

Article

Generalized Type 2 Fuzzy Differential Evolution Applied to a Sugeno Controller

Patricia Ochoa, Cinthia Peraza , Oscar Castillo  and Patricia Melin * 

Tijuana Institute of Technology, TecNM, Tijuana C.P. 22379, Mexico

* Correspondence: pmelin@tectijuana.mx

Abstract: The papers using fuzzy logic have generated continuous improvements for applications, and an example of this is the use of generalized type 2 fuzzy systems in real-world problems. The key idea of this paper is to present a generalized type 2 fuzzy system for augmenting differential evolution with dynamic parameter variation in order to enhance its performance and convergence. A generalized type 2 Sugeno controller was implemented with the aim of enhancing the trajectory of a robot, and an analysis of the execution time and the errors obtained by ITAE, IAE, ITSE, ISEV, RMSE, and MSE is presented. In addition, a comparison with different levels of disturbance applied to the controller was performed with the goal of demonstrating the efficiency of a type 2 fuzzy system.

Keywords: type 2 fuzzy system; differential evolution; Sugeno controller

MSC: 03B52; 03E72; 62P30

1. Introduction

In our day-to-day life, we are faced with situations where we have to make decisions under uncertainty. Fuzzy logic offers a reasoning method that resembles human reasoning that can help manage these situations. This approach is similar to how humans perform decision-making and some current relevant related works are mentioned as follows.

In the energy sector, fuzzy logic plays a vital role in predicting and controlling the performance delivered by energy conversion systems [1]. To increase the packet arrival rate and reduce the end-to-end delay, an intelligent multi-attribute routing scheme is proposed by employing fuzzy logic and design a technique of order preference by similarity to ideal solutions (TOPSIS) algorithm [2]. Another article presents a comprehensive review of almost all existing applications of fuzzy logic in the drying technology to solve problems of nonlinear modeling, simulation, pattern recognition, clustering, classification, control, and optimization [3]. A book considered for this paper reviewed the basic concepts of type 2 fuzzy systems and then described the proposed definitions for interval type 3 fuzzy sets and relations, as well as inference in interval type 3 systems [4]. Another important book that was taken into account presented the recent developments in fuzzy logic, neural networks, and optimization algorithms, as well as their hybrid combinations for solving complex problems [5].

Fuzzy theory and fuzzy systems have made significant progress in many applications since their inception. Fuzzy logic is becoming an extremely suitable and satisfactory tool for the development of real-time applicable expert systems in the medical science; see, for example, a paper presenting the diversity of works on fuzzy logic to identify the risk of various diseases and also revealing a model for the direct service of patients from old age homes to intensive care units (ICU) through ambulance services [6]. A fuzzy logic-based optimal network system for the delivery of medical goods via drones and land transport in remote areas is presented in [7]. There are several hybrid algorithms, such as neuro-fuzzy systems, or adaptive genetic algorithms with fuzzy logic models that have been previously



Citation: Ochoa, P.; Peraza, C.; Castillo, O.; Melin, P. Generalized Type 2 Fuzzy Differential Evolution Applied to a Sugeno Controller. *Axioms* **2023**, *12*, 156. <https://doi.org/10.3390/axioms12020156>

Academic Editor: Ioannis G. Tsoulos

Received: 24 December 2022

Revised: 27 January 2023

Accepted: 29 January 2023

Published: 2 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

deployed for developing a clinical decision support system for oncological, psychiatric, and cardiac-based diseases. This work highlights the potential of fuzzy logic systems and their hybrid intelligent models to develop support systems for various types of cancer, including their limitations and future prospects [8]. The combination of the hydraulic hybrid vehicle (HHV) architecture with optimum fuzzy logic control can result in reducing the level of emissions, as presented in [9]. Finally, we can mention the use of fuzzy logic in the problems of test control of a bypass turbojet engine gas generator [10].

In the field of engineering, numerous industrial applications of fuzzy logic have achieved good results. Control is a very common concept and is widely used by many people in everyday life. The term is usually used to refer to the interaction between man and the environment, more specifically to man–machine interaction. The fuzzy logic augmentation of control systems gives rise to what we call fuzzy controllers. Within control systems, there are two large areas, modeling or identification and control itself or direct control, and some works of interest (related to the work presented in this paper) cover the interval type 2 fuzzy model subject to nonperiodic denial of service (DoS) attacks [11], a fuzzy logic control strategy for a four-wheel independent-drive electric vehicle [12], a type 1 and type 2 fuzzy-based harmony search applied to a two-tank level control and a DC motor angular position control [13]; a classical PID controller and a fuzzy PID were developed for a photovoltaic system in [14]; a solution to the control problem for a class of multiple-input–multiple-output (MIMO) underactuated systems subject to plant uncertainties and actuator dead zones was presented in [15].

The goal of making a system follow desired reference signals is one of the fundamental problems in the control area. Therefore, it is a subject that has been studied by many researchers and to resolve which different methods have been designed.

In the last decade, there has been sustained growth in the field of metaheuristics for search and optimization. Such methods, due to their capability of finding very-close-to-optimal solutions within a reasonable period, have started to be used to solve different engineering problems; an example of these methods is the differential evolution (DE) algorithm.

For this work, general type 2 fuzzy systems were used, which are an alternative to traditional (type 1) fuzzy systems where type 2 can really handle uncertainty. There are currently numerous works that use general type 2 fuzzy systems in different areas, and we mention some of the works that are considered novel and important for our work.

General type 2 fuzzy logic systems (GT2 FLSs) provide a more flexible way of overcoming an uncertain lack of uniformity in different applications. Centroid-type reduction is one of the major components of GT2 FLSs, and it is currently one of the key factors restricting the GT2 FLS efficiency for real-time solutions [16]. A study on weight-based discrete noniterative algorithms for computing the centroids of general type 2 fuzzy sets is presented in [17]. The design of discrete noniterative algorithms for the center-of-sets-type reduction of general type 2 fuzzy logic systems can be found in [18]. A unified general type 2 fuzzy PID (UGT2-FPID) controller using the upper and lower bounds of one α -plane is proposed in [19]. The UGT2-FPID controller contains another two adjustment parameters, and the analytical structure of a UGT2-FPID controller is obtained by adapting an input combination method [19]. The application of general type 2 fuzzy systems in the dynamic adjustment of the parameters of a recent nature-based metaheuristic that follows the rules of feeding strategies of predators and prey in ecosystems is found in [20]. A medical diagnosis application using type 2 fuzzy logic and the Ant Lion Optimizer to classify blood pressure levels is shown in [21]. One of the most studied application areas of intelligent systems is the classification area, and this is because classification covers a wide range of real-world problems. Some examples are fault diagnosis, image segmentation, and medical diagnosis, among others. In most cases, the intelligent systems designed for the solution of this kind of problems are based on supervised learning, which is based on learning how to classify with previous datasets for finding relations between the inputs and outputs [22] and improving the differential evolution algorithm with the utilization of shadowed and general type 2 fuzzy systems to dynamically adapt one of the parameters of

the evolutionary method [23]. As a difference from previous works, the contribution of this paper is the use of general type 2 fuzzy systems for parameter adaptation in differential evolution and its application to design a Sugeno controller.

This work comprises the following sections: In Section 2, the general type 2 fuzzy systems theory is reviewed. Section 3 summarizes the differential evolution terminology. Section 4 describes a parametric general type 2 fuzzy model. Section 5 comprises the experimentation, and in Section 6, the conclusions are outlined.

2. General Type 2 Fuzzy Sets

A general type 2 fuzzy system (GT2FS) allows uncertainty to be modeled more efficiently [24]. Equation (1) postulates a generalized (or general) type 2 fuzzy set:

$$\tilde{A} = \{(x, u), \mu_{\tilde{A}}(x, u) | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \tag{1}$$

where X is the universe for the primary variable of \tilde{A} , x . The 3D membership function is usually denoted by $\mu_{\tilde{A}}(x, u)$, where $x \in X$ and $u \in U \subseteq [0, 1]$ and $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$ is continuous, and it can be expressed as Equation (2):

$$\tilde{A} = \left\{ \int_{x \in X} \int_{u \in [0,1]} \mu_{\tilde{A}}(x, u) / (x, u) \right\} \tag{2}$$

where $\int \int$ denotes the union over all admissible x and u . For discrete universes of discourse, \tilde{A} can be denoted by Equation (3):

$$\tilde{A} = \left\{ \sum_{x \in X} \sum_{u \in [0,1]} \mu_{\tilde{A}}(x, u) / (x, u) \right\} \tag{3}$$

In Equations (1)–(3), u is called the secondary variable, and has the domain $U = [0, 1]$, where $x \in X$. When $\mu_{\tilde{A}}(x, u)$ for $\forall x \in X$ and $\forall u \in U$, then \tilde{A} is called an interval type 2 fuzzy set (IT2 FS).

