algorithms

MDPI

Article

Shadowed Type-2 Fuzzy Systems for Dynamic Parameter Adaptation in Harmony Search and Differential Evolution Algorithms

Oscar Castillo *, Patricia Melin, Fevrier Valdez, Jose Soria, Emanuel Ontiveros-Robles, Cinthia Peraza and Patricia Ochoa

Tijuana Institute of Technology, Calzada Tecnologico s/n, Fracc. Tomas Aquino, Tijuana 22379, Mexico; pmelin@tectijuana.mx (P.M.); fevrier@tectijuana.mx (F.V.); jsoria57@gmail.com (J.S.); emanuel.ontiveros18@tectijuana.edu.mx (E.O.-R.); cinthia.peraza18@tectijuana.edu.mx (C.P.); martha.ochoa18@tectijuana.edu.mx (P.O.)

* Correspondence: ocastillo@tectijuana.mx

Received: 22 November 2018; Accepted: 3 January 2019; Published: 9 January 2019

check for updates

Abstract: Nowadays, dynamic parameter adaptation has been shown to provide a significant improvement in several metaheuristic optimization methods, and one of the main ways to realize this dynamic adaptation is the implementation of Fuzzy Inference Systems. The main reason for this is because Fuzzy Inference Systems can be designed based on human knowledge, and this can provide an intelligent dynamic adaptation of parameters in metaheuristics. In addition, with the coming forth of Type-2 Fuzzy Logic, the capability of uncertainty handling offers an attractive improvement for dynamic parameter adaptation in metaheuristic methods, and, in fact, the use of Interval Type-2 Fuzzy Inference Systems (IT2 FIS) has been shown to provide better results with respect to Type-1 Fuzzy Inference Systems (T1 FIS) in recent works. Based on the performance improvement exhibited by IT2 FIS, the present paper aims to implement the Shadowed Type-2 Fuzzy Inference System (ST2 FIS) for further improvements in dynamic parameter adaptation in Harmony Search and Differential Evolution optimization methods. The ST2 FIS is an approximation of General Type-2 Fuzzy Inference Systems (GT2 FIS), and is based on the principles of Shadowed Fuzzy Sets. The main reason for using ST2 FIS and not GT2 FIS is because the computational cost of GT2 FIS represents a time limitation in this application. The paper presents a comparison of the conventional methods with static parameters and the dynamic parameter adaptation based on ST2 FIS, and the approaches are compared in solving mathematical functions and in controller optimization.

Keywords: Shadowed Type-2 Fuzzy Logic; differential evolution; harmony search

1. Introduction

Nowadays, metaheuristic optimization methods represent a very interesting alternative for the optimization of complex problems without the mathematical modeling of the problem, and they have been successfully applied in several kinds of application, for example, in control applications [1–3], optimizing Artificial Neural Networks [4–6], optimizing a controller applied in an complex electromechanical process [7], fuzzy controllers [8,9], etc. On the other hand, dynamic parameter adaptation in metaheuristic methods based on fuzzy logic can improve their optimization performance as can be observed in [10–13]. However, this dynamic adaptation based on fuzzy logic significantly increases the computational cost of the optimization process. There are some works where the dynamic adaptation of metaheuristic parameters is realized through Interval Type-2 Fuzzy systems, for example, in [2,3], and in some works, this adaptation is successfully realized by General Type-2 Fuzzy Systems.

However, the main limitation of applying Type-2 Fuzzy systems, specifically General Type-2 Fuzzy Sets, for the dynamic adaptation of metaheuristic parameters, is the higher computational cost.

The computational cost of Interval Type-2 Fuzzy Systems is nearly double that of Type-1 Fuzzy Systems, and the computational cost of General Type-2 Fuzzy Systems depends on the representation used for modeling the system, for example, using the α-planes representation, the computational cost is directly proportional to the number of α-planes used in approximating the model. In this case, the approximation improves accuracy by increasing the number of α-planes, in other words, the computational cost is significantly elevated.

The contribution of the present work is the application of Shadowed Type-2 Fuzzy Systems as a method for approximating General Type-2 Fuzzy Systems modeled with the α-planes representation. The main difference is that the Shadowed Type-2 Fuzzy Inference System (ST2 FIS) approach requires only two α-planes to model the GT2 FIS, but the values of α are selected with the optimization criteria for shadowed sets proposed by Pedrycz in [14], and recent examples of the ST2 FIS applied in control problems can be found in [15]. On the other hand, the optimization of fuzzy controllers that was previously presented, for example in [1,3,10,16], is presented. The reason for exploring this application is because the fuzzy controllers have been proven to have good performance in complex applications, for example, in [17].

The rest of the paper is organized as follows. Section 2 describes type-2 shadowed sets theory, Section 3 shows metaheuristic algorithm concepts, Section 4 explains the dynamic parameter adaptation process, Section 5 summarizes the simulation results, and finally, Section 6 offers the conclusions.

2. Type-2 Fuzzy Systems and Shadowed Sets

With the emergence of Type-1 Fuzzy Inference Systems (T1 FIS) in 1965 [18], computational science achieved the capability to model the vagueness in the real world and create mathematical models that represent human knowledge. Nowadays, fuzzy sets have evolved to GT2 FIS that allows not only a vagueness model, but, in addition, allows an uncertainty modelling approach to be used, and the mathematical expression of the GT2 FIS is denoted in Equation (1):

$$\widetilde{\widetilde{A}} = \left\{ \left((x,u), u_{\widetilde{A}}(x) \right) \middle| \forall x \in X,\ \forall u \in J_x^u \subseteq [0,1] \right\}. \tag{1}$$

In order to apply the GT2 FIS to real-world applications, some alternatives exist for modeling this system, such as the vertical slices or z-slices representation [19], the Geometric approximation [20] and the horizontal slices or α-planes representation [21]. The present work is focused on the α-planes representation that consists of discretizing the secondary axis of GT2 FIS in several horizontal slices called α-planes. These α-planes are expressed by Equation (2) and can be computed as an IT2 FIS [22]. Then, with the union of every α-plane, the GT2 FIS is modeled, as described in Equation (3):

$$\widetilde{A}_\alpha = \{((x,u),\alpha) | \forall x \in X,\ \forall u \in J_x \subseteq [0,1]\} \tag{2}$$

$$\widetilde{\widetilde{A}} = \bigcup \widetilde{A}_\alpha. \tag{3}$$

In order to reduce the computational cost of the α-planes representation, the Shadowed Type-2 FIS was introduced in [23]. The proposal focused on modeling the GT2 FIS with only two optimal α-planes, eliminating the excessive precision, and the selection of the optimal α-planes was performed through the concepts that Pedrycz proposed for the shadowed set theory [14,24–26].

The basic concepts of shadowed sets consist of realizing two α-cuts on a Type-1 fuzzy set, with α and β values. Based on these α-cuts, three intervals are described, as explained in Equation (4).

$$S_{\mu_A}(x) = \begin{cases} 1,\ if\ \mu_A(x) \geq \alpha \\ 0,\ if\ \mu_A(x) \leq \beta \\ [0,1],\ if\ \alpha \leq \mu_A(x) \geq \beta \end{cases} \tag{4}$$

These intervals can be interpreted as three regions: the elevated region for the membership degree of 1, the reduced region for the membership degree of 0, and for the shadowed area, the membership degree is in [0, 1]. Based on these regions, Pedrycz proposed that the optimal α and β values can be obtained by the following expression shown in Equation (5):

$$elevated\ area_{(\alpha,\beta)}(\mu_A) +\ reduced\ area_{(\alpha,\beta)}(\mu_A) = shadowed\ area_{(\alpha,\beta)}(\mu_A). \tag{5}$$

The graphical interpretation of Equation (5) can be appreciated in Figure 1.

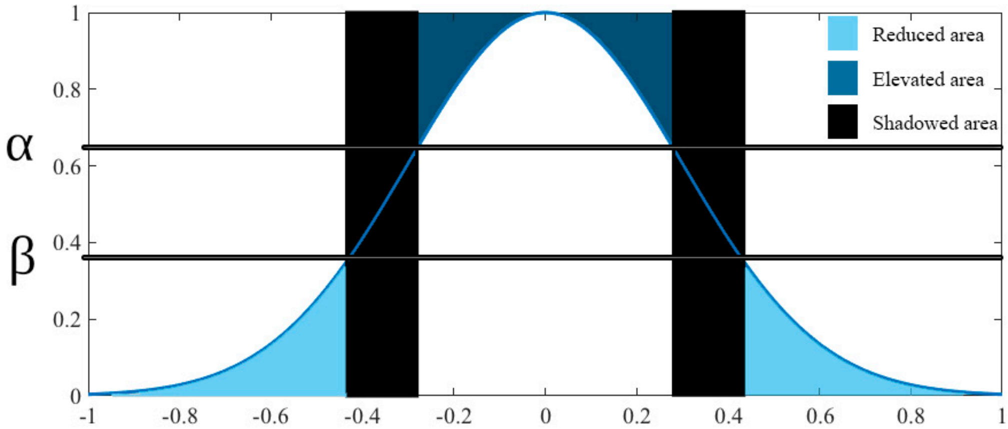


Figure 1. Shadowed set representation.

Then, the optimal α and β values are obtained by the optimization of the $V(\alpha, \beta)$ function described in Equation (6):

$$V(\alpha,\beta) = \left| \int_{x \in A_r} \mu_A(x)dx + \int_{x \in A_e} (1 - \mu_A(x))dx - \int_{x \in S} dx \right|. \tag{6}$$

In [23], Linda and Manic proposed the use of shadowed sets in the secondary axis of the GT2 FIS, finding, in this way, the optimal α and β values and then using these values as α-planes. In this way, the computational cost is reduced, and the implementation of GT2 FIS for dynamic parameter adaptation in metaheuristic algorithms is allowed.

