

Article

Optimization of Intelligent Controllers Using a Type-1 and Interval Type-2 Fuzzy Harmony Search Algorithm

Cinthia Peraza, Fevrier Valdez * and Patricia Melin * 

Division of Graduate Studies and Research, Tijuana Institute of Technology, 22414 Tijuana, Mexico; Cinthia_sita@hotmail.com

* Correspondence: fevrier@tectijuana.mx (F.V.); pmelin@tectijuana.mx (P.M.); Tel.: +52-664-623-6318 (F.V. & P.M.)

Academic Editor: Javier Del Ser Lorente

Received: 15 June 2017; Accepted: 16 July 2017; Published: 20 July 2017

Abstract: This article focuses on the dynamic parameter adaptation in the harmony search algorithm using Type-1 and interval Type-2 fuzzy logic. In particular, this work focuses on the adaptation of the parameters of the original harmony search algorithm. At present there are several types of algorithms that can solve complex real-world problems with uncertainty management. In this case the proposed method is in charge of optimizing the membership functions of three benchmark control problems (water tank, shower, and mobile robot). The main goal is to find the best parameters for the membership functions in the controller to follow a desired trajectory. Noise experiments are performed to test the efficacy of the method.

Keywords: Type-1 fuzzy sets; fuzzy control; Type-1 fuzzy logic systems; interval Type-2 fuzzy logic systems; fuzzy harmony search; dynamic parameter adaptation

1. Introduction

At present there are different techniques to solve optimization problems [1–4]. This article focuses on the optimization of parameters to achieve the tracking of a trajectory applied to benchmark control problems. The proposed method is based on the original harmony search algorithm, which mimics the process of music improvisation, which has been widely used to solve different problems, as shown in [5–9]. The proposed method is called fuzzy harmony search algorithm (FHS) that performs an adaptation of parameters with Type-1 and interval Type-2 fuzzy logic and it is used to optimize fuzzy tracking controllers so that they follow desired trajectories for benchmark control problems. Other metaheuristics that have been used to solve optimization problems based on fuzzy logic are [10–15]. In previous works this HS algorithm has been used for different optimization problems, as can be found in [16–19].

The operation of non-linear plants with traditional logic or Type-1 fuzzy logic can be complex for a person who does not have exact knowledge of each of the characteristics that compose it, and using fuzzy set concepts and the principles of fuzzy logic can model the functioning of a plant in a more natural way with the use of linguistic variables as they could have valves like little, much, etc. so that anyone could understand the operation of such plants. In 1965 Zadeh proposed the concepts of a fuzzy set in [20], fuzzy logic in [21], and the concept of a membership functions in [22]. Based on these concepts, complex problems can be solved by using fuzzy rules to represent an expert knowledge of a problem in a more natural way using linguistic variables.

Zadeh also proposed interval Type-2 fuzzy logic in [22], to achieve better management of the uncertainty of a problem, and this type of system is used to solve complex problems where there is

uncertainty on the degree of membership of a value in a particular fuzzy set. Applying the concepts of Type-2 fuzzy logic, we can design a better fuzzy system since with the handling of uncertainty we can approximate the exact behavior of the real world; this is done using a footprint of uncertainty in the membership functions, which consider a range of membership degrees and not only one particular membership degree.

The main contribution of this paper is the optimization of the values of the membership function of the fuzzy system of the three benchmark control problems presented and the stability obtained in the trajectory tracking of each case. The proposed method uses the harmony search algorithm with dynamic parameter adaptation with Type-1 and interval Type-2 fuzzy logic.

This article is organized as follows: Section 2 gives a description of the original harmony search algorithm, Section 3 shows the fuzzy harmony search algorithm with dynamic parameter adaptation, Section 4 describes the benchmark control problems, Section 5 describes the simulation results, Section 6 shows the statistical comparison, and Section 7 presents conclusions.

2. Harmony Search Algorithm

The harmony search algorithm is a recently proposed metaheuristic search technique that mimics the process of music improvisation [23–27]. There are different variants of this algorithm that have been created to improve it or solve specific problems [19,28–30]. This method is composed of the following steps:

- Step 1: Initialize the problem and parameters

Minimize

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

where $f(x)$ is the objective function, $x(j)$ is the decision variables, n is the number of design variables, and $LB(j)$, $UB(j)$ are the lower and upper of the design variables, respectively.

- Step 2: Initialize the harmony memory (HM)

The harmony memory (HM) is a matrix of harmony solution vectors, where its size depends on the harmony memory size (HMS). The HM is occupied by many randomly generated solution vectors and sorted by the objective function value, where $X_i = \{x_i(1), x_i(2), \dots, x_i(n)\}$ represents the i th harmony vector, which is randomly generated as follows: $x_i(j) = LB(j) + (UB(j) - LB(j)) \times r$ for $j = 1, 2, \dots, n$ and $i = 1, 2, \dots, HMS$, where r is a uniform random number $[0, 1]$.

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

- Step 3: Improvise a new harmony

This step is the most important because it influences the overall performance of the HS algorithm. A new harmony memory vector is improvised using three selected operators, which are the memory considerations (HMR), pitch adjustment (PAR), and randomization. The random numbers, r_1 and r_2 range between 0 and 1. In each improvisation or iteration, if r_1 is less than HMR the solution vector and $x_{new}(j)$ is generated from the memory consideration; otherwise, $x_{new}(j)$ is selected through randomization. Next, if r_2 is less than the PAR, the solution vector $x_{new}(j)$ from the memory consideration is further modified with the bandwidth distance using Equation (3):

$$X_{new}(j) = X_{new}(j) \pm r \times BW, \tag{3}$$

where r is a uniform random number ranging from 0 and 1. The function of BW is an arbitrary distance bandwidth. The value of BW is determined by the changes in quantity of the new solution vector and its presence is enhanced by the performance of HS.

- Step 4: Update harmony memory

To update the HM with a new solution vector, x_{new} , an objective function will be used to evaluate it. A comparison is made to see if the new vector solution is better than the worst historical vector solution and the worst historical solution is excluded and replaced 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. The HMR parameter represents the intensification or exploitation; the PAR and randomization parameters represent the diversification or exploration in the algorithm.