At each point of x , e.g., $x = x'$, the 2D plane whose axes are u and $\mu_{\tilde{A}}(x, u)$ is called a vertical slice of $\mu_{\tilde{A}}(x, u)$. A secondary membership function is a vertical slice of $\mu_{\tilde{A}}(x, u)$ by Equation (4):

$$\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}^u \subseteq [1,0]} f_x(u) / u \tag{4}$$

where $J_{x'}^u$ is the subset of U that is the support of $\mu_{\tilde{A}}(x')$ and is called the primary membership of \tilde{A} . The amplitude of the secondary membership function $f_x(u)$ is called the secondary grade or the secondary set.

The two-dimensional support of $\mu_{\tilde{A}}(x, u)$ is called the footprint of uncertainty (FOU) of \tilde{A} by Equation (5):

$$FOU(\tilde{A}) = \{(x, u) \in X \times [0, 1] | \mu_{\tilde{A}}(x, u) > 0\} \tag{5}$$

$FOU(\tilde{A})$ can also be expressed as the union of all primary memberships expressed in Equation (6):

$$FOU(\tilde{A}) = \cup_{x \in X} J_x^u \tag{6}$$

An α -plane for the GT2FS \tilde{A} denoted by \tilde{A}_α represented by Equation (7) is the union of all primary membership functions of \tilde{A} whose secondary grades are greater than or equal to α ($0 \leq \alpha \leq 1$). For performing the inference in the GT2FS, the α -planes representation was used.

$$\tilde{A}_\alpha = \{(x, u), \mu_{\tilde{A}}(x, u) \geq \alpha | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} = \int_{\forall x \in X} \int_{\forall u \in [0,1]} \{(x, u) | f_x(u) \geq \alpha\} \tag{7}$$

The union of all α -planes is presented in Equation (8), where $R_{\tilde{A}_\alpha}$ is one horizontal slice at level α :

$$\tilde{A}_\alpha = \cup_{\alpha \in [0,1]} R_{\tilde{A}_\alpha} \tag{8}$$

The structure of rules in a GT2FS is represented for a type 2 fuzzy system with p inputs $x_1 \in X_1, \dots, x_p \in X_p$ and one output $y \in Y$, multiple input, single output, if we assume there are M rules and the k th rule in the GT2FS, as described in Equation (9):

$$R^k : \text{IF } x_1 \text{ is } \tilde{F}_1^k \text{ and } \dots x_p \text{ is } \tilde{F}_p^k, \text{ THEN } y \text{ is } \tilde{G}^k \tag{9}$$

The centroid defuzzification process in a GT2FS is expressed in Equation (10), the centroid of the GT2FS \tilde{A} can be obtained by taking the union of the centroids of all the α -planes of \tilde{A} , and then the Karnik–Mendel algorithm can be used for computing the centroid of each α -plane, where $[\alpha_{y_l}, \alpha_{y_r}]$ is the domain of the centroid:

$$Y_c(\alpha) = \text{Centroid}(\tilde{A}(\alpha)) = \int_{u_1 \in \alpha_{J_{x_1}}} \dots \int_{u_N \in \alpha_{J_{x_N}}} \alpha \frac{\sum_{i=1}^N x_i \mu_A(x_i)}{\sum_{i=1}^N \mu_A(x_i)} = \alpha / [\alpha_{y_l}, \alpha_{y_r}] \tag{10}$$

The general type 2 fuzzy inference is approximated using α -planes; for each α -plane, a centroid-type reducer is applied with Equation (10); after that, the results of the α -planes are integrated by Equations (11) and (12). Finally, the defuzzifier output is obtained by using the average of y^l and y^r as presented in Equation (13):

$$\hat{y}_j^l(x') = \frac{\sum_{i=1}^N \alpha_i^{\alpha_i} y_j^l(x')}{\sum_{i=1}^N \alpha_i} \tag{11}$$

$$\hat{y}_j^r(x') = \frac{\sum_{i=1}^N \alpha_i^{\alpha_i} y_j^r(x')}{\sum_{i=1}^N \alpha_i} \tag{12}$$

$$\hat{y}_j(x') = \frac{\hat{y}_j^l(x') + \hat{y}_j^r(x')}{2} \tag{13}$$

3. DE Algorithm

This DE [25] is a direct stochastic parallel-search metaheuristic technique of optimization based on the populations of numerical vectors. Among the most outstanding advantages of this algorithm are simplicity, efficiency, local search property, and speed. The process undertaken by DE to solve an optimization problem is characterized by iterations on a population of vectors to evolve possible solutions based on a fitness function [26–32].

The process of an evolutionary algorithm is made up of the following components: (1) presentation of solutions to the problem; an evaluation function to obtain the fitness or survivability of individuals; (3) initialization of the initial population; (4) selection processes; (5) variation and replacement operators.

The mathematical structure of the DE algorithm is represented by the Equations below.

3.1. Population Structure (NP)

DE’s most versatile implementation maintains a pair of vector populations, both of which contain Np D-dimensional vectors of real-valued parameters. The current population, symbolized by P_x , is composed of those vectors $x_{i,g}$ that have already been found to be acceptable either as initial points or by comparison with other vectors:

$$P_{x,g} = (x_{i,g}), i = 0, 1, 2, \dots, Np - 1, g = 0, 1, 2, \dots, g_{max}, \tag{14}$$

$$x_{i,g} = (x_{j,i,g}), j = 0, 1, 2, \dots, D - 1 \tag{15}$$

Indices start with 0 to simplify working with arrays and modular arithmetics. The index, $g = 0, 1, \dots, g_{max}$, indicates the generation to which the vector belongs. In addition,

each vector is assigned a population index i , which runs from 0 to $Np-1$. Parameters within vectors are indexed with j , which runs from 0 to $D-1$. Once initialized, DE mutates randomly chosen vectors to produce an intermediary population $P_{v,g}$ of Np mutant vectors $v_{i,g}$:

$$P_{v,g} = (v_{i,g}), i = 0, 1, 2, \dots, Np - 1, g = 0, 1, 2, \dots, g_{max}, \tag{16}$$

$$v_{i,g} = (v_{j,i,g}), j = 0, 1, 2, \dots, D - 1 \tag{17}$$

Each vector in the current population is recombined with a mutant vector to produce a trial population P_u of Np , mutant vector $u_{i,g}$:

$$P_{u,g} = (u_{i,g}), i = 0, 1, 2, \dots, Np - 1, g = 0, 1, 2, \dots, g_{max}, \tag{18}$$

$$u_{i,g} = (u_{j,i,g}), j = 0, 1, 2, \dots, D - 1 \tag{19}$$

where:

- Px = current population;
- i = index population;
- g_{max} = maximum number of iterations;
- j = parameters within the vector.

During recombination, trial vectors overwrite the mutant population, so a single array can hold both populations.

3.2. Initialization

Before the population can be initialized, both upper and lower bounds for each parameter must be specified. These 2D values can be collected into two D-dimensional initialization vectors b_L and b_U , for which subscripts L and U indicate the lower and upper bounds, respectively. Once initialization bounds have been specified, a random number generator assigns each parameter of every vector a value from within the prescribed range. For example, the initial value ($g = 0$) of the j^{th} parameter of the i^{th} vector is

$$x_{j,i,0} = rand_j(0,1) \cdot (b_{j,U} - b_{j,L}) + b_{j,L} \tag{20}$$

The random number generator $rand_i(0,1)$ returns a uniformly distributed random number from within the range (0,1), i.e., $0 \leq rand_i(0,1) < 1$. The subscript j indicates that a new random value is generated for each parameter. Even if a variable is discrete or integral, it should be initialized with a real value since DE internally treats all variables as floating-point values regardless of their type.

3.3. Mutation (F)

In particular, the differential mutation uses a random sample equation to combine three different vectors chosen randomly and, in this way, create a mutant vector:

$$v_{i,g} = x_{r_0,g} + F \cdot (x_{r_1,g} - x_{r_2,g}) \tag{21}$$

The scale factor $F \in (0,1)$ is a positive real number that controls the rate at which the population evolves.

3.4. Crossover (CR)

To complement the differential mutation search strategy, DE also uses uniform crossover. Sometimes this is known as discrete recombination. In particular, DE crosses each vector with a mutant vector as indicated by the following expression:

$$u_{i,g} = u_{j,i,g} \begin{cases} u_{j,i,g} & \text{if } (rand_j(0,1) \leq Cr \text{ or } j = j_{rand}) \\ x_{j,i,g} & \text{otherwise} \end{cases} \tag{22}$$

3.5. Selection

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

$$x_{i,g+1} = \{u_{i,g} \text{ if } f(u_{i,g}) \leq f(x_{i,g}) \text{ } x_{i,g} \text{ otherwise} \tag{23}$$

The process of mutation, recombination and selection are repeated until the optimum is found, or a pre-specified terminating criterion is satisfied.

The process of mutation, recombination, and selection is repeated until the optimum is found or termination of the pre-criteria specified is satisfied. DE is a simple but powerful search engine that simulates natural evolution combined with a mechanism to generate multiple search directions based on the distribution of solutions in the current population. Each vector i in the population at generation G $x_{i,g}$ called at this moment of reproduction as the target vector will be able to generate one offspring called trial vector ($u_{i,g}$). This trial vector is generated as follows: First of all, a search direction is defined by calculating the difference between a pair of vectors r_0 and r_1 called “differential vectors”, both chosen at random from the population. This difference vector is also scaled by using a user defined parameter $F \geq 0$. This scaled difference vector is then added to a third vector r_2 called the base vector. As a result, a new vector is obtained, known as the mutation vector. After that, this mutation vector is recombined with the target vector (also called the parent vector) by using discrete recombination (usually binomial crossover) controlled by a crossover parameter $0 \leq CR \leq 1$ whose value determines how similar the trial vector will be with respect to the target vector.