Trapezoidal ST2 MF

For the present paper, it was decided to use the TrapezoidalG (TrapG) ST2 membership function (MF) introduced in [27] that is based on a Trapezoidal GT2 membership function with a Gaussian membership function as a secondary membership function. The equation of this function is expressed in Equation (7), and its illustration is found in Figure 2.

$$\text{TrapG ST2 MF } = \begin{cases} \propto_o \begin{cases} \overline{\mu}_O = \frac{\overline{\mu}_t(x)+\underline{\mu}_t(x)}{2} - 1.449\left|\frac{\overline{\mu}_t(x)-\underline{\mu}_t(x)}{10}\right| \\ \underline{\mu}_O = \frac{\overline{\mu}_t(x)+\underline{\mu}_t(x)}{2} + 1.449\left|\frac{\overline{\mu}_t(x)-\underline{\mu}_t(x)}{10}\right| \end{cases} \\ \propto_I \begin{cases} \overline{\mu}_I = \frac{\overline{\mu}_t(x)+\underline{\mu}_t(x)}{2} - 0.9282\left|\frac{\overline{\mu}_t(x)-\underline{\mu}_t(x)}{10}\right| \\ \underline{\mu}_I = \frac{\overline{\mu}_t(x)+\underline{\mu}_t(x)}{2} + 0.9282\left|\frac{\overline{\mu}_t(x)-\underline{\mu}_t(x)}{10}\right| \end{cases} \end{cases} \tag{7}$$

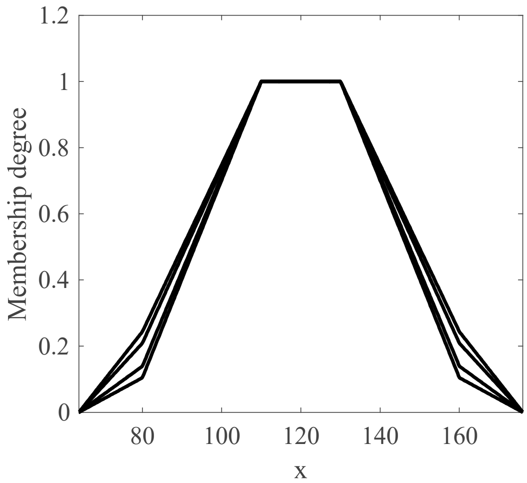


Figure 2. Trapezoidal Shadowed Type-2 (ST2) MF.

3. Metaheuristic Algorithms

The metaheuristic algorithms are iterative methods for general purpose search and optimization. They are iterative procedures that intelligently guide a subordinate heuristic by combining different concepts to properly explore and exploit the search space. This section presents two particular metaheuristic algorithms, which are the Harmony Search algorithm [28] and the Differential Evolution algorithm [29].

3.1. Harmony Search Algorithm

The Harmony Search algorithm (HS) was developed by Zong Woo Geem in 2001 [30]. This algorithm is based on the musical composition, specifically of jazz, and has three main components in the improvisation process which are Harmony Memory Accepting (HMR), Pitch Adjustment (PArate) and Random Selection. The HS includes these 5 steps and their respective equations.

Step 1: Initialize the problem and parameters:

$$\text{Minimize } f(x) s.t.\ \ x(j) \in [LB(j),\ UB(j)\},\ j = 1, 2, \ldots,\ n]. \tag{8}$$

Step 2: Initialize the Harmony Memory (HM):

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \ldots & x_N^1 & f(x^1) \\ x_1^2 & x_2^2 & \ldots & x_N^2 & f(x^2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \ldots & x_N^{HMS} & f(x^{HMS}) \end{bmatrix}. \tag{9}$$

Step 3: Improvise a new Harmony:

$$X_{new}(j) = X_{new}(j) \pm r\ \times BW. \tag{10}$$

Step 4: Update the Harmony Memory:

To update the HM with a new solution vector, x_{new} , the objective function is used to evaluate them. A comparison is made to find out if the new vector solution is better than the worst historical vector solution, and then the worst historical is excluded and substituted with a new one.

Step 5: Check the stopping criteria:

The process is repeated until the number of improvisations (NI) is satisfied; otherwise, the process repeats steps 3 and 4. Finally the best solution is achieved and considered as the best result to the problem. The Harmony Memory Accepting (HMR) parameter represents the intensification or exploitation, the pitch adjustment (PArate), and randomization parameters represent the diversification or exploration of the algorithm.

These components are described in more detail in [31–33].

3.2. Differential Evolution Algorithm

Differential Evolution (DE) is a simple and fairly used algorithm that was originally proposed by Storn and Price in 1994 [34] and is mainly composed of the following operations: initialization of the population structure defined by Equations (11)–(16), initialization by Equation (17), mutation expressed by Equation (18), crossover defined by Equation (19), and selection defined with Equation (20).

The way in which this algorithm works is by initializing its population within a search space depending on the problem. Then, three individuals are selected at random and with them, the mutation and crossover operations are performed. The best individual is selected and passes to the next generation and so on until the stopping criterion of the algorithm is met.

3.2.1. Population Structure

The Differential Evolution algorithm maintains a pair of vector populations, both of which contain Np D-dimensional vectors of real-valued parameters. The current population, symbolized by $P_{\boldsymbol{x}}$, is composed of those vectors, $\boldsymbol{x}_{i,g}$, that have already been found to be acceptable either as initial points, or by comparison with other vectors:

$$P_{\boldsymbol{x},g} = (\boldsymbol{x}_{i,g}),\ i = 0,1,\ldots,\ Np-1,\ \ g = 0,1,\ldots,\ g_{max} \tag{11}$$

$$\boldsymbol{x}_{i,g} = (x_{j,i,g}),\ \ j = 0,1,\ldots,D-1 \tag{12}$$

$$P_{\boldsymbol{v},g} = (\boldsymbol{v}_{i,g}),\ i = 0,1,\ldots,\ Np-1,\ \ g = 0,1,\ldots,\ g_{max} \tag{13}$$

$$\boldsymbol{v}_{i,g} = (v_{j,i,g}),\ \ j = 0,1,\ldots,D-1 \tag{14}$$

$$P_{\boldsymbol{u},g} = (\boldsymbol{u}_{i,g}),\ i = 0,1,\ldots,\ Np-1,\ \ g = 0,1,\ldots,\ g_{max} \tag{15}$$

$$\boldsymbol{u}_{i,g} = (u_{j,i,g}),\ \ j = 0,1,\ldots,D-1. \tag{16}$$

3.2.2. Initialization

Before initializing the population, the upper and lower limits for each parameter must be specified. These 2D values can be collected by two initialized D-dimensional vectors, $\mathrm{b_L}$ and $\mathrm{b_U}$, to which the subscripts L and U indicate the lower and upper limits respectively. Once the initialization limits have been specified, a number generator randomly assigns each parameter in every vector a value within the set range. For example, the initial value (g = 0) of the j-th vector parameter is i^{th}:

$$x_{j,i,0} = rand_j(0,1) \times\ (b_{j,U} - b_{j,L}) +\ b_{j,L}. \tag{17}$$

3.2.3. Mutation

In particular, the differential mutation adds a random sample equation showing how to combine three different vectors chosen randomly to create a mutant vector:

$$\boldsymbol{v}_{i,g} =\ \boldsymbol{x}_{r_0,g} + \mathrm{F} \times (\boldsymbol{x}_{r_1,g} - \boldsymbol{x}_{r_2,g}). \tag{18}$$

3.2.4. Crossover

To complement the differential mutation search strategy, DE also uses uniform crossover which is sometimes known as discrete recombination (dual). In particular, DE crosses each vector with a mutant vector:

$$u_{i,g} = u_{j,i,g}\begin{cases} v_{j,i,g}\ if\ \left(rand_j(0,1) \le Cr\ or\ j = j_{rand}\right) \\ x_{j,i,g} \qquad otherwise \end{cases}. \tag{19}$$

3.2.5. Selection

If the test vector, $\boldsymbol{u}_{i,g}$, has a value of the objective function equal to or less than its target vector, $x_{i,g}$ It replaces the target vector in the next generation; otherwise, the target retains its place in the population for at least another generation:

$$\boldsymbol{x}_{i,g+1} = \begin{cases} \boldsymbol{u}_{i,g}\ if\ f\left(\boldsymbol{u}_{i,g}\right) \le\ f\left(\boldsymbol{x}_{i,g}\right) \\ \boldsymbol{x}_{i,g}\ otherwise \end{cases}. \tag{20}$$

The operations of mutation, recombination, and selection are applied repeatedly until the optimal solution is found, or the specified terminating pre-criteria are satisfied.

4. Dynamic Parameter Adaptation

In this section, we explain in detail the structure of the fuzzy system used for each of the HS and DE algorithms. The fuzzy system used for both algorithms has one input and one output. In the case of the HS algorithm, the input parameter is the iterations and for the output, the HMR parameter is used, and for the DE algorithm, the input parameter is the generations and the output parameter is F (mutation). Equation (21) is used to calculate the input of the fuzzy system according to the method:

$$Experiment = \frac{Current\ Experiment}{Maximum\ of\ experiments}. \tag{21}$$

In Equation (21), the experiment refers to iterations for the FHS method and generations for the FDE method. The current experiment represents the current iterations or generations and the maximum of experiments represents the maximum number of iterations and generations.