3. Fuzzy Harmony Search Algorithm with Dynamic Parameter Adaptation

The value of the HMR parameter is constant during the process of improvisation of the traditional HS algorithm; based on this it was decided to make it a fuzzy parameter for achieving dynamic parameter adaptation. The proposed method is based on the original harmony search algorithm described in Section 2, and specifically focuses on step 3, which is improvisation and uses fuzzy logic to perform the dynamic adjustment of the HMR parameter as the number of iterations or improvisations advance to achieve an improvement in the convergence of the algorithm and obtain better results. To represent this idea, this parameter is converted into a fuzzy parameter. In this method, this value is considered to be fuzzy as it is updated within the FHS progress, and is determined by Equation (4), where HMR has a changing value in the range 0–1:

$$HMR = \frac{\sum_{i=1}^{r_{hmr}} \mu_i^{hmr} (hmr_{1i})}{\sum_{i=1}^{r_{hmr}} \mu_i^{hmr}}, \tag{4}$$

where HMR is the memory considerations; r_{hmr} is the number of rules of the fuzzy system corresponding to hmr ; hmr_{1i} is the output result for rule i corresponding to hmr ; and μ_i^{hmr} is the membership function of rule i corresponding to hmr . To represent this idea in a concise way, the pseudocode of the proposed method is shown in Figure 1:

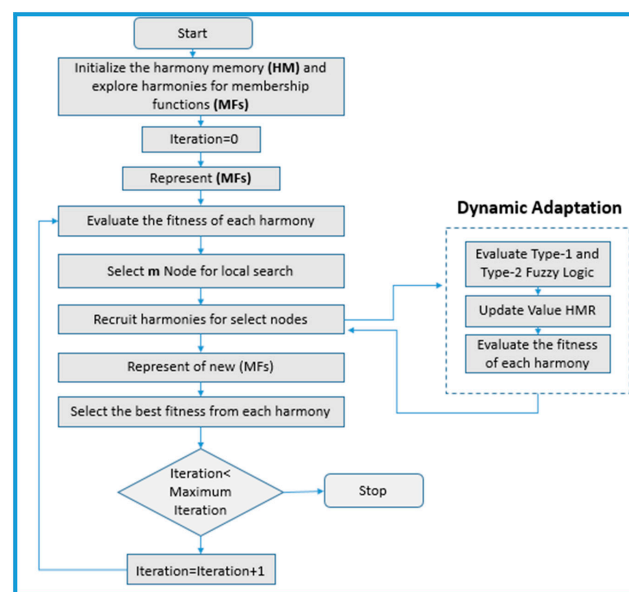


Figure 1. Flowchart of the FHS.

The proposed Type-1 fuzzy system is illustrated in Figures 2 and 3, representing the input and output for this method. The interval Type-2 fuzzy system is illustrated by Figures 4 and 5, which show the input and output for this method.

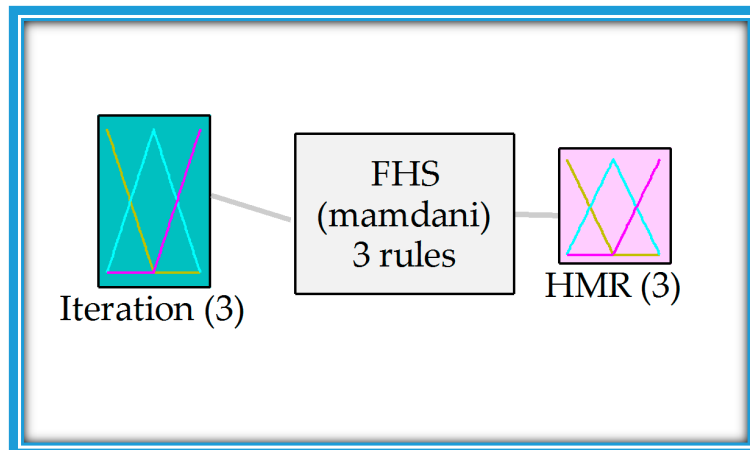


Figure 2. Scheme of the Type-1 fuzzy system method (FHS).

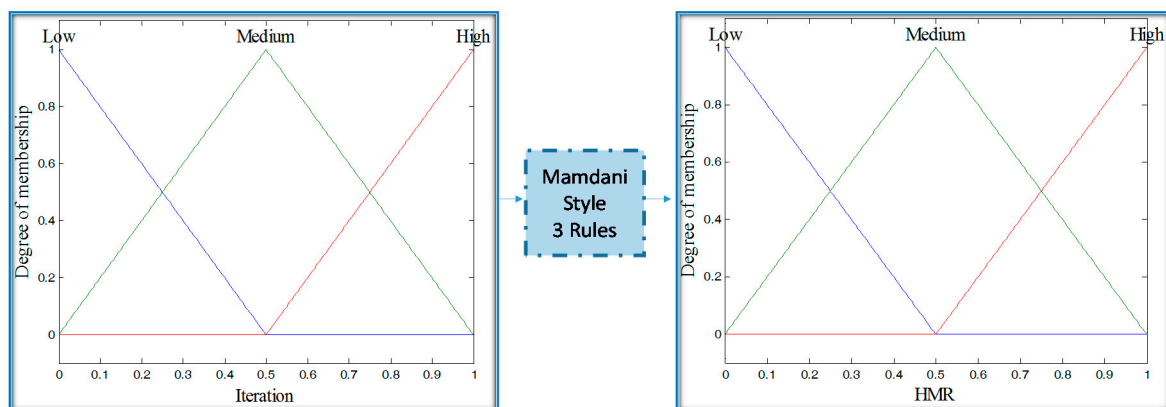


Figure 3. Input and output of the Type-1 fuzzy system.

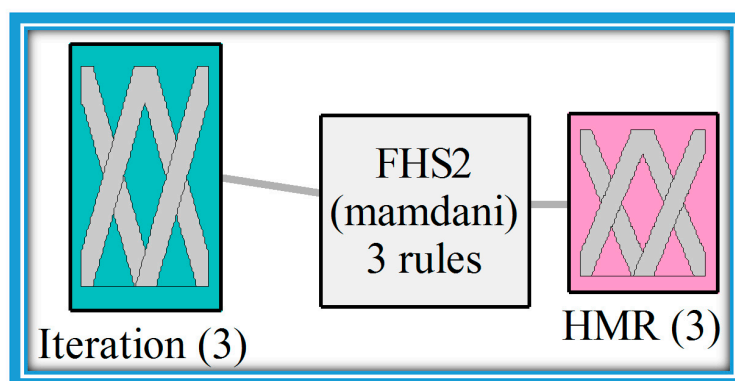


Figure 4. Scheme of the interval Type-2 fuzzy system method (FHS2).