Algorithm 1 illustrates the pseudocode of the original DE algorithm.

Algorithm 1. DE algorithm.

Generate the initial population of individuals

Do

For each individual j in the population

 Choose three numbers x_1, x_2 and x_3 that is, $1 \leq x_0, x_1, x_2 \leq N$ with $x_0 \neq x_1 \neq x_2 \neq j$

 Generate a random integer $i_{rand} \in (1, N)$

For each parameter i

 Calculate generation and diversity using Equations (4) and (5)

 Use a fuzzy system to calculate the new Mutation and Crossover parameters

$$v_{i,g} = x_{r_0,g} + F \cdot (x_{r_1,g} - x_{r_2,g})$$

$$u_{i,g} = u_{j,i,g} \begin{cases} v_{j,i,g} & \text{if } (rand_j(0,1) \leq Cr \text{ or } j = j_{rand}) \\ x_{j,i,g} & \text{otherwise} \end{cases}$$

End For

 Replace $x_{j,i,g}$ with the child $u_{i,g}$ if $u_{i,g}$ is better

End For

Until the termination condition is achieved

4. Parametric Generalized Type 2 Fuzzy Model

Differential evolution has been used in different studies. For example, previous studies were carried out, where the differential evolution algorithm was combined with type 1 fuzzy logic, interval type 2 fuzzy logic, and generalized type 2 fuzzy logic, using benchmark functions and control problems. This has been performed in different ways for the differential evolution algorithm, where the mutation (F) and crossover (CR) parameters are made dynamic in a single fuzzy system, and in other works, they were implemented separately [33–36]. Based on these works, we can find that the F parameter is the one with more impact on the performance of DE.

The F parameter is fixed across generations in the traditional DE algorithm; therefore, the main contribution of this article is to use the theory of a generalized type 2 fuzzy inference system to dynamically adapt this parameter as generations progress. The proposed method is called a generalized type 2 system in a differential evolution algorithm (GT2FDE) in order to control its exploitation phase. Then, the GT2FDE algorithm is utilized to enhance the performance of a Sugeno generalized type 2 controller. Figure 1 represents the diagram of the proposal.

The fuzzy system is constructed with one antecedent, which is the percentage of generations, which is calculated with Equation (24), and one consequent, which is the F parameter value calculated with Equation (25):

$$\text{Generations} = \frac{\text{Current generation}}{\text{Maximum number of generations}} \tag{24}$$

$$F = \frac{\sum_{i=1}^{r_F} \mu_i^F (F_{1i})}{\sum_{i=1}^{r_F} \mu_i^F} \tag{25}$$

where mutation is F ; r_F is the number of rules of the fuzzy system corresponding to F ; F_{1i} is the consequent result for rule i corresponding to F ; μ_i^F is the membership function of rule i corresponding to F .

Algorithm 2 illustrates the pseudocode of the proposed GT2FDE algorithm.

The differential evolution algorithm (DE) is based on the maximum number of generations searches for a desirable solution in most of the search process so that the algorithm can maintain infeasible solutions and thus avoid getting stuck in some feasible local optima.

Equation (26) shows the parametrization of a primary triangular generalized type 2 membership function with a Gaussian secondary membership function (*trigausstype2*); the a_1, b_1 , and c_1 parameters are for the upper MFs, and a_2, b_2 , and c_2 are for the lower MFs:

$$\begin{aligned} \mu(x, u) &= \text{trigausstype2}(x, u, [a_1, b_1, c_1, a_2, b_2, c_2, \rho]) \\ \mu(x, u) &= \exp\left[-\frac{1}{2}\left(\frac{u-p_x}{\sigma_u}\right)^2\right], \text{ where} \\ \mu_1(x) &= \max\left(\min\left(\frac{x-a_1}{b_1-a_1}, \frac{c_1-x}{c_1-b_1}\right), 0\right) \text{ and} \\ \mu_2(x) &= \max\left(\min\left(\frac{x-a_2}{b_2-a_2}, \frac{c_2-x}{c_2-b_2}\right), 0\right) \\ \bar{\mu}(x) &= \begin{cases} \max(\mu_1(x), \mu_2(x)) & \forall x \notin (b_1, b_2) \\ 1 & \forall x \in (b_1, b_2) \end{cases} \\ \underline{\mu}(x) &= \min(\mu_1(x), \mu_2(x)) \\ p_x &= \max\left(\min\left(\frac{x-a_x}{b_x-a_x}, \frac{c_x-x}{c_x-b_x}\right), 0\right), \\ \text{where } a_x &= \frac{a_1+a_2}{2}, b_x = \frac{b_1+b_2}{2}, c_x = \frac{c_1+c_2}{2} \\ \delta &= \bar{\mu}(x) - \underline{\mu}(x) \\ \sigma_u &= \frac{1+\rho}{2\sqrt{3}} \delta \end{aligned} \tag{26}$$

where ρ is fraction of uncertainty of the support secondary MF and δ is the support of the triangular or trapezoidal MFs.

The parameterization of the generalized type 2 membership functions (GT2MFs) utilized in the method is illustrated in Table 1.

Table 1 summarizes the equations for building the GT2MFs of the antecedents and consequents of the type 2 system.

The GT2FDE system is comprised of an antecedent that represents the generations and a consequent that represents the F parameter, where both have the linguistic values of low, medium, and high.

The GT2FS responsible for adjusting the F parameter in the DE algorithm is a Mamdani model with the following listed fuzzy rules:

- R1: If generation is low, then F is high;
- R2: If generation is medium, then F is medium;
- R3: If generation is high, then F is low.

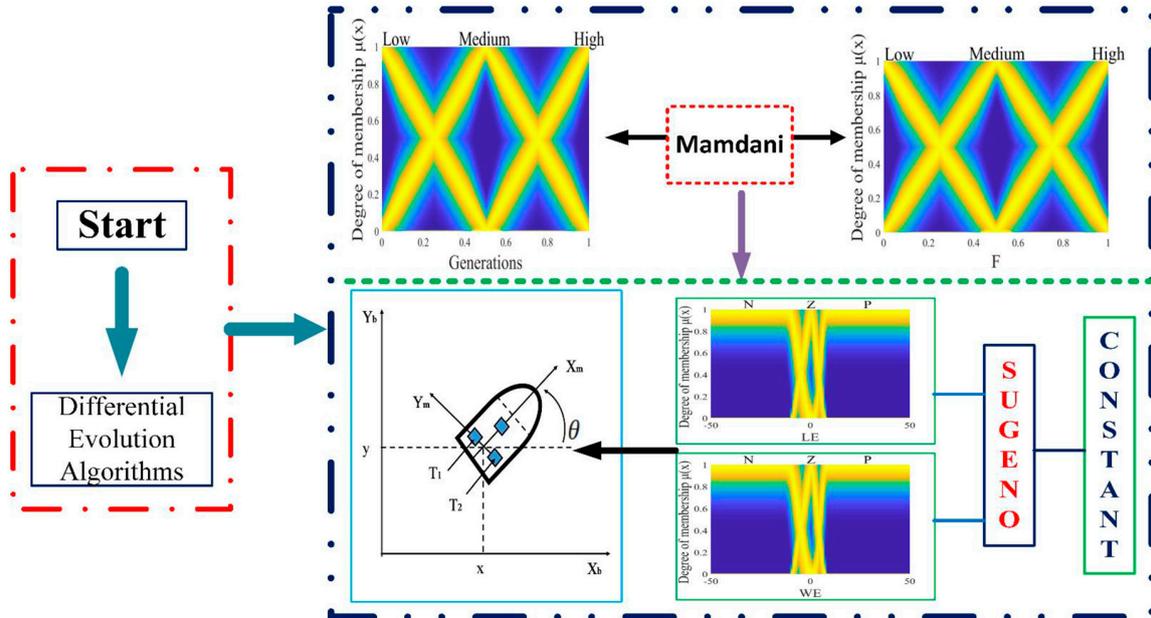


Figure 1. Diagram of the general proposal with general type 2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type 2 and type 1 fuzzy systems.

Algorithm 2. GT2FDE algorithm.

Generate the initial population of individuals

Do

For each individual j in the population

Choose three numbers x_1, x_2 and x_3 that is, $1 \leq x_0, x_1, x_2 \leq N$ with $x_0 \neq x_1 \neq x_2 \neq j$

Generate a random integer $i_{rand} \in (1, N)$

For each parameter i

Calculate generation using Equations (24)

Use a fuzzy system to calculate the new Mutation parameter using Equations (25)

$$v_{i,g} = x_{r_{0,g}} + F \cdot (x_{r_{1,g}} - x_{r_{2,g}})$$

$$u_{i,g} = u_{j,i,g} \begin{cases} v_{j,i,g} & \text{if } (rand_j(0,1) \leq Cr \text{ or } j = j_{rand}) \\ x_{j,i,g} & \text{otherwise} \end{cases}$$

End For

Replace $x_{j,i,g}$ with the child $u_{i,g}$ if $u_{i,g}$ is better

End For

Until the termination condition is achieved

Table 1. GT2MFs for the antecedent and consequent variables.