The parameters of the outputs mentioned above are converted into fuzzy parameters based on the following Equations (22) and (23):

$$HMR = \frac{\sum_{i=1}^{r_{hmr}} \mu_i^{hmr}\ (hmr_{1i})}{\sum_{i=1}^{r_{hmr}} \mu_i^{hmr}} \tag{22}$$

where HMR is the memory consideration; r_{hmr} is the number of rules of the Shadowed Type-2 Fuzzy System corresponding to hmr; hmr_{1i} is the output result for rule i corresponding to hmr; μ_i^{hmr} is the membership function of rule i corresponding to hmr.

$$F = \frac{\sum_{i=1}^{r_F} \mu_i^F\ (F_{1i})}{\sum_{i=1}^{r_F} \mu_i^F} \tag{23}$$

where F is the mutation; r_{hmr} is the number of rules of the Shadowed Type-2 Fuzzy System corresponding to F; F_{1i} is the output result for rule i corresponding to F; μ_i^F is the membership function of rule i corresponding to F.

Both fuzzy systems in the input and outputs are granulated into three membership functions, as shown in Figures 3 and 4 respectively. They are granulated into *low*, *medium*, and *high*, and the rules are described in Tables 1 and 2.

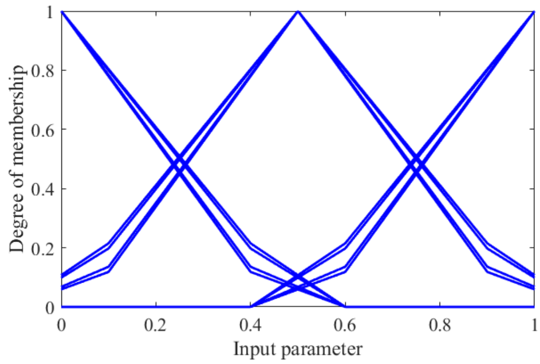


Figure 3. Input parameter membership functions.

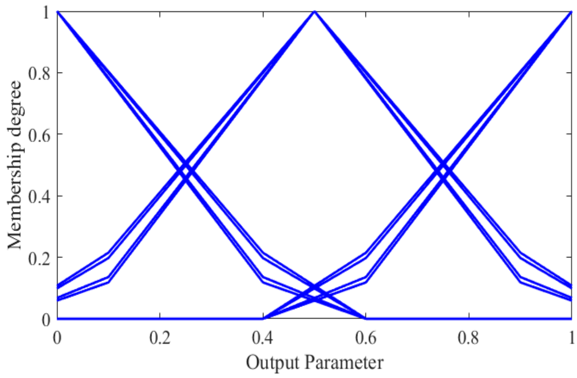


Figure 4. Output parameter membership functions.

Table 1. Rules of the ST2FHS fuzzy system.

Iteration \ **HMR**	**Low**	**Medium**	**High**
Low	Low	–	–
Medium	–	Medium	–
High	–	–	High

Table 2. Rules of the ST2FDE fuzzy system.

Generation \ **F**	**Low**	**Medium**	**High**
Low	–	–	Low
Medium	–	Medium	–
High	High	–	–

The rules are based on previous experimentation as presented in [16,35], and for the ST2FDE method are used in a decreasing fashion and for the ST2FHS method are used in an increasing fashion, respectively.

5. Experiments

The experiments designed to evaluate the proposed approach are divided into two categories: first, the optimization of mathematical functions and secondly the optimization of controllers. Benchmark problems are widely used to validate the appropriate performance of algorithms and their variants. In this case, benchmark mathematical functions and a control problem are used to validate the correct operation of the ST2FHS and ST2FDE methods.

5.1. Mathematical Functions

There are several types of benchmark mathematical function; in this case, the functions chosen for the experiment are summarized in Table 3.

Table 3. Benchmark mathematical functions.

Function	Search Domain	f min	Equation
Sum Squares	$[-10, 10]^{n}$	0	$f(x) = \sum_{i=1}^{n} ix_i^2$
Trid	$[-100, 100]^{n}$	−200	$f(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$
Zakharov	$[-5, 10]^{n}$	0	$f(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i+1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4$
Ackley	$[-15, 30]^{n}$	0	$f(x) = a \cdot exp\left(-b \times \sqrt{\frac{1}{n} \sum_{i=1}^{n} cos(cx_i)}\right) + a + exp(1)$
Dixon & Price	$[-10, 10]^{n}$	0	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} (2x_i^2 - x_{i-1})^2$
Levy	$[-10, 10]^{n}$	0	$f(x) = sin^2(\pi w_1) + \sum_{i=1}^{n-1} (w_i - 1)^2 [1 + 10\, sin^2(\pi w_i + 1)] + (w_n - 1)^2 [1 + sin^2 (2\pi w_n)],\ where\ w_i = 1 + \frac{x_i - 1}{4},\ for\ all\ i = 1, \ldots, n$
Griewank	$[-600, 600]^{n}$	0	$f(x) = \frac{1}{400} \sum_{i=1}^{n} X_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
Powell	$[-4, 5]^{n}$	0	$f(x) = \sum_{i=1}^{n/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$
Power Sum	$[0, 10]^{n}$	0	$f(x) = \sum_{i=1}^{n} [(\sum_{j=1}^{n} x_j^i) - b_i]^2$

The parameters used for the experimentation are the following: 100 individuals or harmonies, 30 executions, 2000 iterations or generations, and 10, 50 dimensions for most of the functions, except for the *Power sum* function that only uses 4 dimensions.

The statistical test that is used is the Z-test that is based on Equation (24), and the parameters used for this test are an alpha of 0.05, a level of confidence of 95%, and a sample size of 30. The main goal is to verify that by using the methods for dynamic parameter adjustment with Shadowed Type-2, we can obtain a better result with respect to the original methods for all values lower than −1.645:

$$Z = \frac{(\overline{X}_1 - \overline{X}_2) - (\mu_1 - \mu_2)}{(\sigma_1 - \sigma_2)}. \tag{24}$$

Thirty experiments were performed using the HS, DE, ST2FHS, and ST2FDE methods. Tables 4–12 summarize the results obtained for the HS and ST2FHS methods, and Tables 13–20 show the results obtained for the DE and ST2FDE methods. In these tables, the best, worst, average, standard deviation, and Z-values obtained for each mathematical function are presented.

Table 4. Results for the Sum Squares function.

Dimension	10		Z-Value	50		Z-Value
Method	HS	ST2FHS		HS	ST2FHS	
Best	3.80×10^{-1}	9.76×10^{-4}		5.89×10^{-1}	1.90×10^{-3}	
Worst	1.26×10	3.50×10^{-3}	−16.78	1.02×10	3.00×10^{-3}	−52.09
Average	7.07×10^{-1}	2.02×10^{-3}		8.10×10^{-1}	2.28×10^{-3}	
SD	2.30×10^{-1}	5.19×10^{-4}		8.48×10^{-2}	2.67×10^{-4}	

Table 5. Results for the Zakharov function.

Dimension	**10**		**Z-Value**	**50**		**Z-Value**
Method	HS	ST2FHS		HS	ST2FHS	
Best	7.34×10^{-11}	4.00×10^{-13}		2.04×10^{-4}	3.12×10^{-4}	
Worst	1.35×10^{-8}	2.0869×10^{-9}	-4.72	6.97×10^{-3}	9.47×10^{-4}	-2.41
Average	3.27×10^{-9}	3.63×10^{-10}		1.42×10^{-3}	6.37×10^{-4}	
SD	3.33×10^{-9}	4.73×10^{-10}		1.76×10^{-3}	1.49×10^{-4}	

Table 6. Results for the Dixon & Price function.

Dimension	**10**		**Z-Value**	**50**		**Z-Value**
Method	HS	ST2FHS		HS	ST2FHS	
Best	7.30×10^{-3}	1.39×10^{-3}		2.56×10	1.53×10	
Worst	8.64×10^{-1}	8.53×10^{-1}	-0.4	1.50×10	4.99×10	-7.67
Average	5.27×10^{-1}	4.94×10^{-1}		7.95×10	2.72×10	
SD	3.19×10^{-1}	3.07×10^{-1}		3.66×10	7.73×10^{-1}	

Table 7. Results for the Levy function.

Dimension	**10**		**Z-Value**	**50**		**Z-Value**
Method	HS	ST2FHS		HS	ST2FHS	
Best	1.18×10^{-4}	5.44×10^{-5}		1.42×10^{-2}	1.19×10^{-2}	
Worst	3.39×10^{-4}	3.25×10^{-4}	-0.23	1.98×10^{-2}	1.63×10^{-2}	-8.4
Average	2.25×10^{-4}	1.90×10^{-4}		1.66×10^{-2}	1.41×10^{-2}	
SD	5.64×10^{-5}	1.92×10^{-4}		1.33×10^{-3}	1.01×10^{-3}	

Table 8. Results for the Griewank function.

Dimension	**10**		**Z-Value**	**50**		**Z-Value**
Method	HS	ST2FHS		HS	ST2FHS	
Best	4.57×10^{-2}	1.55×10^{-1}		3.65×10^{-2}	9.75×10^{-1}	
Worst	8.49×10^{-1}	2.94×10^{-1}	-0.29	2.15×10	1.02×10	0.22
Average	2.28×10^{-1}	2.25×10^{-1}		1.00×10	1.00×10	
SD	4.57×10^{-2}	3.65×10^{-2}		1.59×10^{-2}	1.02×10^{-2}	

Table 9. Results for the Power Sum function.

Dimension	**4**		**Z-Value**
Method	HS	ST2FHS	
Best	2.02×10^{-2}	0.00×10	
Worst	2.47×10	0.00×10	-0.59
Average	2.23×10	1.71×10	
SD	4.48×10	1.84×10	

Table 10. Results for the Trid function.