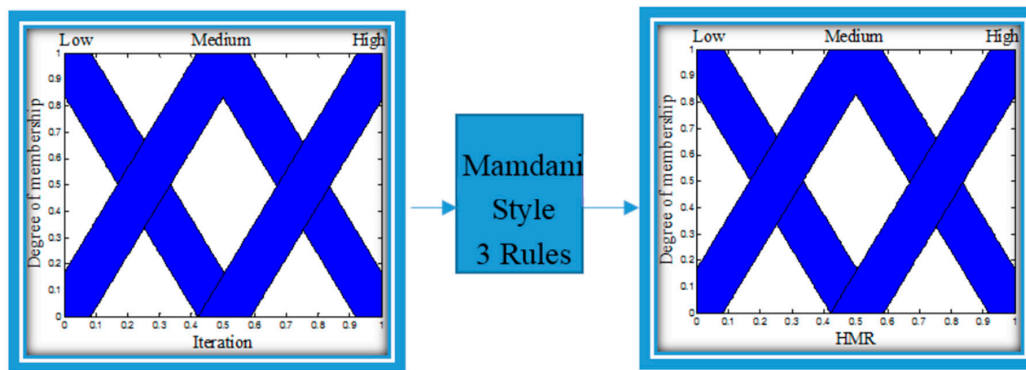


Figure 5. Input and output of the interval Type-2 fuzzy system.

The Type-1 and Type-2 FHS methods use an output that is *HMR* parameter and the input is the iterations as presented in Figures 3 and 5, respectively.

The representation of the input and the output variables in the fuzzy systems are shown in Figures 3 and 5. The input is known as the “iteration”, and is granulated into three triangular membership functions, which are low, medium, and high. The output called “*HMR*” uses values from 0 to 1, the output is granulated into three triangular membership functions, low, medium, and high, and the fuzzy systems are of Mamdani type and contain three rules.

The rules are designed based on the study of parameters of the algorithm, so that the initial iterations explore and the final iterations exploit the search space. In this case the rules have an increasing trend, as shown in Table 1. These rules are the same in the Type-1 and interval Type-2 fuzzy systems.

Table 1. Rules for the Type-1 and interval Type-2 fuzzy system.

Iteration \ HMR	Low	Medium	High
Low	Low	-	-
Medium	-	Medium	-
High	-	-	High

Table 1 represents the idea of increasing the rules as iterations are progressing. To represent the linguistic variable of iterations, the following equation is used:

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

This phase considers a percentage of elapsed iterations to find the values of *HMR*. It initializes with low values of *HMR* so that the algorithm has diversification and then achieves intensification.

4. Control Cases

This section gives a description of the three cases of benchmark control considered for achieving the control of the reference and with these three problems the operation of the proposed method is verified using dynamical parameter adaptation with Type-1 and interval Type-2 fuzzy logic.

4.1. Water Tank Controller

The main goal of the water tank controller is to maintain the level of water in a tank and the activation of the valve to keep the tank full, preventing the water from draining out of the tank. The diagram of the operation of this problem is shown in Figure 6.

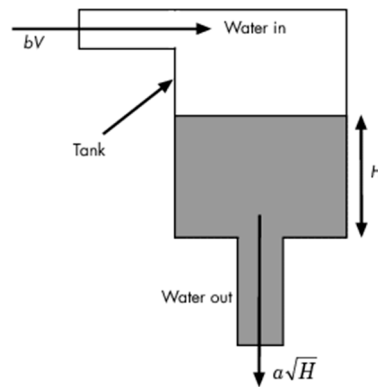


Figure 6. Diagram of the water tank controller.

Water enters the tank from the top at a rate proportional to the voltage, V , applied to the pump. The water leaves through an opening in the tank base at a rate that is proportional to the square root of the water height, H , in the tank. This controller uses fuzzy logic to control the valve that determines how fast the water can enter the tank to maintain the desired level of water. The presence of the square root in the water flow rate results in a nonlinear plant. The mathematical model of the water tank can be represented as a differential equation and is given by Equation (6):

$$\frac{d}{dt}Vol = A \frac{dH}{dt} = bV - a\sqrt{H}. \tag{6}$$

Equation (6) describes the height of water H as a function of time, due to the difference between flow rates into and out of the tank.

- Vol is the volume of water in the tank
- V is the voltage applied to the pump
- A is the cross-sectional area of the tank
- b is a constant related to the flow rate into the tank
- a is a constant related to the flow rate out of the tank

This controller has two inputs and one output; the first input is the level, which has three Gaussian membership functions with linguistic variables called high, okay, and low. The second input is the rate, which has three Gaussian membership functions with linguistic variables of negative, none, and positive. The output is the valve, which has five triangular membership functions with linguistic variables of close fast, close low, no change, open slow, and open fast. This controller is of the Mamdani type and uses five rules, which are presented in detail in Table 2.

Table 2. Rules for the water tank controller.

Rule Number	Inputs			Output
	Level	Operator	Rate	Valve
1	okay	-	-	no change
2	low	-	-	open fast
3	high	-	-	close fast
4	okay	and	positive	close slow
5	okay	and	negative	open slow

The rules that are outlined in Table 2 are those that maintain control of the opening of the valve and with this have leveled the water tank. Figure 7 shows the structure of the fuzzy system for this controller.

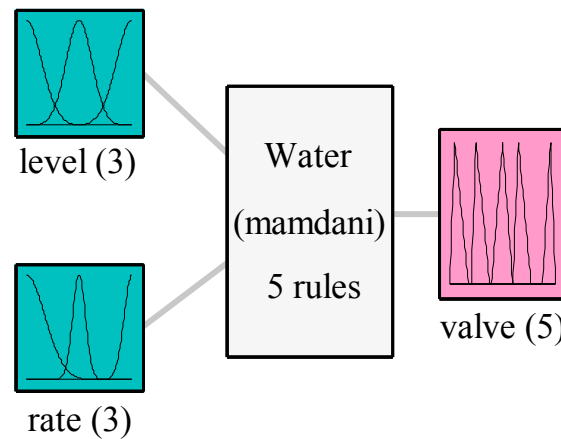


Figure 7. Structure of the water tank fuzzy system for control.

4.2. Temperature Controller

The main goal of this controller is to regulate the temperature and water flow. This controller is composed of two inputs and two outputs, the first input is the temp, which has two trapezoidal and one triangular membership functions with linguistic variables of Cold, Good, and Hot. The second input is the flow, which has two trapezoidal and one triangular membership functions with linguistic variables of Soft, Good, and Hard. The first output is the cold variable, which has five triangular membership functions with linguistic values of close fast, close low, steady, open slow, and open fast. The second output is the hot variable, which has five triangular membership functions with linguistic values of close fast, close low, steady, open slow, and open fast. The diagram of the operation of this controller is illustrated in Figure 8.