Linguistic Variable	Equation
Low	$\mu_1(x) = \max\left(\min\left(\frac{x+0.5807}{-0.0806+0.5807}, \frac{0.4193-x}{0.4193+0.0806}\right), 0\right)$ $\mu_2(x) = \max\left(\min\left(\frac{x+0.4141}{0.0859+0.4141}, \frac{0.5859-x}{0.5859-0.0859}\right), 0\right)$ $\bar{\mu}(x) = \{ \max(\mu_1(x), \mu_2(x)) \forall x \notin (-0.0868, 0.0859) \ 1 \forall x \in (-0.0868, 0.0859)$ $\underline{\mu}(x) = \min(\mu_1(x), \mu_2(x))$ $p_x = \max\left(\min\left(\frac{x-a_x}{b_x-a_x}, \frac{c_x-x}{c_x-b_x}\right), 0\right), \text{ where}$ $a_x = \frac{-0.5807-0.4141}{2}, b_x = \frac{-0.0806-0.0859}{2}, c_x = \frac{0.4193+0.5859}{2}$ $\delta = \bar{\mu}(x) - \underline{\mu}(x)$ $\sigma_u = \frac{1+\rho}{2\sqrt{3}}\delta + \varepsilon,$ $\text{where } \rho = 0.5$
Medium	$\mu_1(x) = \max\left(\min\left(\frac{x+0.0833}{0.4167+0.0833}, \frac{0.9167-x}{0.9167-0.4167}\right), 0\right)$ $\mu_2(x) = \max\left(\min\left(\frac{x-0.0833}{0.5833-0.0833}, \frac{1.083-x}{1.083-0.5833}\right), 0\right)$ $\bar{\mu}(x) = \{ \max(\mu_1(x), \mu_2(x)) \forall x \notin (0.4167, 0.5833) \ 1 \forall x \in (0.4167, 0.5833)$ $\underline{\mu}(x) = \min(\mu_1(x), \mu_2(x))$ $p_x = \max\left(\min\left(\frac{x-a_x}{b_x-a_x}, \frac{c_x-x}{c_x-b_x}\right), 0\right), \text{ where}$ $a_x = \frac{-0.0833+0.0833}{2}, b_x = \frac{0.4167+0.5833}{2}, c_x = \frac{0.9167+1.083}{2}$ $\delta = \bar{\mu}(x) - \underline{\mu}(x)$ $\sigma_u = \frac{1+\rho}{2\sqrt{3}}\delta + \varepsilon,$ $\text{where } \rho = 0.5$
High	$\mu_1(x) = \max\left(\min\left(\frac{x-0.422}{0.922-0.422}, \frac{1.422-x}{1.422-0.922}\right), 0\right)$ $\mu_2(x) = \max\left(\min\left(\frac{x-0.5886}{1.088-0.5886}, \frac{1.588-x}{1.588-1.088}\right), 0\right)$ $\bar{\mu}(x) = \{ \max(\mu_1(x), \mu_2(x)) \forall x \notin (0.922, 1.088) \ 1 \forall x \in (0.922, 1.088)$ $\underline{\mu}(x) = \min(\mu_1(x), \mu_2(x))$ $p_x = \max\left(\min\left(\frac{x-a_x}{b_x-a_x}, \frac{c_x-x}{c_x-b_x}\right), 0\right), \text{ where}$ $a_x = \frac{0.422+0.5886}{2}, b_x = \frac{0.922+1.088}{2}, c_x = \frac{1.422+1.588}{2}$ $\delta = \bar{\mu}(x) - \underline{\mu}(x)$ $\sigma_u = \frac{1+\rho}{2\sqrt{3}}\delta + \varepsilon,$ $\text{where } \rho = 0.5$

Figure 2 shows the surface of the GT2FS.

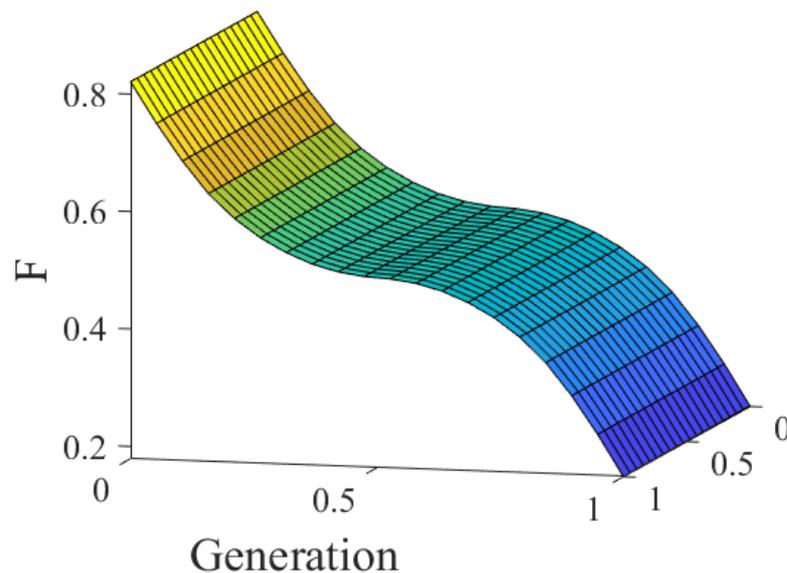


Figure 2. Surface of the GT2FS.

5. General Type 2 Fuzzy Controller

Recently, one of the most used control cases to test the efficiency of some method has been that of trajectory-following of a mobile robot [37]. The mathematical model of this robot can be found in [38,39]. In this article, the implementation of a general type 2 controller was utilized to test the efficiency of the GT2FDE method, with the aim of validating its performance. Figure 3 represents a graphical illustration of the mobile robot model.

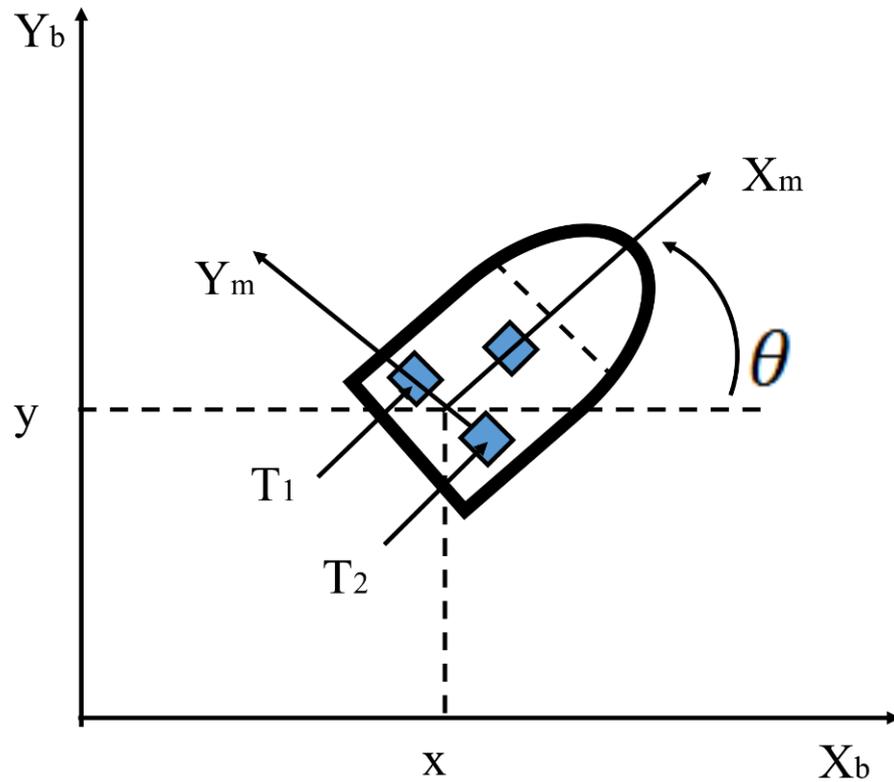


Figure 3. Illustration of the mobile robot model.

The operation of the robot model is determined by Equations (27) and (28):

$$M(q)\dot{v} + C(q, \dot{q})v + Dv = \tau + P(t) \tag{27}$$

where

$q = (x, y, \theta)^T$ is the vector of the configuration coordinates;

$v = (v, \omega)^T$ is the vector of velocities;

$\tau = (\tau_1, \tau_2)$ is the vector of the torques applied to the wheels of the robot where τ_1 and τ_2 denote the torques of the right wheel and the 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 (28):

$$\dot{q} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{28}$$

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, and v and ω are the linear and angular velocities, respectively.