Dimension	**10**		**Z-Value**	**50**		**Z-Value**
Method	HS	ST2FHS		HS	ST2FHS	
Best	$-1.23 \times 10^{+2}$	$-1.90 \times 10^{+2}$		$-3.49 \times 10^{+3}$	$-2.98 \times 10^{+3}$	
Worst	$-1.19 \times 10^{+2}$	$-1.80 \times 10^{+2}$	-8.31	$-1.33 \times 10^{+3}$	$-1.39 \times 10^{+3}$	0.49
Average	$-1.21 \times 10^{+2}$	$-1.85 \times 10^{+2}$		$-2.26 \times 10^{+3}$	$-2.20 \times 10^{+3}$	
SD	9.64×10^{-1}	1.99×10		$5.19 \times 10^{+2}$	$4.07 \times 10^{+2}$	

Table 11. Results for the Ackley function.

Dimension	10		Z-Value	50		Z-Value
Method	HS	ST2FHS		HS	ST2FHS	
Best	5.66×10^{-5}	6.68×10^{-5}		5.14×10^{-2}	5.75×10^{-2}	
Worst	6.64×10^{-4}	8.46×10^{-4}	−0.05	1.67×10	5.75×10^{-2}	−0.6
Average	2.76×10^{-4}	2.74×10^{-4}		1.66×10^{-1}	1.21×10^{-1}	
SD	1.50×10^{-4}	1.65×10^{-4}		3.48×10^{-1}	2.08×10^{-1}	

Table 12. Results for the Powell function.

Dimension	10		Z-Value	50		Z-Value
Method	HS	ST2FHS		HS	ST2FHS	
Best	8.20×10^{-3}	3.64×10^{-4}		1.23×10^{-2}	4.30×10^{-3}	
Worst	1.70×10^{-1}	2.40×10^{-3}	−7.03	4.32×10^{-1}	1.76×10^{-2}	−4.33
Average	5.88×10^{-2}	1.37×10^{-3}		8.18×10^{-2}	1.11×10^{-2}	
SD	4.47×10^{-2}	5.17×10^{-4}		8.90×10^{-2}	3.43×10^{-3}	

The results shown in Table 4 through Table 12 using the set of functions achieve significant evidence using 10 and 50 dimensions for the Sum Square, Zakharov, and Powell functions.

In the Dixon & Price and Levy functions, only significant evidence with 50 dimensions is obtained, and for the Trid function, there is only evidence in 10 dimensions.

Table 13. Results for the Sum Squares function.

Dimension	10		Z-Value	50		Z-Value
Method	DE	ST2FDE		DE	ST2FDE	
Best	9.48146×10^{-33}	2.2026×10^{-31}		3.703507	2.7471×10^{-7}	
Worst	3.80055×10^{-31}	1.0944×10^{-29}	4.5513	6.582105	8.4137×10^{-7}	−34.1921
Average	1.14269×10^{-31}	1.7395×10^{-30}		4.851488	4.72×10^{-7}	
SD	9.32651×10^{-32}	1.9536×10^{-30}		0.777158	1.41×10^{-7}	

Table 14. Results for the Zakharov function.

Dimension	10		Z-Value	50		Z-Value
Method	DE	ST2FDE		DE	ST2FDE	
Best	1.04×10^{-4}	1.84×10^{-8}		7.70×10	2.57×10^{-2}	
Worst	7.64×10^{-4}	1.87×10^{-7}	−10.5785	$1.43 \times 10^{+2}$	1.06×10^{-1}	−37.0422
Average	3.40×10^{-4}	8.13×10^{-8}		$1.13 \times 10^{+2}$	5.87×10^{-2}	
SD	1.76×10^{-4}	4.82×10^{-8}		1.67×10	2.41×10^{-2}	

Table 15. Results for the Dixon & Price function.

Dimension	10		Z-Value	50		Z-Value
Method	DE	ST2FDE		DE	ST2FDE	
Best	1.77×10^{-1}	2.54×10^{-1}		$1.60 \times 10^{+2}$	6.68×10^{-1}	
Worst	6.67×10^{-1}	6.67×10^{-1}	0.4408	$4.01 \times 10^{+2}$	7.55×10^{-1}	−26.3743
Average	5.98×10^{-1}	6.13×10^{-1}		$2.66 \times 10^{+2}$	6.79×10^{-1}	
SD	1.49×10^{-1}	1.12×10^{-1}		5.51×10	1.81×10^{-2}	

Table 16. Results for the Levy function.

Dimension	10		Z-Value	50		Z-Value
Method	DE	ST2FDE		DE	ST2FDE	
Best	1.50×10^{-32}	1.50×10^{-32}		3.20×10	4.74×10^{-7}	
Worst	1.50×10^{-32}	1.50×10^{-32}	0	5.94×10	1.55×10^{-6}	−33.7726
Average	1.50×10^{-32}	1.50×10^{-32}		4.68×10	9.43×10^{-7}	
SD	2.78×10^{-48}	8.35×10^{-48}		7.59×10^{-1}	2.69×10^{-7}	

Table 17. Results for the Power Sum function.

Dimension	4		Z-Value
Method	DE	ST2FDE	
Best	5.80×10^{-4}	7.67×10^{-4}	
Worst	2.21×10^{-2}	2.26×10^{-2}	−1.6605
Average	9.15×10^{-3}	6.65×10^{-3}	
SD	6.18×10^{-3}	5.46×10^{-3}	

Table 18. Results for the Trid function.

Dimension	10		Z-Value	50		Z-Value
Method	DE	ST2FDE		DE	ST2FDE	
Best	−209	−209		−861.94	−884.656	
Worst	−209	−209	0	−854.374	−884.646	−76.9252
Average	−209	−209		−858.184	−884.651	
SD	0	0		1.884478	0.002398	

Table 19. Results for the Ackley function.

Dimension	10		Z-Value	50		Z-Value
Method	DE	ST2FDE		DE	ST2FDE	
Best	4.44×10^{-15}	8.88×10^{-16}		1.82×10	1.60×10^{-4}	
Worst	4.44×10^{-15}	4.44×10^{-15}	−5.3864	2.55×10	2.74×10^{-4}	−65.4158
Average	4.44×10^{-15}	2.66×10^{-15}		2.15×10	2.16×10^{-4}	
SD	1.60×10^{-30}	1.81×10^{-15}		1.80×10^{-1}	2.75×10^{-5}	

Table 20. Results for the Powell function.

Dimension	10		Z-Value	50		Z-Value
Method	DE	ST2FDE		DE	ST2FDE	
Best	3.30×10^{-7}	3.15×10^{-8}		$7.59 \times 10^{+2}$	3.54×10^{-2}	
Worst	5.76×10^{-6}	3.54×10^{-7}	−9.5085	$1.32 \times 10^{+3}$	1.13×10^{-1}	−36.7394
Average	1.99×10^{-6}	9.44×10^{-8}		$1.08 \times 10^{+3}$	6.62×10^{-2}	
SD	1.09×10^{-6}	6.50×10^{-8}		$1.61 \times 10^{+2}$	1.94×10^{-2}	

The results shown in Table 13 through Table 20 using the set of functions achieve significant evidence using 10 and 50 dimensions for the Zakharov, Dixon & Price, Ackley, and Powell functions. In the Sum Square, Levy, and Trid functions, significant evidence is only found with 50 dimensions.

5.2. Controllers Optimization

As a benchmark control problem it was decided to deal with controlling the angular position of a DC Motor, as this is a non-stable control problem that has been considered in the literature to evaluate controllers, for example, in [36–39]. The plant is illustrated in Figure 5, and the space-state equations are expressed in Equations (25) and (26), respectively. Table 21 shows the parameters of the motor position.

$$\frac{d}{dt}\begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{K}{J} \\ 0 & -\frac{K}{L} & -\frac{R}{L} \end{bmatrix}\begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} \tag{25}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} \tag{26}$$

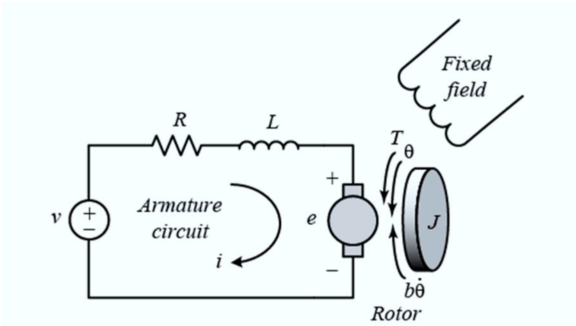


Figure 5. Motor position plant.

Table 21. Parameters of the motor position.

Symbol	Definition	Value
J	Moment of inertia of the rotor	3.2284×10^{-6} kg.m^2
b	Motor viscous friction constant	3.5077×10^{-6} Nms
Ke	Electromotive force constant	0.0274 V/rad/sec
Kt	Motor torque constant	0.0274 Nm/Amp
R	Electric resistance	4 ohm
L	Electric inductance	2.75×10^{-6} H

5.2.1. Fuzzy controller

The motor position is regulated with a Type-1 (T1) fuzzy controller, which is composed of two inputs and one output, granulated into trapezoidal and triangular membership functions.

The mathematical expression of the trapezoidal membership function is described in Equation (27), and the triangular membership function is described in Equation (28), and the membership functions parameters are presented in Table 22, and the fuzzy system contains 15 fuzzy rules, which are shown in Figure 6.

$$trapmf(x,a,b,c,d) = \begin{cases} 0, \ x \leq a \\ \frac{x-a}{b-a}, \ a \leq x \leq b \\ 1, \ b \leq x \leq c \\ \frac{d-x}{d-c}, \ c \leq x \leq d \\ 0, \ x \geq d \end{cases} \tag{27}$$

$$trimf(x,a,b,c) = \begin{cases} 0, \ x \leq a \\ \frac{x-a}{b-a}, \ a \leq x \leq b \\ \frac{c-x}{c-b}, \ b \leq x \leq c \\ 0, \ c \geq x \end{cases} \tag{28}$$

Table 22. Type-1 membership functions.