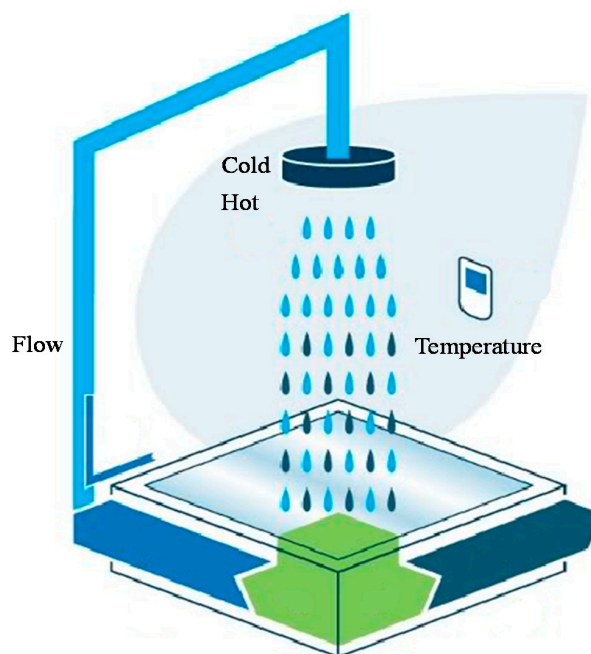


Figure 8. Diagram of the temperature controller.

This controller is of the Mamdani type and uses nine fuzzy rules, which are presented in a detailed fashion in Table 3.

Table 3. Fuzzy rules for the temperature controller.

Rule Number	Inputs			Output	
	Temp	Operator	Flow	Cold	Hot
1	cold	and	soft	open slow	open fast
2	cold	and	good	close slow	open slow
3	cold	and	hard	close fast	close slow
4	good	and	soft	open slow	open slow
5	good	and	good	steady	steady
6	good	and	hard	close slow	close slow
7	hot	and	soft	open fast	open slow
8	hot	and	good	open slow	close slow
9	hot	and	hard	close slow	close fast

Table 3 presents in a detailed fashion each of the rules that compose the temperature controller to achieve the control objective of this controller, and Figure 9 shows the structure of the fuzzy system for this controller.

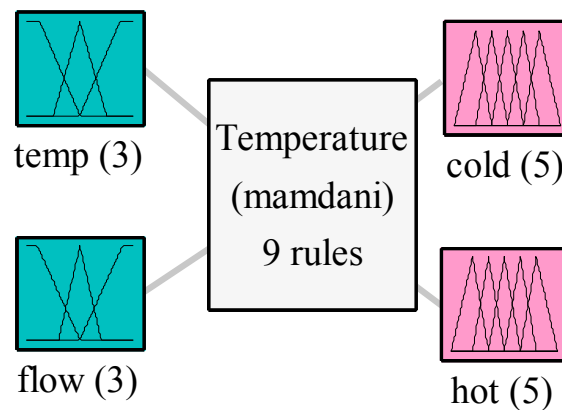


Figure 9. Structure of the water tank fuzzy system for control.

4.3. Mobile Robot Controller

The main goal of this controller is to follow a reference trajectory, based on the model of a unicycle mobile robot [31], which is composed of two drive wheels located on the same axis and a front free wheel that is used only for stability. The graphical representation of this model can be found in Figure 10.

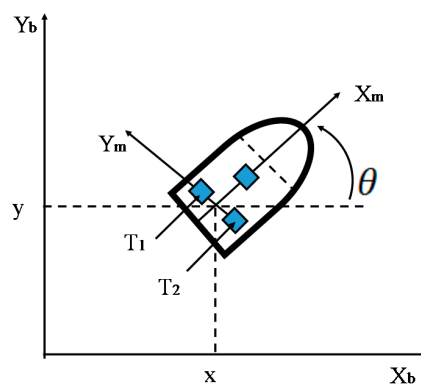


Figure 10. Diagram of the robot mobile controller.

The operation of the robot model is determined by Equations (7) and (8):

$$M(q)\dot{v} + C(q, \dot{q})v + Dv = \tau + P(t), \tag{7}$$

where

- $q = (x, y, \theta)^T$ is the vector of the configuration coordinates;
- $v = (v, w)^T$ is the vector of velocities;
- $\tau = (\tau_1, \tau_2)$ is the vector of torques applied to the wheels of the robot where τ_1 and τ_2 denote the torques of the right and left wheel, respectively;
- $P \in R^2$ is the uniformly bounded disturbance vector;
- $M(q) \in R^{2 \times 2}$ is the positive-definite inertia matrix;
- $C(q, \dot{q})\theta$ is the vector of centripetal and Coriolis forces;
- $D \in R^{2 \times 2}$ is a diagonal positive-definite damping matrix.

The kinematic system is determined by Equation (8):

$$\dot{q} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}, \tag{8}$$

where

- (x, y) is the position in the $X - Y$ (world) reference frame;
- θ is the angle between the heading direction and the x -axis;
- v and w are the linear and angular velocities, respectively

Equation (9) represents the non-holonomic constraint this system has, which corresponds to a non-slip wheel condition preventing the robot from moving sideways:

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0 \tag{9}$$

This controller is composed of two inputs and two outputs; the first input is the ev (error in the linear velocity), which has two trapezoidal and one triangular membership functions with linguistic variables called N, Z, and P. The second input is the ew (error in the angular velocity), which has two trapezoidal and one triangular membership functions with linguistic variables of N, Z, and P. The first output is T1 (torque 1), which has three triangular membership functions with linguistic variables of N, Z, and P. The second output is T2 (torque 2), which has three triangular membership functions with linguistic variables of N, Z, and P. This controller is of the Mamdani type and uses nine fuzzy rules, which are presented in detail in Table 4.

Table 4. Fuzzy rules for the robot mobile controller.

Rule Number	Inputs			Output	
	Ev	Operator	Ew	T1	T2
1	N	and	N	N	N
2	N	and	Z	N	Z
3	N	and	P	N	P
4	Z	and	N	Z	N
5	Z	and	Z	Z	Z
6	Z	and	P	Z	P
7	P	and	N	P	N
8	P	and	Z	P	Z
9	P	and	P	P	P

Table 4 presents in detail each of the rules that compose the robot mobile controller to achieve the main objective of this controller. Figure 11 shows the structure of the fuzzy system for this controller.