Equation (29) shows the parametrization of a trapezoidal primary generalized type 2 MF with a Gaussian secondary MF (*trapagausstype2*): the $a_1, b_1, c_1,$ and d_1 parameters are for the upper MFs, and $a_2, b_2, c_2,$ and d_2 are for the lower MFs:

$$\mu(x, u) = \text{trapagausstype2}(x, u, [a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2, \alpha, \rho])$$

$$\mu(x, u) = \exp\left[-\frac{1}{2}\left(\frac{u-p_x}{\sigma_u}\right)^2\right],$$

where

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

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

$$\bar{\mu}(x) = \begin{cases} \max(\mu_1(x), \mu_2(x)) & \forall x \notin (b_1, c_2) \\ 1 & \forall x \in (b_1, c_2) \end{cases} \tag{29}$$

$$\underline{\mu}(x) = \min(\alpha, \min(\mu_1(x), \mu_2(x)))$$

$$p_x = \max\left(\min\left(\frac{x-a_x}{b_x-a_x}, 1, \frac{d_x-x}{d_x-c_x}\right), 0\right),$$

where $a_x = \frac{a_1+a_2}{2}, b_x = \frac{b_1+b_2}{2}, c_x = \frac{c_1+c_2}{2}, d_x = \frac{d_1+d_2}{2}$

$$\delta = \bar{\mu}(x) - \underline{\mu}(x)$$

$$\sigma_u = \frac{1+\rho}{2\sqrt{3}}\delta + \varepsilon$$

where ρ is the fraction of uncertainty of the support secondary membership function.

The general type 2 fuzzy controller (GT2FC) for this problem is a Sugeno system, and it is made up of two antecedents, the linear error (LE) which is observed in Figure 4 and the angular error (WE) which is observed in Figure 5, where the linguistic variables of the antecedents are granulated into negative, zero, and positive linguistic values (N, Z, P). The consequents of this controller consist of constant values, which are identified as torque 1 (T1), which is observed in Figure 6, and torque 2 (T2), which is observed in Figure 7.

The parameters for the GT2MF construction of antecedents are presented in Table 2.

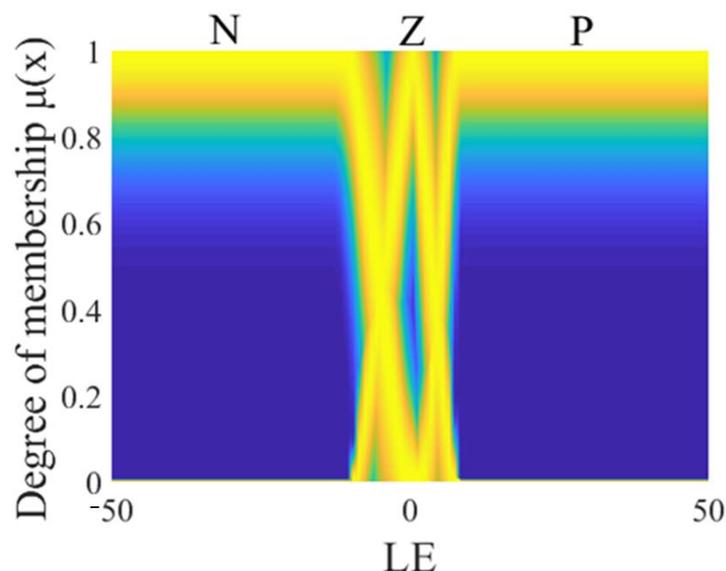


Figure 4. Antecedent of the GT2FC.

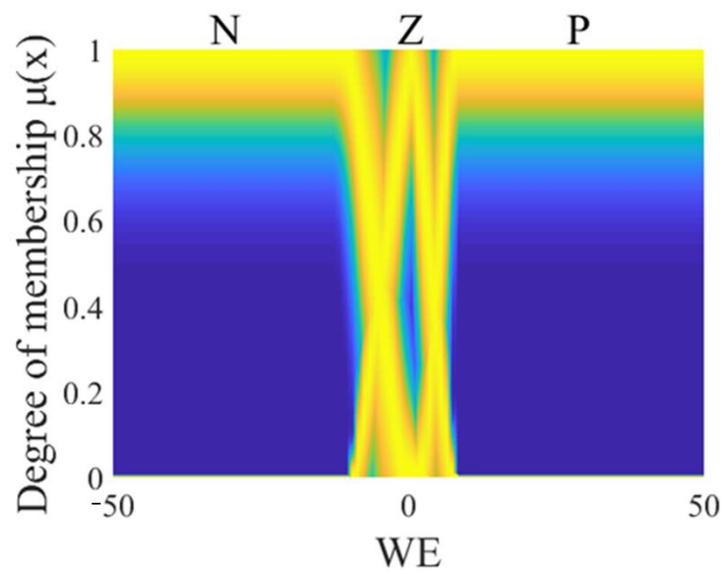


Figure 5. Antecedent of the GT2FC.

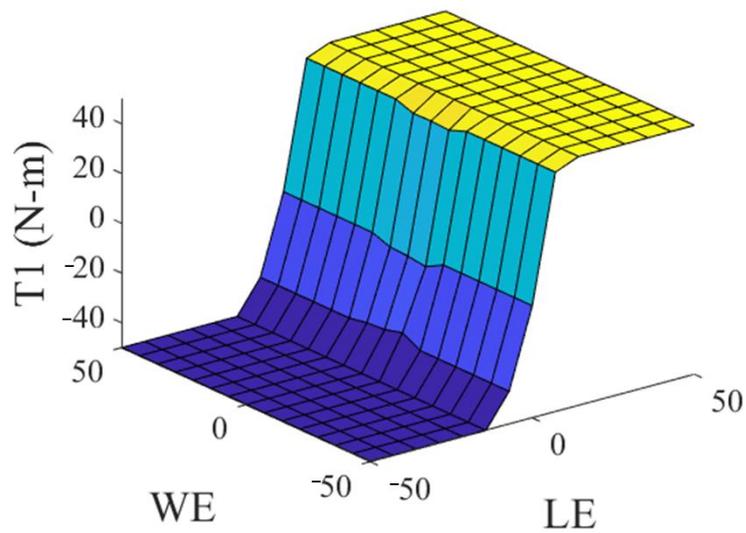


Figure 6. Consequent of the GT2FC.

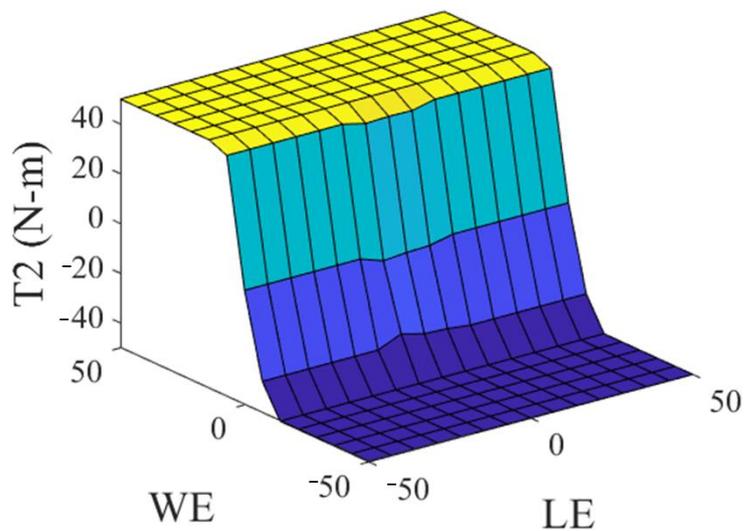


Figure 7. Consequent of the GT2FC.

Table 2. GT2MF antecedent parameters.

LE										
GT2MF	a_1	b_1	c_1	d_1	a_2	b_2	c_2	d_2	α	ρ
N	-50	-50	-15	-5	-50	-50	-5	3	0.7	1
Z	-10	-2	6	-	-8	3	8	-	-	2
P	0	5	50	50	5	10	50	50	0.7	1
WE										
GT2MF	a_1	b_1	c_1	d_1	a_2	b_2	c_2	d_2	α	ρ
N	-50	-50	-15	-5	-50	-50	-5	3	0.7	1
Z	-10	-2	6	-	-8	3	8	-	-	2
P	0	5	50	50	5	10	50	50	0.7	1

In the consequents, the values of the variables are constant: -50, 0, and 50 for N, Z, and P, with the fuzzy rules presented as follows:

1. R1: If the LE is N and WE is N, then T1 is -50 and T2 is -50.
2. R2: If the LE is N and WE is Z, then T1 is -50 and T2 is 0.
3. R3: If the LE is N and WE is P, then T1 is -50 and T2 is 50.
4. R4: If the LE is Z and WE is N, then T1 is 0 and T2 is -50.
5. R5: If the LE is Z and WE is Z, then T1 is 0 and T2 is 0.
6. R6: If the LE is Z and WE is P, then T1 is 0 and T2 is 50.
7. R7: If the LE is P and WE is N, then T1 is 50 and T2 is -50.
8. R8: If the LE is P and WE is Z, then T1 is 50 and T2 is 0.
9. R9: If the LE is P and WE is P, then T1 is 50 and T2 is 50.

The plant model of the mobile robot used for the experimentation is shown in Figure 8.