		Input *Error*		
MF	A	b	c	d
NegV	−1	−1	−0.5	0
CeroV	−0.5	0	0.5	-
PosV	0	0.5	1	1
		Input *Error Change*		
ErrNeg	−1	−1	−0.4	−0.1
ErrNegM	−0.4	−0.2	0	-
SinErr	−0.09	0	0.10	-
ErrMaxM	0	0.2	0.4	-
ErrMax	0.1	0.4	1	-

Table 22. *Cont.*

	Output *Voltage*			
MDis	−1	−1	−0.6	−0.09
MDism	−0.4	−0.2	0	-
Man	−0.1	0	0.1	-
Aumm	0	0.2	0.4	-
Aum	0.09	0.6	1	1

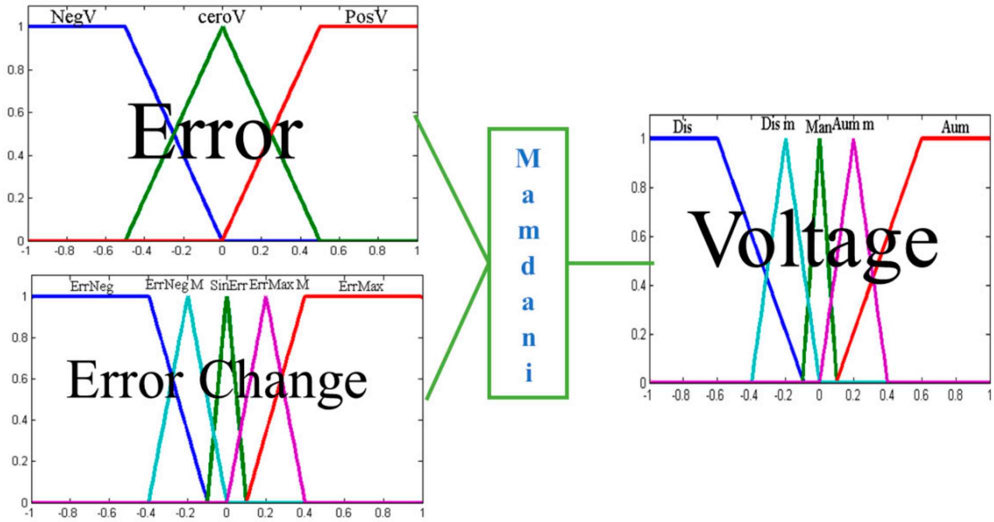


Figure 6. Structure of the motor position.

A graphic representation of Table 22 is illustrated in Figure 6. The contents of the fuzzy system are two inputs and one output. The first input, called *Error*, is composed of three membership functions, both of the edges are of the trapezoidal type, and the central is triangular. The second input, called *Error Change*, contains five functions of membership, two of the edges are the trapezoidal type, and the three central ones are the triangular type. Finally, the output, called *Voltage*, contains five functions of membership, two of the edges are the trapezoidal type, and the three central ones are the triangular type.

Figure 7 represents the surface of the fuzzy system for the motor position and Table 23 summarize the rules of the controller.

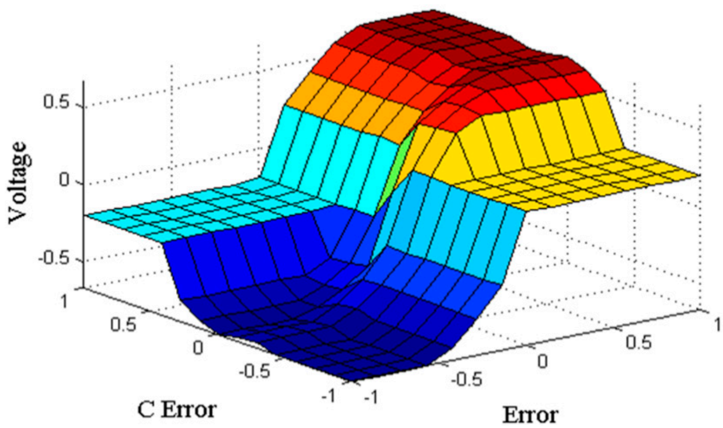


Figure 7. Surface of the fuzzy system.

Table 23. Fuzzy rules for the controller.

Voltage		**Error Change**				
		ErrNeg	ErrNeg_M	SinErr	ErrMax_M	ErrMax
Error	NegV	Dis	Dis	Dis	Dis	Dis_m
	CeroV	Aum_m	Aum_m	Man	Dis_m	Dis_m
	PosV	Aum_m	Aum	Aum	Aum	Aum

The proposed ST2FHS and ST2FDE methods are used to optimize the values of the membership functions of the motor position controller in order to minimize the RMSE error described in Equation (29). Figure 8 shows the vector that represents the information of the individuals.

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(x_t - \hat{x}_t)^2} \tag{29}$$

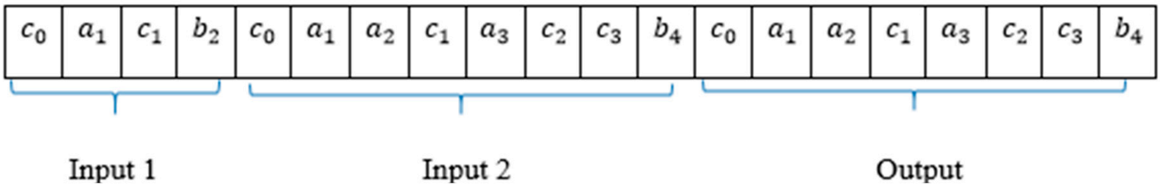


Figure 8. Representation of the individuals for the T1 FIS.

The vector contains the information of each individual; the latter represents each position of the triangular or trapezoidal membership function. In total, there are 44 positions of which 25 are fixed and 20 are optimized. The limits of the positions that are optimized are shown in Table 24.

Table 24. Boundary T1 membership functions parameters of the vector.

	Input 1	**Input 2**	**Output**
MF parameters	First MF $a_0 = b_0 = -1$ $-1 < c_0 < -0.5$ $d_0 = 0$	First MF $a_0 = b_0 = -1$ $-1 < c_0 < -0.4$ $d_0 = -0.1$	First MF $a_0 = b_0 = -1$ $-1 < c_0 < -0.6$ $d_0 = -0.09$
		Second MF $-1 < a_1 < -0.1$ $b_1 = -0.2$ $-0.1 < c_1 < 0$	Second MF $-1 < a_1 < -0.4$ $b_1 = -0.2$ $-0.2 < c_1 < 0$
	Second MF $-1 < a_1 < 0$ $b_1 = 0$ $0 < c_1 < 1$	Third MF $-0.09 < a_2 < 0$ $b_2 = 0$ $0 < c_2 < 0.10$	Third MF $-0.1 < a_2 < 0$ $b_2 = 0$ $0 < c_2 < 0.1$
		Fourth MF $0 < a_3 < 0.10$ $b_3 = 0.2$ $0.1 < c_3 < 1$	Fourth MF $0 < a_3 < 0.1$ $b_3 = 0.2$ $0.09 < c_3 < 1$
	Third MF $a_2 = 0$ $0 < b_2 < 0.5$ $c_2 = d_2 = 1$	Fifth MF $a_4 = 0.1$ $0.2 < b_4 < 0.4$ $c_4 = d_4 = 1$	Fifth MF $a_4 = 0.09$ $0.2 < b_4 < 0.6$ $c_4 = d_4 = 1$

Tables 25 and 26 show the results obtained by optimizing the parameters of the DC Motor controller, with the ST2FHS and ST2FDE methods, respectively. The noise applied to this controller is 0.5 (Gaussian random number).

Table 25. Results for the ST2FHS method.

Method	HS-FLC without Noise	ST2FHS-FLC without Noise	HS-FLC with Noise	ST2FHS-FLC FLC with Noise
Best	7.86×10^{-3}	7.32×10^{-3}	4.22×10^{-2}	1.95×10^{-2}
Worst	5.16×10^{-1}	5.66×10^{-2}	1.09×10	9.07×10^{-1}
Average	1.65×10^{-1}	9.22×10^{-3}	5.90×10^{-1}	4.62×10^{-1}
SD	1.37×10^{-1}	3.45×10^{-3}	3.07×10^{-1}	2.83×10^{-1}

Table 26. Results for the ST2FDE method.

Method	DE-FLC without Noise	ST2FDE-FLC without Noise	DE-FLC with Noise	ST2FDE-FLC with Noise
Best	7.34×10^{-3}	4.35×10^{-3}	2.24×10^{-2}	5.89×10^{-4}
Worst	2.1×10^{-2}	7.43×10^{-3}	4.85×10^{-1}	7.47×10^{-2}
Average	1.71×10^{-2}	7.24×10^{-3}	2.44×10^{-1}	2.18×10^{-2}
SD	2.81×10^{-3}	5.35×10^{-4}	1.36×10^{-1}	1.90×10^{-2}

Figures 9 and 10 illustrate the best results obtained with the HS method without noise and with noise respectively, Figures 11 and 12 illustrate the best controller surface.