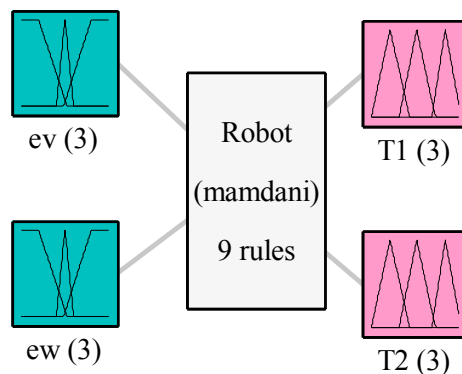


Figure 11. Structure of the mobile robot fuzzy system for control.

5. Simulation Results

This section presents the results obtained by optimizing the membership functions using the dynamic parameter adaptation with Type-1 and interval Type-2 fuzzy logic. This methodology was applied to the three benchmark problems shown in Section 4, where the objective is to follow a desired trajectory, and 30 experiments were carried out with disturbances and without disturbances with the original algorithm of harmony search (HS), Type-1 fuzzy harmony search algorithm (FHS), and interval Type-2 fuzzy harmony search (FHS2). The noise applied to each plant was at a level of 0.5 using the uniform random number generator. The objective function of the methods is the root mean square error (RMSE) of the trajectory, although other control metrics were also used: MSE (Mean Squared Error), ISE (Integral Squared error), IAE (Integral Absolute Error), ITSE (Integral of Time-weighted Squared Error) and ITAE (Integral Time-weighted Absolute Error), and SNR (Signal to noise ratio), respectively, as shown in Equations (10)–(16).

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (x_t - \hat{x}_t)^2} \tag{10}$$

$$MSE = \frac{1}{N} \sum_{K=1}^N [X(K) - Y(K)]^2 \tag{11}$$

$$I_1 = \int_0^{\infty} e^2(t) dt \tag{12}$$

$$I_2 = \int_0^{\infty} e(t) dt \tag{13}$$

$$I_3 = \int_0^{\infty} te^2(t) dt \tag{14}$$

$$I_4 = \int_0^{\infty} te(t) dt \tag{15}$$

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right) \tag{16}$$

The parameters used to perform the experiments are as follows: 100 iterations, PA rate of 0.75, HMR of 0.95 (Only for the HS), and HM of 30. Experimental results in Tables 5–7 show the best experiment without perturbation in the fuzzy logic controller for each algorithm applied to each case study (water tank controller, temperature controller, and robot mobile controller). Tables 8–10 show

the results obtained using the water tank controller, temperature controller, and robot mobile controller with noise. These results correspond to the averages of the 30 experiments for each method.

Table 5. Errors obtained from the water tank controller without noise.

Performance Index	HS	FHS	FHS2
RMSE	3.18×10^{-2}	2.61×10^{-1}	2.49×10^{-2}
MSE	4.58×10^{-2}	7.56×10^{-2}	5.75×10^{-2}
ITAE	3.87×10^4	9.93×10^5	3.68×10^4
ITSE	2.57×10^4	2.43×10^6	2.39×10^4
IAE	7.91×10	2.01×10^3	7.53×10
ISE	5.28×10	4.96×10^3	4.90×10

Table 6. Errors obtained from the temperature controller without noise.

Performance Index	HS	FHS	FHS2
RMSE	6.34×10^{-2}	6.75×10^{-2}	6.25×10^{-2}
MSE	2.44×10^{-2}	4.55×10^{-3}	4.62×10^{-3}
ITAE	7.01×10^5	7.04×10^5	7.03×10^5
ITSE	1.06×10^6	1.07×10^6	1.07×10^6
IAE	1.40×10^3	1.41×10^3	1.41×10^3
ISE	2.11×10^3	2.12×10^3	2.12×10^3

Table 7. Errors obtained from the robot mobile controller without noise.

Performance Index	HS	FHS	FHS2
RMSE	2.33×10^{-1}	1.37×10^{-1}	1.11×10^{-1}
MSE	8.56×10	5.07×10	6.06×10
ITAE	2.18×10^5	2.11×10^5	3.03×10^6
ITSE	1.05×10^5	9.44×10^4	2.86×10^7
IAE	4.38×10^2	4.24×10^2	6.09×10^3
ISE	2.11×10^2	1.90×10^2	5.72×10^4

Table 8. Errors obtained from the water tank controller with noise.

Performance Index	HS	FHS	FHS2
RMSE	3.18×10^{-2}	2.49×10^{-2}	1.32×10^{-2}
MSE	4.58×10^{-2}	5.75×10^{-2}	6.36×10^{-2}
ITAE	3.87×10^4	3.68×10^4	3.49×10^4
ITSE	2.57×10^4	2.39×10^4	2.08×10^4
IAE	7.91×10	7.53×10	7.17×10
ISE	5.28×10	4.90×10	4.29×10

Table 9. Errors obtained from the temperature controller with noise.

Performance Index	HS	FHS	FHS2
RMSE	6.34×10^{-2}	1.24×10^{-1}	6.29×10^{-4}
MSE	2.44×10^{-2}	1.54×10^{-2}	3.64×10^{-2}
ITAE	7.01×10^5	6.98×10^5	6.98×10^5
ITSE	1.06×10^6	1.05×10^6	1.05×10^6
IAE	1.40×10^3	1.39×10^3	1.40×10^3
ISE	2.11×10^3	2.10×10^3	2.10×10^3

The noise used for the two methods was at a level of 0.5 with the uniform random number 10.7128 dB for Type-1 and 12.1449 dB for Type-2 for the water tank problem.

Table 10. Errors obtained from the robot mobile controller with noise.

Performance Index	HS	FHS	FHS2
RMSE	2.33×10^{-1}	3.82×10^{-2}	3.69×10^{-2}
MSE	8.56×10	7.67×10	6.06×10
ITAE	2.18×10^5	3.09×10^6	3.10×10^6
ITSE	1.05×10^5	3.26×10^7	3.28×10^7
IAE	4.38×10^2	6.46×10^3	6.47×10^3
ISE	2.11×10^2	6.47×10^4	6.49×10^4

The noise used for the two methods was 0.5 with the uniform random number 19.9196 dB for Type-1 and 22.1073 dB for Type-2 for the temperature problem.

The noise used for the two methods was 0.5 with the uniform random number 39.6227 dB for Type-1 and 41.2338 dB for Type-2 for the robot mobile problem.