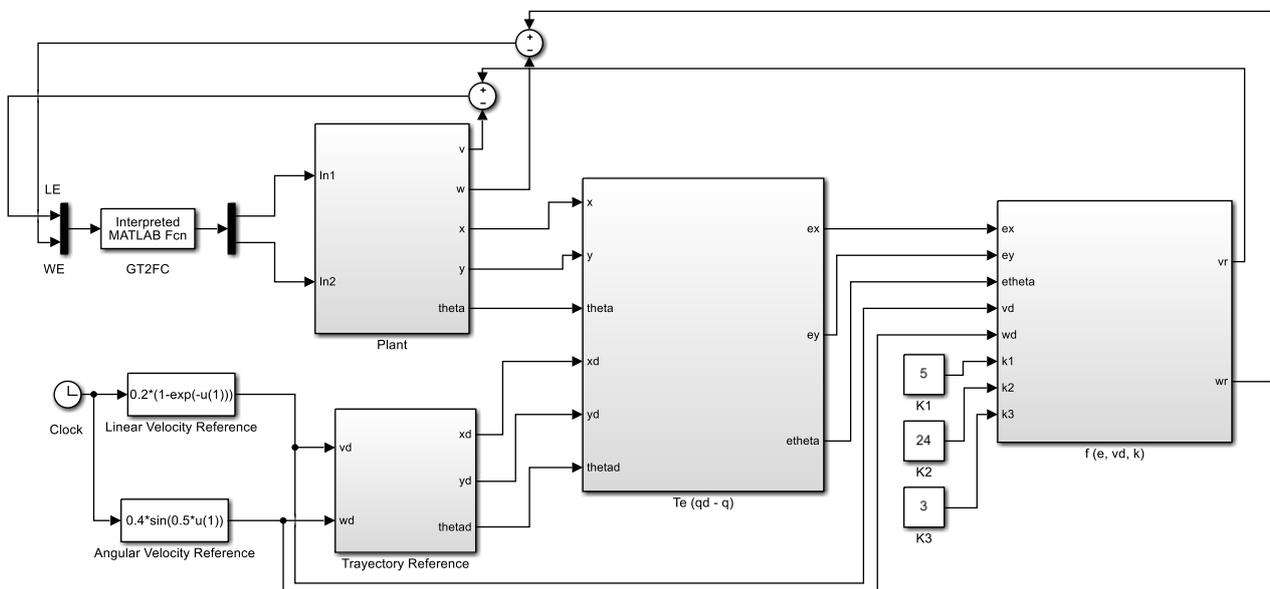


Figure 8. GT2FC of the mobile robot.

The GT2FDE method was used to test the efficiency of the general type 2 fuzzy controller. The configuration of the GT2FDE algorithm parameters used in the experimentation are as follows: CR parameter at 0.5, population of 45, number of simulations is 30, 100 generations.

To test the performance index of the GT2FC controller, simulations were performed without disturbances and with disturbances in the controller. The following Table 3 indicates the types of disturbances and the configurations of the used parameters.

Table 3. Types of disturbances applied to the controller.

Disturbance Number	Type of Disturbance	Configuration of the Parameters
1	Uniform random number	0.05
2	Band-limited white	0.1
3	Pulse-generated	1, 1, 5, and 0

The performance index used for the simulations are those presented in the following Equations (30)–(35):

$$ITAE = \sum_{t=1}^N t(x(t) - \hat{x}(t)) \tag{30}$$

$$ITSE = \sum_{t=1}^N t((x(t) - \hat{x}(t))^2) \tag{31}$$

$$IAE = \sum_{t=1}^N |x(t) - \hat{x}(t)| \tag{32}$$

$$ISE = \sum_{t=1}^N |(x(t) - \hat{x}(t))^2| \tag{33}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (x_t - \hat{x}_t)^2} \tag{34}$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \tag{35}$$

Table 4 summarizes the simulations without applying disturbance to the GT2FC, LE and WE DE both use a fixed value of F of 0.5 and a GT2FC.

Table 4. Performance index results without disturbance.

Performance Index	LE DE	WE DE	LE GT2FC	WE GT2FC
ITAE	1.53×10^2	1.79×10^1	2.16×10^0	1.50×10^2
ITSE	3.55×10^1	5.15×10^{-01}	7.61×10^{-01}	3.51×10^1
IAE	6.49×10^0	1.77×10^0	1.84×10^0	6.34×10^0
ISE	2.32×10^0	1.38×10^0	1.24×10^0	2.13×10^0
RMSE	2.98×10^0	2.98×10^0	2.82×10^0	2.82×10^0
MSE	8.90×10^0	8.88×10^0	7.94×10^0	7.96×10^{-02}
Execution time	12,227.514 s		12,592.143 s	

Figures 9–12, the pink color indicates the desired reference signal and the blue color represents the obtained signal.

Figure 9 represents the trajectories of the LE and the WE, without considering disturbances in the controller, of the best result obtained in the experiment, where it can be seen that the trajectories for the LE and the WE are very close to the desired trajectories.

Table 5 shows the simulation results with different disturbances affecting the controller as well as the different errors that were calculated in the experimentation.

Figure 10 shows the best result of the trajectories obtained for the LE and the WE when using disturbance 1; for the LE, it was observed that the trajectory had a small separation from the desired line, while the WE was closer to the desired line.

In the case of disturbance 2, the best trajectories obtained for the LE and the WE are presented in Figure 11, in which we can note that this type of disturbance generates a variation between the desired line and the one obtained for both trajectories.

Finally, the trajectories of the LE and the WE for disturbance 3 are represented in Figure 12. In this case, it was observed that they had a small variation with respect to the desired line.

Regarding the results presented in Tables 3 and 4, the different errors show that it is possible to obtain stability in the LE and WE trajectories for the different types of disturbances applied to the controller. It is also observed that the different metrics used showed stability in terms of their values, which is corroborated by Figures 10–12.

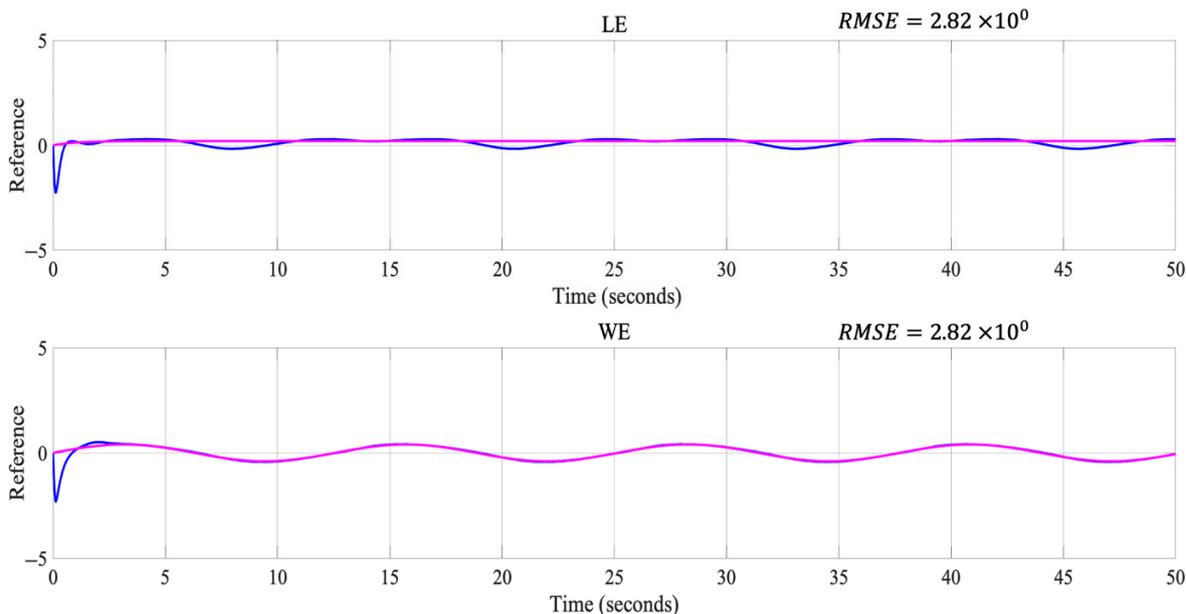


Figure 9. Representation of the best result representing the LE and the WE.