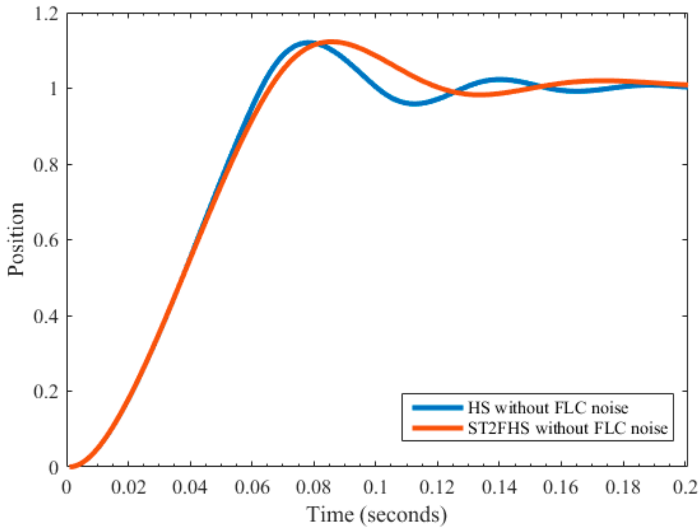


Figure 9. HS and ST2FHS without noise.

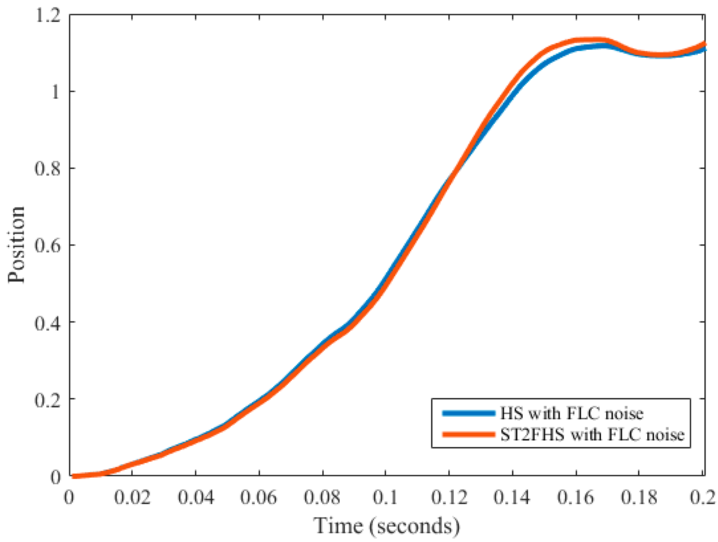
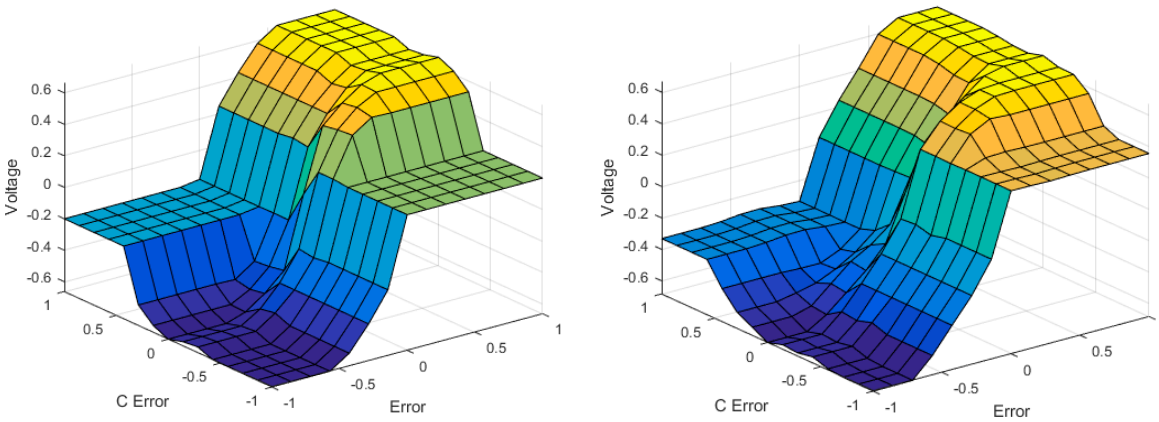
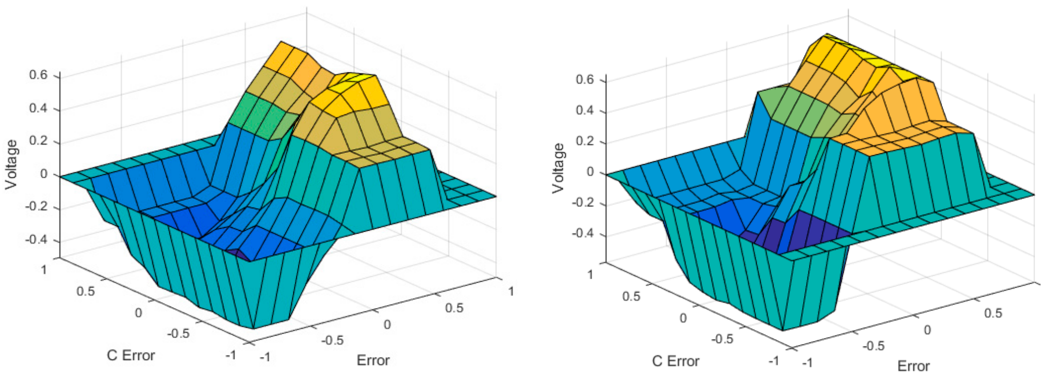


Figure 10. HS and ST2FHS with noise.



(**a**) (**b**)

Figure 11. Comparison of the surface for each method for the motor position controller: (**a**) HS without noise algorithm; (**b**) ST2FHS without noise algorithm.



(**a**) (**b**)

Figure 12. Comparison of the surface for each method for the motor position controller: (**a**) HS with noise algorithm; (**b**) ST2FHS with noise algorithm.

Figures 13 and 14 illustrate the best results obtained with the DE method without noise and with noise, respectively. Figures 15 and 16 illustrate the best controller surface.

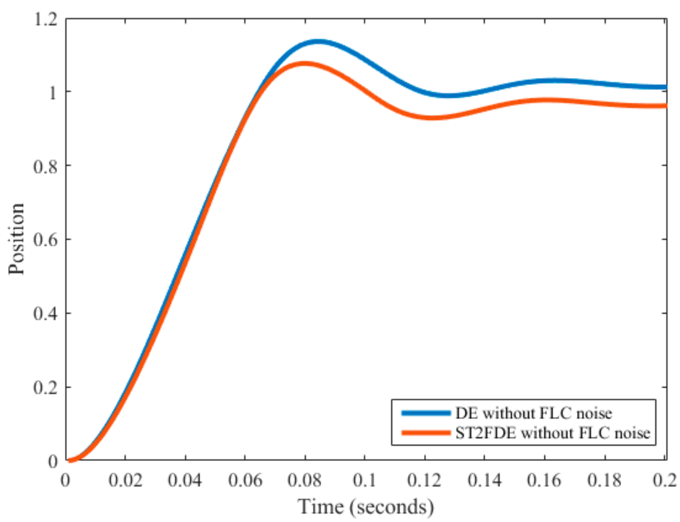


Figure 13. DE and ST2FDE without noise.

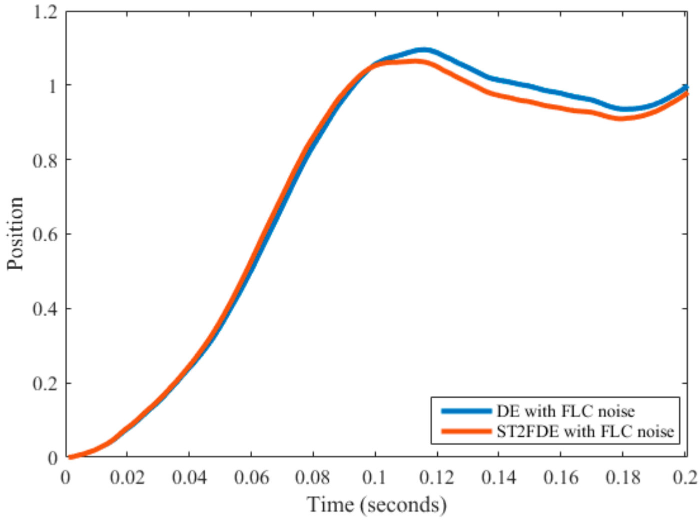
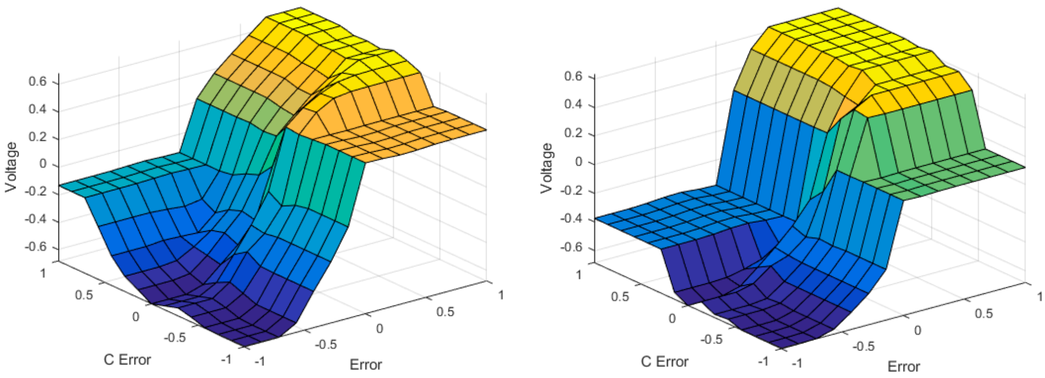
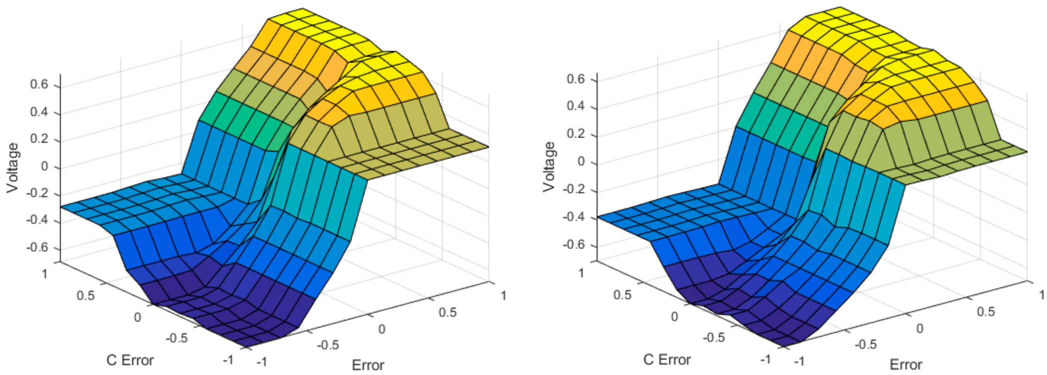


Figure 14. DE and ST2FDE with noise.