Analyzing the results obtained in Tables 5–10, we can verify an improvement in the experiments when using the Type-1 and Type-2 methods with noise. Figures 12–14 show the best trajectory found for each method out of 30 experiments using noise: Figure 12 shows the best result for the water tank controller problem, Figure 13 shows the best result for the temperature controller problem, and Figure 14 shows the best result for the robot mobile controller. It is observed that the original method is more unstable in all cases and the Type-1 and Type-2 fuzzy methods achieve better stability and tracking of the desired trajectory.

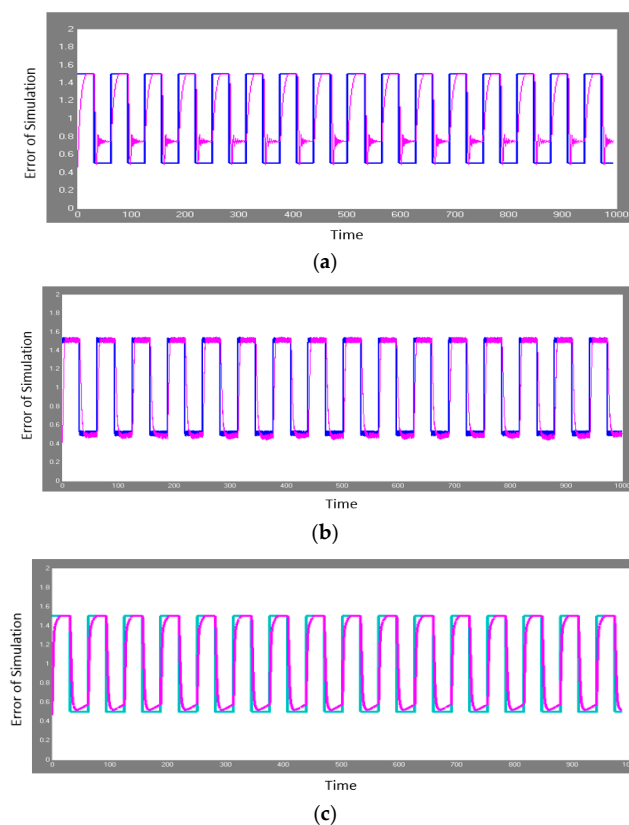


Figure 12. Simulation results of the water tank controller for the three methods. (a) The best result is shown using original harmony search algorithm (HS); (b) the best result is shown using the Type-1 fuzzy harmony search (FHS); (c) the best result is shown using the Type-2 fuzzy harmony search (FHS2). These methods were applied with noise to verify the stability of the methods. The blue line represents the desired trajectory and the pink line the obtained trajectory; the objective is for the obtained line to resemble the desired one.

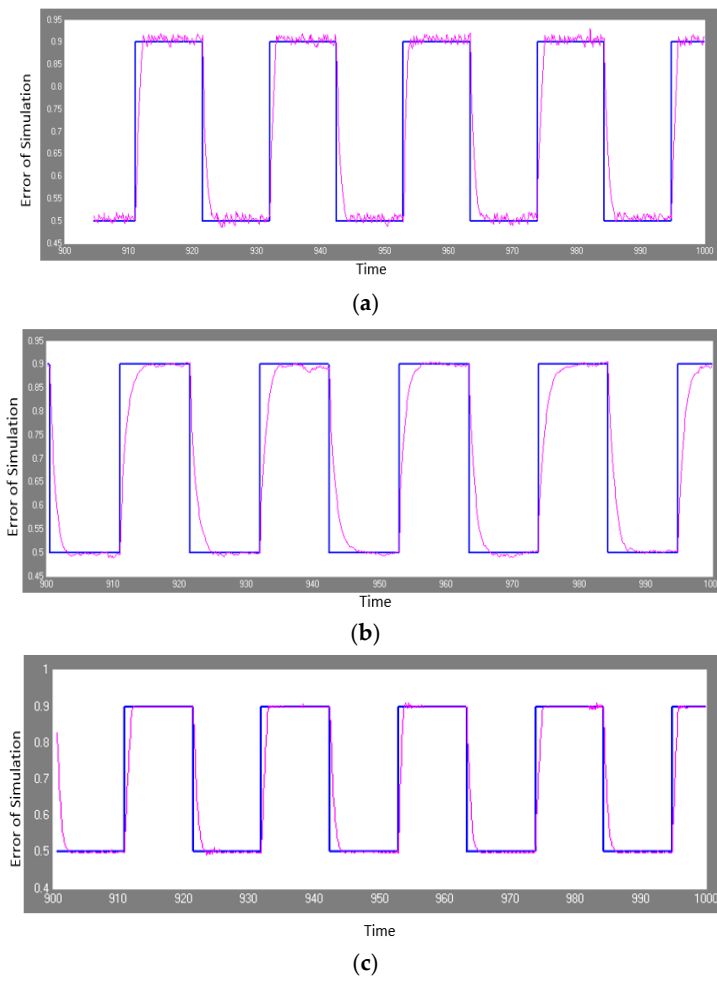


Figure 13. Simulation results of the temperature controller for the three methods. (a) The best result is shown using original harmony search algorithm (HS); (b) the best result is shown using the Type-1 fuzzy harmony search (FHS); (c) the best result is shown using the Type-2 fuzzy harmony search (FHS2). These methods were applied with noise to verify the stability of the methods. The blue line represents the desired trajectory and the pink line the obtained trajectory; the objective is for the obtained line to resemble the desired one.

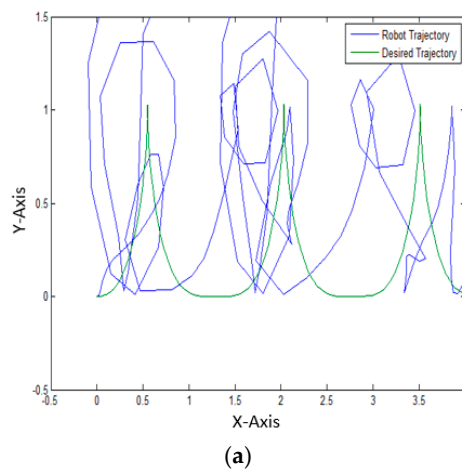


Figure 14. Cont.