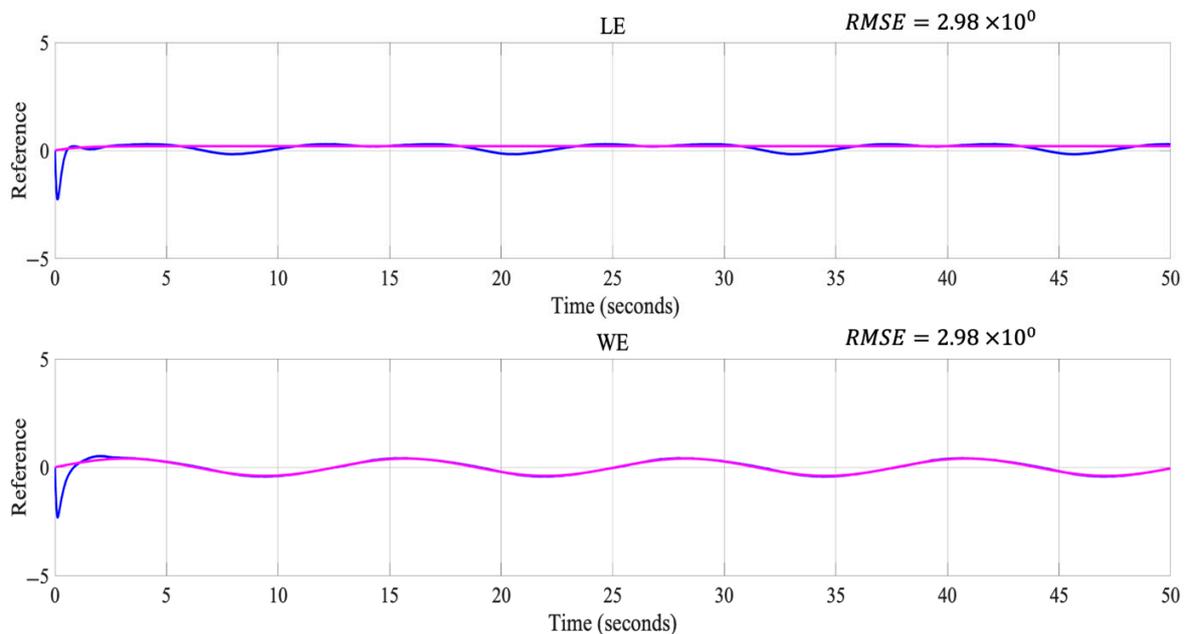


Figure 10. Representation of the best result representing the LE and the WE.

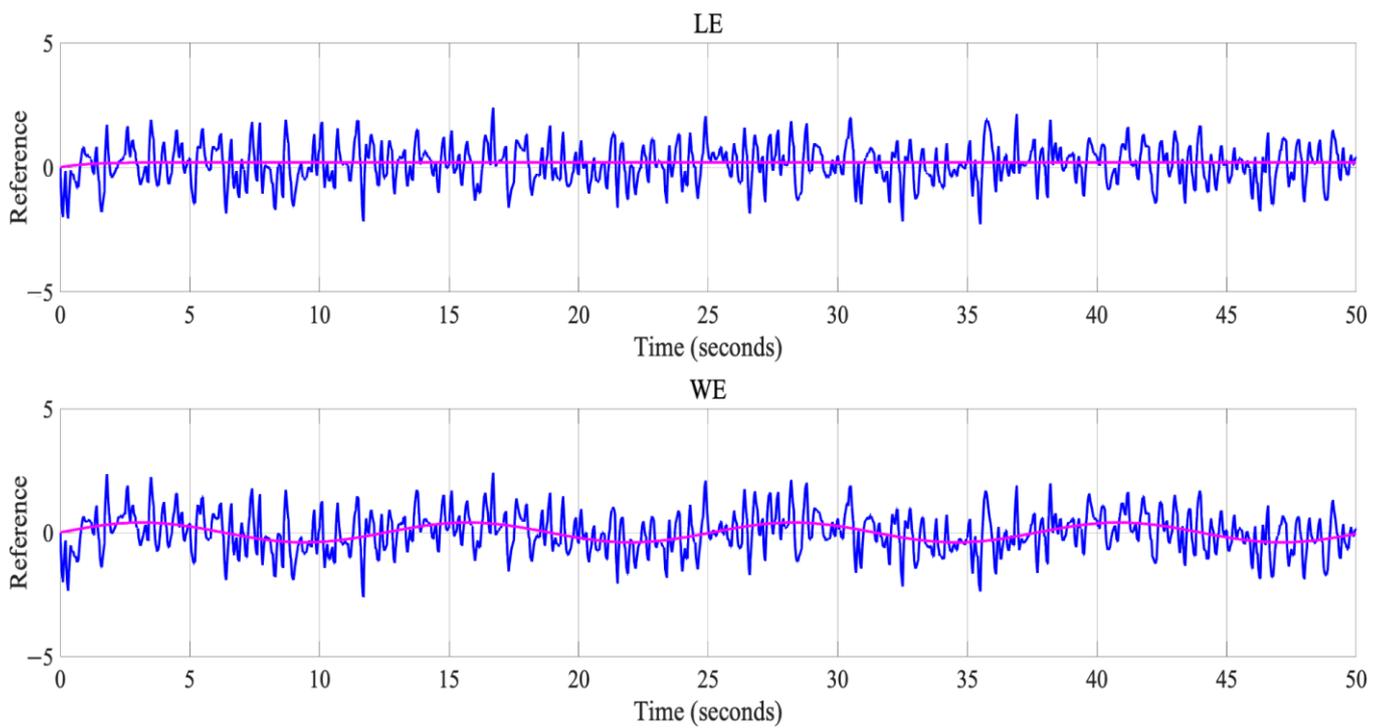


Figure 11. Representation of the best result representing the LE and the WE.

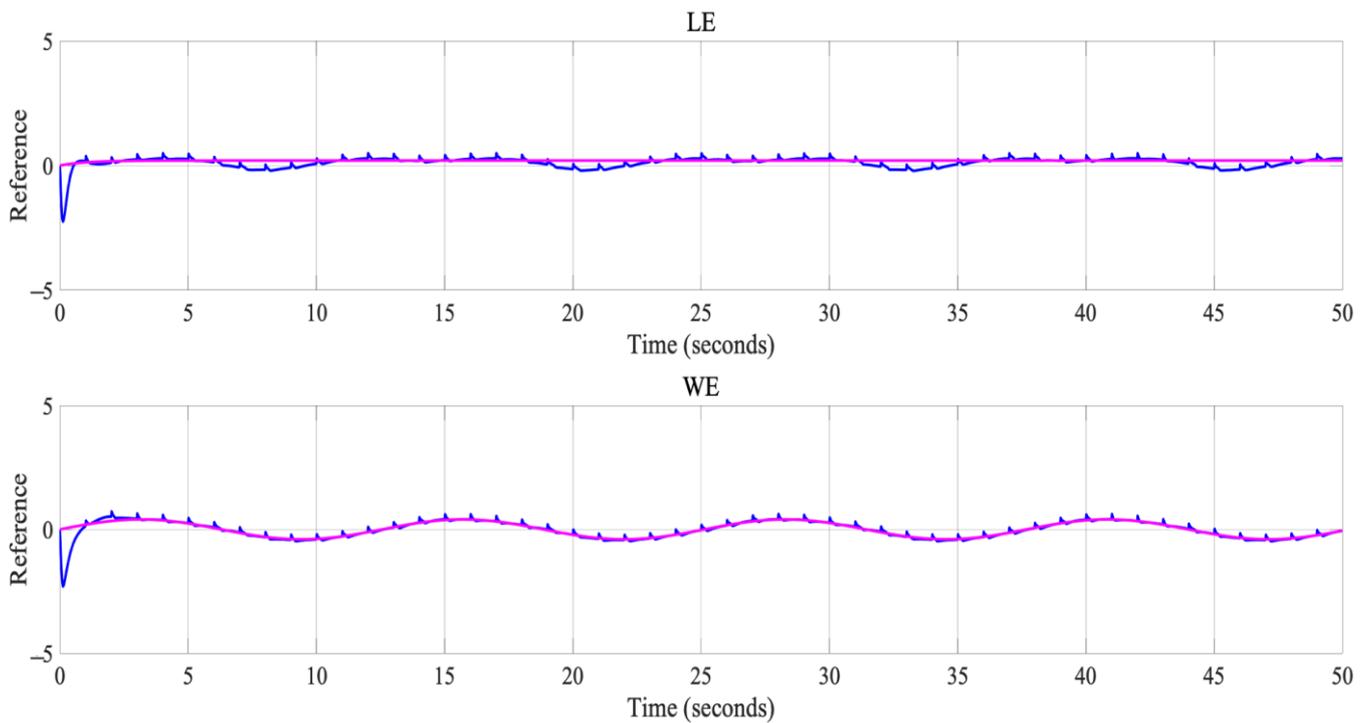


Figure 12. Representation of the best result representing the LE and the WE.

Table 5. Simulations obtained applying one, two, and three disturbances, respectively.

Performance Index	1		2		3	
	LE GT2FC	WE GT2FC	LE GT2FC	WE GT2FC	LE GT2FC	WE GT2FC
ITAE	1.79×10^1	1.53×10^2	7.01×10^2	6.55×10^2	4.24×10^1	1.60×10^2
ITSE	5.15×10^{-1}	3.55×10^2	6.31×10^2	2.93×10^1	3.15×10^0	3.83×10^1
IAE	1.77×10^0	6.49×10^0	2.88×10^1	1.17×10^0	2.68×10^0	6.78×10^0
ISE	1.38×10^0	2.32×10^0	2.67×10^1	1.70×10^{-01}	1.47×10^0	2.42×10^0
RMSE	2.98×10^0	2.98×10^0	3.44×10^0	4.40×10^{-01}	2.94×10^0	2.94×10^0
MSE	8.88×10^0	8.90×10^0	1.18×10^1	6.70×10^{-01}	8.65×10^0	8.65×10^0
Execution time	23,134.33 s		36,826.203 s		7067.877 s	

6. Conclusions

The present work shows generalized type 2 fuzzy logic being utilized to dynamically modify the F parameter in differential evolution. In the same way, a robot controller was implemented with a generalized type 2 fuzzy system. In fact, we can highlight this as the main contribution of this article since there is no similar work in the literature regarding the differential evolution algorithm.