(**a**) (**b**)

Figure 15. Comparison of the surface for each method for the motor position controller: (**a**) DE without noise algorithm; (**b**) ST2FDE without noise algorithm.



(**a**) (**b**)

Figure 16. Comparison of the surface for each method for the motor position controller: (**a**) DE with noise algorithm; (**b**) ST2FDE with noise algorithm.

It is noticeable in Figures 13 and 15 that the proposed approach obtains a better performance than the conventional approach. However, in order to validate this, a statistical test is realized. Based on the parameters of the z-test statistic presented in Section 5.1 and Equation (24), the results of the Z-values are presented in Table 27.

Table 27. Results for the statistical test of the DC motor.

Method	μ_1	μ_2	Z-Value
ST2FHS	FLC without noise	HS	−6.23
	FLC with noise	HS with noise	−1.67
ST2FDE	FLC without noise	DE	−18.8799
	FLC with noise	DE with noise	−8.8627

The Z-values obtained for the statistical test demonstrate the improvement of the proposed approach with respect to the original method.

5.2.2. PID Control

The PID controller (Proportional-Integral-Derivative) is a feedback control mechanism that is widely used in industrial control systems, for example, in [40–42]. The PID control algorithm uses three different parameters: the proportional (k_p), the integral (k_i), and the derivative (k_d) gains. The proportional value depends on the current error. The integral depends on past errors, and the derivative is a prediction of future errors. The sum of these three actions is used to adjust the process by means of a control element, such as the position of a control valve or the power supplied to a heater. The general PID algorithm is expressed in Equation (30):

$$r\ (t) = MV\ (t) = Kp\ e(t) + Ki\ \int_o^t e\ (t)\ \mathrm{d}t + Kd\ \frac{de(t)}{dt}. \tag{30}$$

In this case, the PID angular position of the motor is used, the ST2FHS and ST2FDE methods are used for the optimization of this control problem, and the control objective is to minimize the settling time. The transfer function for this controller is expressed in Equation (31), and the structure of the control system is shown in Figure 17:

$$C(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}. \tag{31}$$

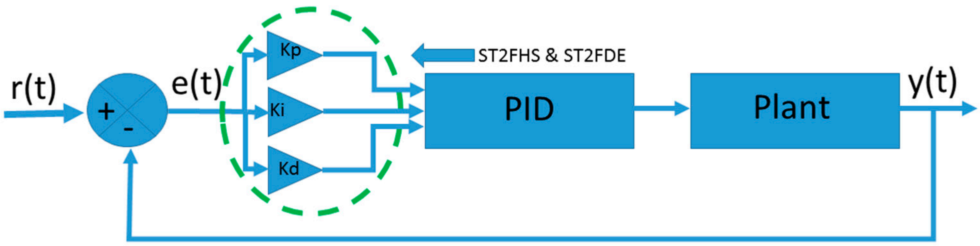


Figure 17. Structure of the PID (Proportional-Integral-Derivative) DC motor.

Table 28 shows the best parameters obtained from the optimization with the HS, ST2FHS, DE, and ST2FDE methods and the time in which the objective of stabilizing the motor position is achieved. Figures 18 and 19 show the graphical representation of the simulation of these parameters for the HS and ST2FHS, DE, and ST2FDE methods, respectively. Table 29 shows the results for the statistical test.

Table 28. Results for the experiments with PID for the DC motor.

Method	K_p	K_i	K_d	Best Settling Time
PID	21	500	0.15	0.0338
HS	600	12,000	6	0.00029
DE	900	18,000	9	0.00020
ST2FHS	900	30,000	9	0.00020
ST2FDE	1500	30,000	15	0.00012

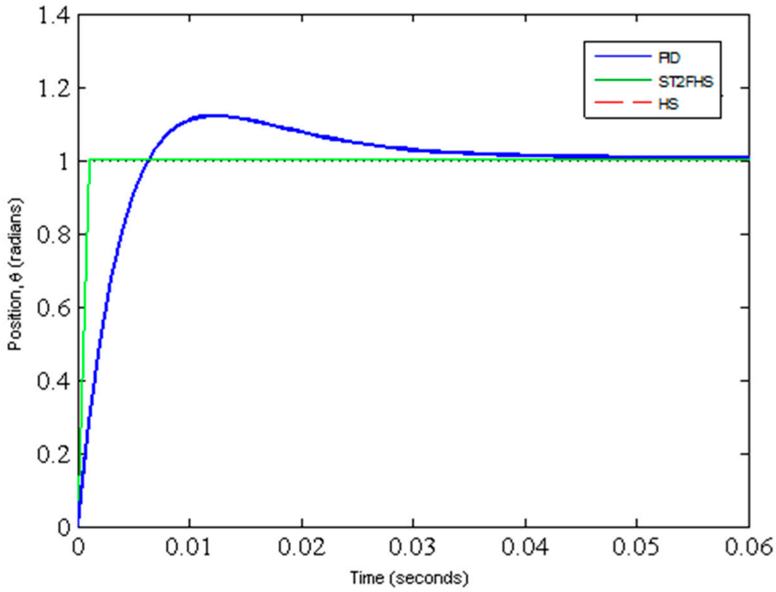


Figure 18. Best result obtained from the HS and ST2FHS methods.

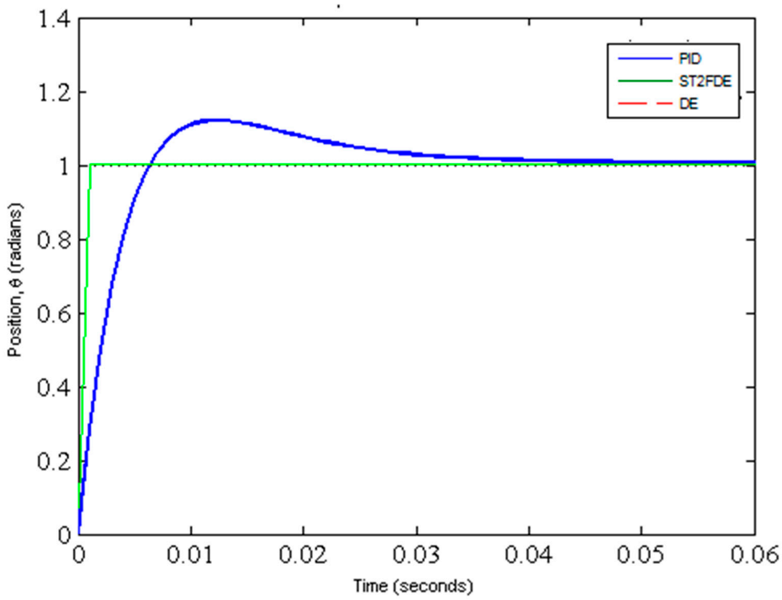


Figure 19. Best results obtained from the DE and ST2FDE methods.

Figures 18 and 19 show visually similar results, Table 27 contains the results of the values of each PID parameter obtained when using the proposed method, and it can be noted that there is a difference in the settling time between the two methods.

Table 29 presents the statistical test to validate the improvement of the proposed approach with respect to the conventional method.

Table 29. Results for the statistical test of the PID motor.

Method	ST2FHS		HS		Z-Value
Method	Average	Std.	Average	SD	−3.21
	3.90×10^{-4}	1.83×10^{-4}	5.72×10^{-3}	9.07×10^{-3}	
	ST2FDE		DE		
	Average	Std.	Average	SD	−2.53
	3.03×10^{-4}	2.15×10^{-4}	4.22×10^{-3}	8.46×10^{-3}	

The Z-values obtained in Table 29 demonstrate that the proposed approach obtains better performance with respect to the conventional approach.

6. Conclusions

In this study, we present the use of a dynamic adaptation of parameters based on the Shadowed Type-2 Fuzzy Inference System theory using the original HS and DE algorithms, which, in this paper, we call ST2FHS and ST2FDE, respectively.

Three case studies were considered. The first was done to obtain the minimum of each benchmark mathematical function using the ST2FHS and ST2FDE methods; the second and third case studies optimized the membership functions of the problem of motor position plant of the engine with the proposed methodology. The difference between the second and third case studies was the type of controller used. For the second case, an FLC was used with noise and without noise and for the third, a PID was used. We can conclude generally and statistically that, for both algorithms, by using the proposed methodology, favorable results were obtained for all cases considered in this paper. The successful implementation of ST2 FIS corresponds to an implementation of an approximation of GT2 FIS; however, with this approach, the computational cost cannot be a limitation in the application of this kind of method in an application that requires several executions.