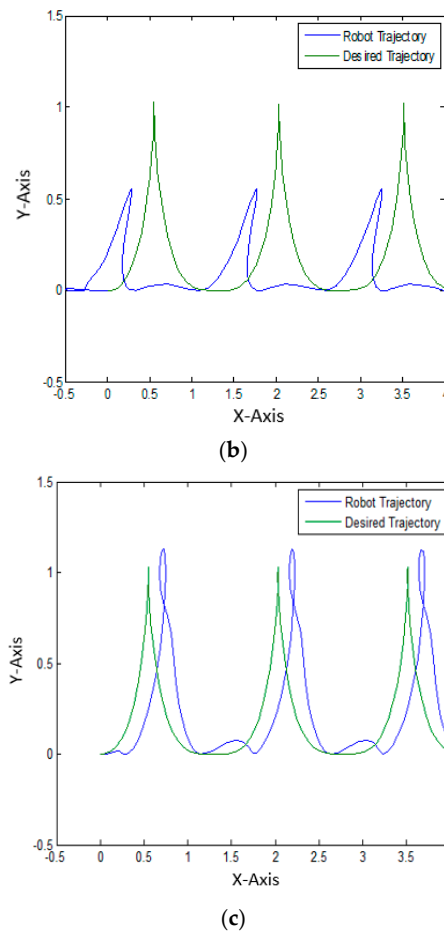


Figure 14. Simulation results of the robot mobile controller for the three methods. (a) The best result is shown using original harmony search algorithm (HS); (b) the best result is shown using the Type-1 fuzzy harmony search (FHS); (c) the best result is shown using the Type-2 fuzzy harmony search (FHS2). These methods were applied with noise to verify the stability of the methods. The green line is the desired trajectory and the blue line the obtained trajectory, and the objective is for the obtained line to resemble the desired one.

6. Statistical Comparison

This section shows a statistical comparison of the original method and the proposed methods applied to the three control problems mentioned above. The statistical test method that was applied is the z-test, and the parameters used are shown in Table 11.

The alternative hypothesis indicates that the Type-1 (FHS) and Type-2 (FHS2) fuzzy harmony search methods values are smaller than the original harmony search algorithm (HS) method and the null hypothesis indicates otherwise, with a rejection region for values lower than -1.645 . The equation of the z-test is shown as follows:

$$Z = \frac{(\overline{X}_1 - \overline{X}_2) - (u_1 - u_2)}{\sigma_{\overline{X}_1 - \overline{X}_2}} \tag{17}$$

Tables 12–17 show the values of z, “S” means that evidence of significance is found and “N.S” refers to a case in which no evidence of significance is found. In all Tables 12–17 the result of the Evidence of the first row represents the comparison between FHS and HS, the result of the Evidence of the second row represents the comparison between FHS2 and HS, and finally the result of the Evidence of the third row is the comparison between FHS and FHS2 methods.

Table 11. Parameters for the statistical z-test.

Parameter	Value
Level of Significance	95%
Alpha	0.05%
H _a	$\mu_1 < \mu_2$
H ₀	$\mu_1 \geq \mu_2$
Critical Value	-1.645

Table 12. Results for the statistical test with Type-1 and Type-2 FHS and HS without noise.

Water Tank Controller				
Method	Mean	Standard Deviation	z-Value	Evidence
HS	3.19×10^{-2}	3.13×10^{-2}	35.8789	N.S
FHS	2.60×10^{-1}	1.55×10^{-2}	-50.7296	S
FHS2	2.56×10^{-2}	1.97×10^{-2}	-1.1733	N.S

Table 13. Results for the statistical test with Type-1 and Type-2 FHS and HS without noise.

Temperature Controller				
Method	Mean	Standard Deviation	z-Value	Evidence
HS	6.34×10^{-2}	3.12×10^{-3}	6.4226	N.S
FHS	6.75×10^{-2}	1.59×10^{-3}	-14.0672	S
FHS2	6.25×10^{-2}	1.13×10^{-3}	-1.4778	N.S

Table 14. Results for the statistical test with Type-1 and Type-2 FHS and HS without noise.

Robot Mobile Controller				
Method	Mean	Standard Deviation	z-Value	Evidence
HS	2.33×10^{-1}	2.18×10^{-1}	-2.0706	S
FHS	1.37×10^{-1}	1.57×10^{-1}	-0.7959	N.S
FHS2	1.11×10^{-1}	1.37×10^{-1}	-2.7100	S

Table 15. Results for the statistical test with Type-1 and Type-2 FHS and HS with noise.

Water Tank Controller				
Method	Mean	Standard Deviation	z-Value	Evidence
HS	3.18×10^{-2}	3.13×10^{-2}	-1.0134	N.S
FHS	2.49×10^{-2}	2.01×10^{-2}	-2.5236	S
FHS2	1.32×10^{-2}	1.54×10^{-2}	-2.9164	S

Table 16. Results for the statistical test with Type-1 and Type-2 FHS and HS with noise.

Temperature Controller				
Method	Mean	Standard Deviation	z-Value	Evidence
HS	6.34×10^{-2}	3.12×10^{-3}	72.5412	N.S
FHS	1.24×10^{-1}	3.34×10^{-3}	-201.3932	S
FHS2	6.29×10^{-4}	1.07×10^{-4}	-109.8982	S

Table 17. Results for the statistical test with Type-1 and Type-2 FHS and HS with noise.

Robot Controller				
Method	Mean	Standard Deviation	z-Value	Evidence
HS	2.33×10^{-1}	2.18×10^{-1}	-4.8553	S
FHS	3.82×10^{-2}	4.22×10^{-2}	-0.2277	N.S
FHS2	3.69×10^{-2}	4.20×10^{-2}	-4.8866	S

7. Conclusions

In this paper a comparative study of dynamic parameter adaptation using the Type-1 and Type-2 fuzzy harmony search algorithm is presented. The proposed method uses fuzzy logic for dynamic parameter adaptation and was applied to three benchmark problems—the water tank, temperature, and robot mobile controller—for disturbed and undisturbed trajectory tracking. In the analysis of the results we obtain a significant improvement in the three benchmark control problems when using dynamic parameter adaptation with interval Type-2 fuzzy logic. This verifies that for complex problems the handling of uncertainty is important for obtaining better stability and therefore better results with or without noise.

It was verified that the proposed method manages to maintain the control of a reference line in the three cases presented in Section 5, and in the statistical tests it can be validated that when using dynamic parameter adaptation the optimization of the values of the membership functions of these control cases is achieved by reducing the mean quadratic error and following the desired path.

In future work, the proposed method will be tested with different types of problems or controllers to verify the efficiency of the method; we will also try to improve the fuzzy system with which the internal parameters of the algorithm are adapted to achieve improvements when applied to control problems.

Acknowledgments: We would like to express our gratitude to CONACYT, Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

Author Contributions: Fevrier Valdez contributed to the proposal of the fuzzy approach in harmony search, in addition to the discussion and analysis of the results; Patricia Melin reviewed the state of the art and analyzed the experimental results; Cinthia Peraza analyzed the original method and applied Type-1 and Type-2 fuzzy logic for dynamic parameter adaptation in harmony search for tracking trajectories, and contributed to the performed experiments.