Regarding the results presented in Tables 4 and 5, the different errors show that it is possible to obtain stability in the LE and WE trajectories in the different disturbances applied to the controller. The analysis for the metrics is explained below. In the case of the ITAE metric, it can be observed that the damping of the controller for the different applied disturbances is in the same order. ITSE has the characteristic that when faced with a unit step type input, the response has a large initial error given that it has a small weight, and it can be observed that the results obtained vary with respect to the LE and the WE, where it is observed that the LE has lower values in disturbances 1 and 2. The IAE performance index is not easy to evaluate; however, it is observed in the results they were of the same order, and we could consider them as good. ISE is one of the most important metrics to evaluate a controller. In this case, our results show that for the three disturbances as well as the controller without a disturbance, we had a good performance because it is possible to minimize this performance index. Regarding the RMSE and MSE, all the obtained results were of the same order and indicated that the difference between the desired and obtained lines was minimal, as shown in Figures 10–12, where it can be seen that the trajectories for the LE and the WE were very close.

In the future, we plan to consider other kinds of applications [40] and extend the presented approach to type 3 fuzzy logic as has been outlined in [41–44].

Author Contributions: Methodology and conceptualization, O.C. and P.M.; software, C.P. and P.O.; validation, C.P., P.O. and O.C.; writing—original draft preparation, C.P. and P.O.; writing—review and editing, O.C. and P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: We thank Tijuana Institute of Technology for the support.

Conflicts of Interest: The authors declare that there is no conflict of interest.

References

1. Sridharan, M. Short review on various applications of fuzzy logic-based expert systems in the field of solar energy. *Int. J. Ambient. Energy* **2022**, *43*, 5112–5128. [CrossRef]
2. Zhao, L.; Yin, Z.; Yu, K.; Tang, X.; Xu, L.; Guo, Z.; Nehra, P. A Fuzzy Logic Based Intelligent Multi-Attribute Routing Scheme for Two-layered SDVNs. *IEEE Trans. Netw. Serv. Manag.* **2022**, *1*. [CrossRef]
3. Hosseinpour, S.; Martynenko, A. Application of fuzzy logic in drying: A review. *Dry. Technol.* **2022**, *40*, 797–826. [CrossRef]
4. Castillo, O.; Castro, J.R.; Melin, P. Type-2 fuzzy logic systems. In *Interval Type-3 Fuzzy Systems: Theory and Design*; Springer: Cham, Switzerland, 2022; pp. 5–11. [CrossRef]

5. Castillo, O.; Melin, P. *New Perspectives on Hybrid Intelligent System Design based on Fuzzy Logic. Neural Networks and Metaheuristics*; Springer Nature: Cham, Switzerland, 2022; Volume 1050.
6. Kaur, J.; Khehra, B.S.; Singh, A. Significance of Fuzzy Logic in the Medical Science. In *Computer Vision and Robotics*; Springer: Singapore, 2022; pp. 497–509. [[CrossRef](#)]
7. Quek, S.G.; Selvachandran, G.; Sham, R.; Siau, C.S.; Ramli, M.H.M.; Ahmad, N. A Fuzzy Logic Based Optimal Network System for the Delivery of Medical Goods via Drones and Land Transport in Remote Areas. In *International Conference on Intelligent Systems Design and Applications*; Springer: Cham, Switzerland, 2022; pp. 1306–1312. [[CrossRef](#)]
8. Qazi, S.; Iqbal, N.; Raza, K. Fuzzy Logic-Based Hybrid Models for Clinical Decision Support Systems in Cancer. In *Computational Intelligence in Oncology*; Springer: Singapore, 2022; pp. 201–213. [[CrossRef](#)]
9. Eckert, J.J.; Barbosa, T.P.; Silva, F.L.; Roso, V.R.; Silva, L.C.; da Silva, L.A.R. Optimum fuzzy logic controller applied to a hybrid hydraulic vehicle to minimize fuel consumption and emissions. *Expert Syst. Appl.* **2022**, *207*, 117903. [[CrossRef](#)]
10. Inozemtsev, A.; Petrochenkov, A.; Kazantsev, V.; Shmidt, I.; Sazhenkov, A.; Dadenkov, D.; Gribkov, I.; Ivanov, P. The Fuzzy Logic in the Problems of Test Control of a Bypass Turbojet Engine Gas Generator. *Mathematics* **2022**, *10*, 484. [[CrossRef](#)]
11. Pan, Y.; Wu, Y.; Lam, H.-K. Security-Based Fuzzy Control for Nonlinear Networked Control Systems with DoS Attacks via a Resilient Event-Triggered Scheme. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 4359–4368. [[CrossRef](#)]
12. Miranda, M.H.; Silva, F.L.; Lourenço, M.A.; Eckert, J.J.; Silva, L.C. Electric vehicle powertrain and fuzzy controller optimization using a planar dynamics simulation based on a real-world driving cycle. *Energy* **2022**, *238*, 121979. [[CrossRef](#)]
13. Patel, H.R. Fuzzy-based metaheuristic algorithm for optimization of fuzzy controller: Fault-tolerant control application. *Int. J. Intell. Comput. Cybern.* **2022**, *15*, 599–624. [[CrossRef](#)]
14. Boudia, A.; Messalti, S.; Harrag, A.; Boukhni, M. New hybrid photovoltaic system connected to superconducting magnetic energy storage controlled by PID-fuzzy controller. *Energy Convers. Manag.* **2021**, *244*, 114435. [[CrossRef](#)]
15. Yang, T.; Sun, N.; Fang, Y. Adaptive Fuzzy Control for a Class of MIMO Underactuated Systems with Plant Uncertainties and Actuator Deadzones: Design and Experiments. *IEEE Trans. Cybern.* **2022**, *52*, 8213–8226. [[CrossRef](#)]
16. Wu, L.; Qian, F.; Wang, L.; Ma, X. An improved type-reduction algorithm for general type-2 fuzzy sets. *Inf. Sci.* **2022**, *593*, 99–120. [[CrossRef](#)]
17. Chen, Y. Study on Weighted-Based Discrete Noniterative Algorithms for Computing the Centroids of General Type-2 Fuzzy Sets. *Int. J. Fuzzy Syst.* **2022**, *24*, 587–606. [[CrossRef](#)]
18. Chen, Y.; Li, C.; Yang, J. Design of Discrete Noniterative Algorithms for Center-of-Sets Type Reduction of General Type-2 Fuzzy Logic Systems. *Int. J. Fuzzy Syst.* **2022**, *24*, 2024–2035. [[CrossRef](#)]
19. Shi, J. A unified general type-2 fuzzy PID controller and its comparative with type-1 and interval type-2 fuzzy PID controller. *Asian J. Control.* **2022**, *24*, 1808–1824. [[CrossRef](#)]
20. Cuevas, F.; Castillo, O.; Cortés-Antonio, P. Generalized Type-2 Fuzzy Parameter Adaptation in the Marine Predator Algorithm for Fuzzy Controller Parameterization in Mobile Robots. *Symmetry* **2022**, *14*, 859. [[CrossRef](#)]
21. Carvajal, O.; Melin, P.; Miramontes, I. Optimal Design and Internet of Things Implementation of a General Type-2 Classifier for Blood Pressure Levels. In *International Conference on Intelligent and Fuzzy Systems*; Springer: Cham, Switzerland, 2022; pp. 722–729. [[CrossRef](#)]
22. Ontiveros-Robles, E.; Castillo, O.; Melin, P. Towards asymmetric uncertainty modeling in designing General Type-2 Fuzzy classifiers for medical diagnosis. *Expert Syst. Appl.* **2021**, *183*, 115370. [[CrossRef](#)]
23. Ochoa, P.; Castillo, O.; Melin, P.; Soria, J. Differential Evolution with Shadowed and General Type-2 Fuzzy Systems for Dynamic Parameter Adaptation in Optimal Design of Fuzzy Controllers. *Axioms* **2021**, *10*, 194. [[CrossRef](#)]
24. Mendel, J.M. General type-2 fuzzy logic systems made simple: A tutorial. *IEEE Trans. Fuzzy Syst.* **2014**, *22*, 1162–1182. [[CrossRef](#)]
25. Price, K.V.; Storn, R.M.; Lampinen, J.A. The differential evolution algorithm. In *Differential Evolution: A Practical Approach to Global Optimization*; Springer: Cham, Switzerland, 2005; pp. 37–134.
26. Song, Y.; Cai, X.; Zhou, X.; Zhang, B.; Chen, H.; Li, Y.; Deng, W.; Deng, W. Dynamic hybrid mechanism-based differential evolution algorithm and its application. *Expert Syst. Appl.* **2023**, *213*, 118834. [[CrossRef](#)]
27. Abdelkader, E.M.; Moselhi, O.; Marzouk, M.; Zayed, T. An exponential chaotic differential evolution algorithm for optimizing bridge maintenance plans. *Autom. Constr.* **2022**, *134*, 104107. [[CrossRef](#)]
28. Deng, W.; Ni, H.; Liu, Y.; