO

Experiment	MSE	Experiment	MSE
1	**$1.90 \text{ X } 10^{-05}$**	16	$1.64 \text{ X } 10^{-04}$
2	$1.90 \text{ X } 10^{-05}$	17	$1.64 \text{ X } 10^{-04}$
3	$5.40 \text{ X } 10^{-05}$	18	$2.60 \text{ X } 10^{-04}$
4	$5.40 \text{ X } 10^{-05}$	19	$2.93 \text{ X } 10^{-04}$
5	$6.20 \text{ X } 10^{-05}$	20	$4.33 \text{ X } 10^{-04}$
6	$6.20 \text{ X } 10^{-05}$	21	$1.62 \text{ X } 10^{-03}$
7	$6.90 \text{ X } 10^{-05}$	22	$1.65 \text{ X } 10^{-03}$
8	$8.70 \text{ X } 10^{-05}$	23	$6.21 \text{ X } 10^{-03}$
9	$8.70 \text{ X } 10^{-05}$	24	$6.35 \text{ X } 10^{-03}$
10	$9.10 \text{ X } 10^{-05}$	25	$7.73 \text{ X } 10^{-03}$
11	$9.80 \text{ X } 10^{-05}$	26	$1.67 \text{ X } 10^{-02}$
12	$1.05 \text{ X } 10^{-04}$	27	$1.67 \text{ X } 10^{-02}$
13	$1.15 \text{ X } 10^{-04}$	28	$2.65 \text{ X } 10^{-02}$
14	$1.38 \text{ X } 10^{-04}$	29	$2.92 \text{ X } 10^{-02}$
15	$1.60 \text{ X } 10^{-04}$	30	$5.7 \text{ X } 10^{-02}$

TABLE VIII
RESULTS OF THE OPTIMIZATION METHODS

	WDO	SSO
Average	$6.00 \text{ X } 10^{-03}$	$4.6 \text{ X } 10^{-03}$
Standard Deviation	$1.26 \text{ X } 10^{-02}$	$1.10 \text{ X } 10^{-02}$

VI. Conclusions

The proposed methodology was created to optimize problems, and was tested with the unimodal and multimodal benchmark functions and comparing to the SSO and WDO methods. This was done with 30 experiments as the limitation; each of them was put into competition with the same parameters for a fair competition. It was obtained that SSO was better for benchmark functions. On the other hand, for the optimization of the parameters of the membership functions, the SSO method was the metaheuristic that found the data vector that managed to optimize the functions of the fuzzy controller in

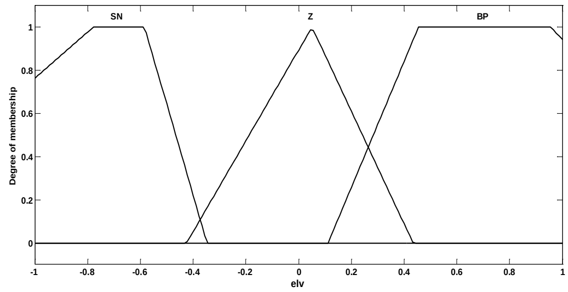

Fig. 20 Optimized input 1: linear speed error e_{lv} .

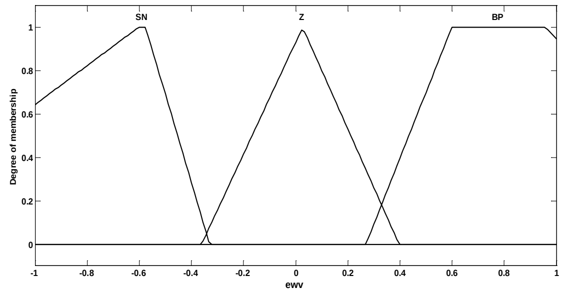

Fig. 21 Optimized Input 2: Angular velocity error e_{wl} .

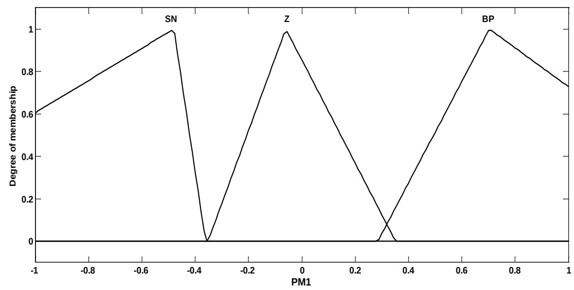

Fig. 22 Optimized Output 1: Parmotor 1 PM_1 .

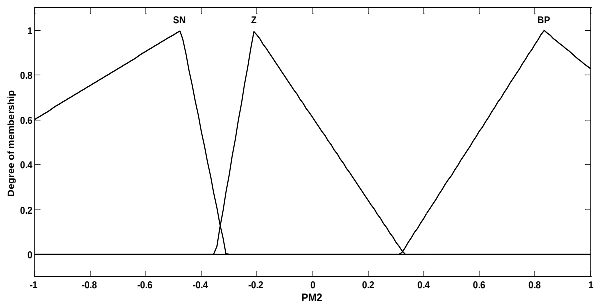

Fig. 23 Optimized Output 2: Parmotor 2 PM_2.

such a way that the robot approached the desired trajectory. As future work, we plan to perform more experimentation with other optimization problems. In addition, 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 parmotor1 and parmotor2 are changing when analyzing the results with the integral parameters applied and the times of path. The proposed approach implements a type-1 and type-2 fuzzy controller that produces the parmotor forces in the wheels for path tracking, with the ability for path tracking that arise in the environment so that the robot can travel the path without losing it and with the least possible error.

REFERENCES

[1] A. R. Yildiz, "Comparison of evolutionary-based optimization algorithms for structural design optimization," *Eng. Appl. Artif. Intell.*, vol. 26, no. 1, pp. 327–333, 2013.

[2] J. Vesterstrm and R. Thomsen, "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, CEC2004*, 2004.

[3] J. Perez, F. Valdez, O. Castillo, and O. Roeva, "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*, 2016, pp. 120–127.

[4] J. Pérez, F. Valdez, and O. Castillo, "Modification of the bat algorithm using type-2 fuzzy logic for dynamical parameter adaptation," in *Studies in Computational Intelligence*, vol. 667, 2017, pp. 343–355.

[5] O. Castillo and P. Melin, "Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic," *Stud. Fuzziness Soft Comput.*, vol. 223, pp. 121–132, 2008.

[6] E. H. Mamdani and S. Assilian, "An experimental in linguistic synthesis with a fuzzy logic controller," *Int. J. Man. Mach. Stud.*, vol. 7, no. 1, pp. 1–13, 1975.

[7] N. Karnik, J. M.-C. on F. S., and undefined 1998, "Introduction to type-2 fuzzy logic systems," *ieeexplore.ieee.org*.

[8] F. Olivas, F. Valdez, O. Castillo, C. I. Gonzalez, G. Martinez, and P. Melin, "Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems," *Appl. Soft Comput. J.*, vol. 53, no. C, pp. 74–87, Apr. 2017.

[9] E. Bernal, O. Castillo, J. Soria, and F. Valdez, "Galactic swarm optimization with adaptation of parameters using fuzzy logic for the optimization of mathematical functions," in *Studies in Computational Intelligence*, vol. 749, 2018, pp. 131–140.

[10] F. Olivas, F. Valdez, and O. Castillo, "Fuzzy classification system design using PSO with dynamic parameter adaptation through fuzzy logic," *Stud. Comput. Intell.*, vol. 574, pp. 29–47, 2015.

[11] C. Caraveo, F. Valdez, and O. Castillo, "Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation," *Appl. Soft Comput. J.*, vol. 43, pp. 131–142, Jun. 2016.

[12] L. Amador-Angulo, O. Mendoza, J. R. Castro, A. Rodríguez-Díaz, P. Melin, and O. Castillo, "Fuzzy sets in dynamic adaptation of parameters of a bee colony optimization for controlling the trajectory of an autonomous mobile robot," *Sensors (Switzerland)*, vol. 16, no. 9, p. 1458, Sep. 2016.

[13] P. Ochoa, O. Castillo, and J. Soria, "Differential Evolution with Dynamic Adaptation of Parameters for the Optimization of Fuzzy Controllers," 2014, pp. 275–288.

[14] P. Ochoa, O. Castillo, and J. Soria, "Differential evolution algorithm with interval type-2 fuzzy logic for the optimization of the mutation parameter," in *Studies in Computational Intelligence*, vol. 749, 2018, pp. 55–65.

[15] P. Ochoa, O. Castillo, and J. Soria, "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*, 2016, pp. 113–118.

[16] A. A. Zamani, E. Bijami, F. Sheikholeslam, and B. Jafrasteh, "Optimal fuzzy load frequency controller with simultaneous auto-tuned membership functions and fuzzy control rules," *Turkish J. Electr. Eng. Comput. Sci.*, 2014.

[17] G. Oveis Abedinia Nima Amjady Ali, "A new metaheuristic algorithm based on shark smell optimization," *Complexity*, vol. 21, no. 5, pp. 97–116, 2014.

[18] O. Abedinia, N. Amjady, and A. Ghasemi, "A new metaheuristic algorithm based on shark smell optimization," *Complexity*, 2016.

[19] N. Gnanasekaran, S. Chandramohan, P. S. Kumar, and A. Mohamed Imran, "Optimal placement of capacitors in radial distribution system using shark smell optimization algorithm," *Ain Shams Eng. J.*, 2016.