Author Contributions: Formal analysis, J.S.; Investigation, P.M. and E.O.-R.; Methodology, O.C. and F.V.; Software, C.P.; Validation, P.O.

Funding: This research was funded by Consejo Nacional de Ciencia y Tecnología: 122.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. 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.* **2016**, *43*, 131–142. [CrossRef]
2. Castillo, O.; Amador-Angulo, L.; Castro, J.R.; Garcia-Valdez, M. A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems. *Inf. Sci.* **2016**, *354*, 257–274. [CrossRef]
3. 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.* **2017**, *53*, 74–87. [CrossRef]
4. Gaxiola, F.; Melin, P.; Valdez, F.; Castro, J.R.; Castillo, O. Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO. *Appl. Soft Comput.* **2016**, *38*, 860–871. [CrossRef]
5. Pulido, M.; Melin, P.; Castillo, O. Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange. *Inf. Sci.* **2014**, *280*, 188–204. [CrossRef]
6. Saadat, J.; Moallem, P.; Koofigar, H. Training Echo State Neural Network Using Harmony Search Algorithm. *Int. J. Artif. Intell.* **2017**, *15*, 163–179.
7. Martin, D.; del Toro, R.; Guerra, R.E.H.; Dorronsoro, J. Optimal tuning of a networked linear controller using a multi-objective genetic algorithm and its application to one complex electromechanical process. *Int. J. Innov. Comput. Inf. Control.* **2009**, *5*, 3405–3414.

8. Precup, R.; David, R.; Petriu, E.M. Grey Wolf Optimizer Algorithm-Based Tuning of Fuzzy Control Systems with Reduced Parametric Sensitivity. *IEEE Trans. Ind. Electron.* **2017**, *64*, 527–534. [CrossRef]
9. Vrkalovic, S.; Lunca, E.-C.; Borlea, I.-D. Model-Free Sliding Mode and Fuzzy Controllers for Reverse Osmosis Desalination Plants. *Int. J. Artif. Intell.* **2018**, *16*, 208–222.
10. Peraza, C.; Valdez, F.; Melin, P. Optimization of Intelligent Controllers Using a Type-1 and Interval Type-2 Fuzzy Harmony Search Algorithm. *Algorithms* **2017**, *10*, 82. [CrossRef]
11. Peraza, C.; Valdez, F.; Castillo, O. Study on the Use of Type-1 and Interval Type-2 Fuzzy Systems Applied to Benchmark Functions Using the Fuzzy Harmony Search Algorithm. In *Fuzzy Logic in Intelligent System Design*; Melin, P., Castillo, O., Kacprzyk, J., Reformat, M., Melek, W., Eds.; Springer: Cham, Switzerland, 2018; Volume 648, pp. 94–103.
12. Liu, J.; Lampinen, J. A Fuzzy Adaptive Differential Evolution Algorithm. *Soft Comput.* **2005**, *9*, 448–462. [CrossRef]
13. Brest, J.; Zumer, V.; Maucec, M.S. Self-Adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization. In Proceedings of the 2006 IEEE International Conference on Evolutionary Computation, Vancouver, BC, Canada, 16–21 July 2006; pp. 215–222.
14. Pedrycz, W. Shadowed sets: Representing and processing fuzzy sets. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1998**, *28*, 103–109. [CrossRef] [PubMed]
15. Melin, P.; Ontiveros-Robles, E.; Gonzalez, C.I.; Castro, J.R.; Castillo, O. An approach for parameterized shadowed type-2 fuzzy membership functions applied in control applications. *Soft Comput.* **2018**, 1–15, Accepted for be published. [CrossRef]
16. Ochoa, P.; Castillo, O.; Soria, J. Differential Evolution Algorithm with Interval Type-2 Fuzzy Logic for the Optimization of the Mutation Parameter. In *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*; Castillo, O., Melin, P., Kacprzyk, J., Eds.; Springer: Cham, Switzerland, 2018; Volume 749, pp. 55–65.
17. Melgarejo, M.A.; Munoz, C.M.; Leottau, L. A hierarchical design approach for interval type-2 fuzzy controllers applied to mobile robots. *Int. J. Robot. Autom.* **2012**, *27*, 330. [CrossRef]
18. Zadeh, L.A. Fuzzy sets. *Inf. Control.* **1965**, *8*, 338–353. [CrossRef]
19. Wagner, C.; Hagras, H. Toward General Type-2 Fuzzy Logic Systems Based on zSlices. *IEEE Trans. Fuzzy Syst.* **2010**, *18*, 637–660. [CrossRef]
20. Coupland, S.; John, R. Geometric Type-1 and Type-2 Fuzzy Logic Systems. *IEEE Trans. Fuzzy Syst.* **2007**, *15*, 3–15. [CrossRef]
21. Mendel, J.M.; Liu, F.; Zhai, D. α-Plane Representation for Type-2 Fuzzy Sets: Theory and Applications. *IEEE Trans. Fuzzy Syst.* **2009**, *17*, 1189–1207. [CrossRef]
22. Mendel, J.M.; John, R.I.; Liu, F. Interval Type-2 Fuzzy Logic Systems Made Simple. *IEEE Trans. Fuzzy Syst.* **2006**, *14*, 808–821. [CrossRef]
23. Wijayasekara, D.; Linda, O.; Manic, M. Shadowed Type-2 Fuzzy Logic Systems. In Proceedings of the 2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), Singapore, 16–19 April 2013; pp. 15–22.
24. Pedrycz, W. From fuzzy sets to shadowed sets: Interpretation and computing. *Int. J. Intell. Syst.* **2009**, *24*, 48–61. [CrossRef]
25. Pedrycz, W.; Song, M. Granular fuzzy models: A study in knowledge management in fuzzy modeling. *Int. J. Approx. Reason.* **2012**, *53*, 1061–1079. [CrossRef]
26. Pedrycz, W.; Vukovich, G. Granular computing in the development of fuzzy controllers. *Int. J. Intell. Syst.* **1999**, *14*, 419–447. [CrossRef]
27. Gonzalez, C.I.; Melin, P.; Castillo, O.; Juarez, D.; Castro, J.R. Toward General Type-2 Fuzzy Logic Systems Based on Shadowed Sets. *Adv. Fuzzy Log. Technol.* **2017**, *642*, 131–142.
28. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A New Heuristic Optimization Algorithm: Harmony Search. *Simulation* **2001**, *76*, 60–68. [CrossRef]
29. Tang, L.; Zhao, Y.; Liu, J. An Improved Differential Evolution Algorithm for Practical Dynamic Scheduling in Steelmaking-Continuous Casting Production. *IEEE Trans. Evol. Comput.* **2014**, *18*, 209–225. [CrossRef]
30. Geem, Z.W. State-of-the-Art in the Structure of Harmony Search Algorithm. In *Recent Advances in Harmony Search Algorithm*; Geem, Z.W., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 270, pp. 1–10.

31. Geem, Z.W.; Choi, J.-Y. Music Composition Using Harmony Search Algorithm. In *Applications of Evolutinary Computing*; Giacobini, M., Ed.; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4448, pp. 593–600.
32. Lee, K.S.; Geem, Z.W. A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 3902–3933. [CrossRef]
33. Kim, J.H.; Lee, H.M.; Yoo, D.G. Investigating the Convergence Characteristics of Harmony Search. In *Harmony Search Algorithm*; Kim, J.H., Geem, Z.W., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 382, pp. 3–10.
34. Ochoa, P.; Castillo, O.; Soria, J. Differential Evolution with Dynamic Adaptation of Parameters for the Optimization of Fuzzy Controllers. In *Recent Advances on Hybrid. Approaches for Designing Intelligent Systems*; Springer: Cham, Switzerland, 2014; pp. 275–288.
35. Peraza, C.; Valdez, F.; Castillo, O. Improved Method Based on Type-2 Fuzzy Logic for the Adaptive Harmony Search Algorithm. In *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*; Castillo, O., Melin, P., Kacprzyk, J., Eds.; Springer: Cham, Switzerland, 2018; Volume 749, pp. 29–37.
36. Lin, P.-H.; Hwang, S.; Chou, J. Comparison on fuzzy logic and PID controls for a DC motor position controller. In Proceedings of the 1994 IEEE Industry Applications Society Annual Meeting, Denver, CO, USA, 2–6 October 1994; Volume 3, pp. 1930–1935.
37. Buzi, E.; Marango, P. A Comparison of conventional and nonconventional methods of DC motor speed control. *IFAC Proc. Vol.* **2013**, *46*, 50–53. [CrossRef]
38. Puangdownreong, D.; Nawikavatan, A.; Thammarat, C. Optimal Design of I-PD Controller for DC Motor Speed Control System by Cuckoo Search. *Procedia Comput. Sci.* **2016**, *86*, 83–86. [CrossRef]
39. El-samahy, A.A.; Shamseldin, M.A. Brushless DC motor tracking control using self-tuning fuzzy PID control and model reference adaptive control. *Ain Shams Eng. J.* **2018**, *9*, 341–352. [CrossRef]
40. Lianghong, W.; Yaonan, W.; Shaowu, Z.; Wen, T. Design of PID controller with incomplete derivation based on differential evolution algorithm. *J. Syst. Eng. Electron.* **2008**, *19*, 578–583. [CrossRef]
41. Verma, B.; Padhy, P.K. Optimal PID controller design with adjustable maximum sensitivity. *IET Control. Theory Appl.* **2018**, *12*, 1156–1165. [CrossRef]
42. Wei, C.; Söffker, D. Optimization Strategy for PID-Controller Design of AMB Rotor Systems. *IEEE Trans. Control. Syst. Technol.* **2016**, *24*, 788–803. [CrossRef]