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

References

1. Precup, R.-E.; David, R.-V.; Petriu, E.M.; Preitla, S.; Rădac, M.-B. Novel adaptive charged system search algorithm for optimal tuning of fuzzy controllers. *Exp. Syst. Appl.* **2014**, *41*, 1168–1175. [[CrossRef](#)]
2. Kiran, M.S.; Findik, O. A directed artificial bee colony algorithm. *Appl. Soft Comput.* **2015**, *26*, 454–462.
3. Osaba, E.; Onieva, E.; Diaz, F.; Diaz, F.; Carballedo, R.; Lopez, P.; Perallo, A. A migration strategy for distributed evolutionary algorithms based on stopping non-promising subpopulations: A case study on routing problems. *Int. J. Artif. Intell.* **2015**, *13*, 46–56.
4. Solos, I.P.; Tassopoulos, I.X.; Beligiannis, G.N. Optimizing shift scheduling for tank trucks using an effective stochastic variable neighbourhood approach. *Int. J. Artif. Intell.* **2016**, *14*, 1–26.
5. Geem, Z. Multiobjective optimization of water distribution networks using fuzzy theory and harmony search. *Water* **2015**, *7*, 3613–3625. [[CrossRef](#)]
6. Kar, P.; Swain, S.C. A harmony search-firefly algorithm based controller for damping power oscillations. *Comput. Intell. Commun. Technol.* **2016**, *8*, 351–355.
7. Roy, N.; Ghosh, A.; Sanyal, K. Normal boundary intersection based multi-objective harmony search algorithm for environmental economic load dispatch problem. *Power Syst.* **2016**, *10*, 1–6.
8. Gao, K.Z.; Suganthan, P.N.; Pan, Q.K.; Chua, T.J.; Cai, T.X.; Chong, C.S. Discrete harmony search algorithm for flexible job shop scheduling problem with multiple objectives. *J. Intell. Manuf.* **2016**, *27*, 363–374. [[CrossRef](#)]
9. Alia, O.M. Dynamic relocation of mobile base station in wireless sensor networks using a cluster-based harmony search algorithm. *Inf. Sci.* **2017**, *385*, 76–95. [[CrossRef](#)]
10. 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](#)]
11. Gonzalez, C.I.; Castro, J.R.; Mendoza, O.; Melin, P.; Castillo, O. Optimization by cuckoo search of interval Type-2 fuzzy logic systems for edge detection. In *Recent Developments and New Direction in Soft-Computing Foundations and Applications*; Springer: Basel, Switzerland, 2016; pp. 141–154.

12. Castillo, O.; Ochoa, P.; Soria, J. Differential Evolution with Fuzzy Logic for Dynamic Adaptation of Parameters in Mathematical Function Optimization. In *Imprecision and Uncertainty in Information Representation and Processing*; Springer: Basel, Switzerland, 2016; pp. 361–374.
13. Bernal, E.; Castillo, O.; Soria, J.; Valdez, F. Imperialist Competitive Algorithm with Dynamic Parameter Adaptation Using Fuzzy Logic Applied to the Optimization of Mathematical Functions. *Algorithms* **2017**, *10*, 18. [[CrossRef](#)]
14. Caraveo, C.; Valdez, F.; Castillo, O. Optimization mathematical functions for multiple variables using the algorithm of self-defense of the plants. In *Nature-Inspired Design of Hybrid Intelligent Systems*; Springer: Basel, Switzerland, 2017; pp. 631–640.
15. 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](#)]
16. Peraza, C.; Valdez, F.; Garcia, M.; Melin, P.; Castillo, O. A new fuzzy harmony search algorithm using fuzzy logic for dynamic parameter adaptation. *Algorithms* **2016**, *9*, 69. [[CrossRef](#)]
17. Peraza, C.; Valdez, F.; Castillo, O. An adaptive fuzzy control based on harmony search and its application to optimization. In *Nature-Inspired Design of Hybrid Intelligent Systems*; Springer: Basel, Switzerland, 2017; pp. 269–283.
18. Peraza, C.; Valdez, F.; Castillo, O. Interval Type-2 fuzzy logic for dynamic parameter adaptation in the harmony search algorithm. *Intell. Syst.* **2016**, *10*, 106–112.
19. Mahdavi, M.; Fesanghary, M.; Damangir, E. An improved harmony search algorithm for solving optimization problems. *Appl. Math. Comput.* **2007**, *188*, 1567–1579. [[CrossRef](#)]
20. Zadeh, L.A. Fuzzy sets. *Inf. Control.* **1965**, *8*, 338–353. [[CrossRef](#)]
21. Zadeh, L.A. Fuzzy logic. *Computer* **1988**, *21*, 83–93. [[CrossRef](#)]
22. Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning—I. *Inf. Sci.* **1975**, *8*, 199–249. [[CrossRef](#)]
23. 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](#)]
24. Lee, K.S.; Geem, Z.W. A new structural optimization method based on the harmony search algorithm. *Comput. Struct.* **2004**, *82*, 781–798. [[CrossRef](#)]
25. Geem, Z.W. State-of-the-art in the structure of harmony search algorithm. In *Recent Advances in Harmony Search Algorithm*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–10.
26. Geem, Z.W.; Choi, J.-Y. Music composition using harmony search algorithm. In *Applications of Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 593–600.
27. Manjarres, D.; Landa-Torres, I.; Gil-Lopez, S.; Sera, J.D.; Bilbaob, M.N.; Salcedo-Sanz, S.; Geem, Z.W. survey on applications of the harmony search algorithm. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1818–1831. [[CrossRef](#)]
28. Alia, O.M.; Mandava, R. The variants of the harmony search algorithm: An overview. *Artif. Intell. Rev.* **2011**, *36*, 49–68. [[CrossRef](#)]
29. Omran, M.G.H.; Mahdavi, M. Global-best harmony search. *Appl. Math. Comput.* **2008**, *198*, 643–656. [[CrossRef](#)]
30. Wang, C.-M.; Huang, Y.-F. Self-adaptive harmony search algorithm for optimization. *Expert Syst. Appl.* **2010**, *37*, 2826–2837. [[CrossRef](#)]
31. Melin, P.; Astudillo, L.; Castillo, O.; Valdez, F.; Garcia, M. 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.* **2013**, *40*, 3185–3195. [[CrossRef](#)]