[20] O. Abedinia and N. Amjady, "Short-term wind power prediction based on Hybrid Neural Network and chaotic shark smell optimization," *Int. J. Precis. Eng. Manuf. - Green Technol.*, 2015.

[21] M. Bidar and H. R. Kanan, "Modified firefly algorithm using fuzzy tuned parameters," in *13th Iranian Conference on Fuzzy Systems, IFSC 2013*, 2013.

[22] J. Pérez, F. Valdez, and O. Castillo, "A new bat algorithm with fuzzy logic for dynamical parameter adaptation and its applicability to fuzzy control design," *Stud. Comput. Intell.*, vol. 574, 2015.

[23] L. Amador-Angulo and O. Castillo, "Optimization of the type-1 and type-2 fuzzy controller design for the water tank using the bee colony optimization," in *2014 IEEE Conference on Norbert Wiener in the 21st Century: Driving Technology's Future, 21CW 2014 - Incorporating the Proceedings of the 2014 North American Fuzzy Information Processing Society Conference, NAFIPS 2014, Conference Proceedings*, 2014, pp. 1–8.

[24] M. Ahmadigorji and N. Amjady, "A multiyear DG-incorporated framework for expansion planning of distribution networks using binary chaotic shark smell optimization algorithm," *Energy*, 2016.

[25] M. Bagheri, A. Sultanbek, O. Abedinia, M. S. Naderi, M. S. Naderi, and N. Ghadimi, "Multi-objective Shark Smell Optimization for Solving the Reactive Power Dispatch Problem," in *Proceedings - 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe, EEEIC/I and CPS Europe 2018*, 2018.

[26] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the International Symposium on Micro Machine and Human Science*, 1995.

[27] R. C. Eberhart and Y. Shi, "Particle swarm optimization: Developments, applications and resources," in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 2001.

[28] L. A. Zadeh, "Knowledge representation in fuzzy logic," *IEEE Trans. Knowl. Data Eng.*, vol. 1, no. 1, pp. 89–100, 1989.

[29] R. Giles, "Łukasiewicz logic and fuzzy set theory," *Int. J. Man. Mach. Stud.*, 1976.

[30] S. Galichet and L. Foulloy, "Fuzzy Controllers: Synthesis and Equivalences," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 2, pp. 140–148, 1995.

[31] Z. Zhao, M. Tomizuka, S. I.-I. transactions on Systems, and U. 1993, "Fuzzy gain scheduling of PID controllers," *ieeexplore.ieee.org*.

[32] S. E. Oltean, M. Dulau, and R. Puskas, "Position control of Robotino mobile robot using fuzzy logic," in *2010 IEEE International Conference on Automation, Quality and Testing, Robotics, AQTR 2010 - Proceedings*, 2010, vol. 1, pp. 366–371.

[33] E. Lizarraga, O. Castillo, J. Soria, and F. Valdez, "A fuzzy control design for an autonomous mobile robot using ant colony optimization," *Stud. Comput. Intell.*, vol. 547, pp. 289–304, 2014.

[34] F. Cuevas and O. Castillo, *Design and implementation of a fuzzy path optimization system for omnidirectional autonomous mobile robot control in real-time*, vol. 749. 2018.

[35] P. Melin, L. Astudillo, O. Castillo, F. Valdez, and M. Garcia, "Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm," *Expert Syst. Appl.*, vol. 40, no. 8, pp. 3185–3195, Jun. 2013.

[36] O. Castillo, P. Melin, and W. Pedrycz, "Design of interval type-2 fuzzy models through optimal granularity allocation," *Appl. Soft Comput. J.*, vol. 11, no. 8, pp. 5590–5601, 2011.

[37] L. Astudillo, P. Melin, and O. Castillo, "Chemical optimization algorithm for fuzzy controller design," in *Springer Briefs in Applied Sciences and Technology*, 2014.

[38] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, 1965.

[39] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets Syst.*, 1978.

[40] F. Cuevas, O. Castillo, and P. Cortes-Antonio, "Towards an Adaptive Control Strategy Based on Type-2 Fuzzy Logic for Autonomous Mobile Robots," in *IEEE International Conference on Fuzzy Systems*, 2019, vol. 2019-June.

[41] F. Cuevas, O. Castillo, and P. Cortes, "Towards a Control Strategy Based on Type-2 Fuzzy Logic for an Autonomous Mobile Robot," 2020, pp. 301–314.

[42] M. Molga and C. Smutnicki, "Test functions for optimization needs," *Test Funct. Optim. needs*, 2005.

[43] L. Astudillo, P. Melin, and O. Castillo, "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*, vol. 294, pp. 3–26, 2013.

[44] C. Caraveo, F. Valdez, and O. Castillo, "A new meta-heuristics of optimization with dynamic adaptation of parameters using type-2 fuzzy logic for trajectory control of a mobile robot," *Algorithms*, vol. 10, no. 3, 2017.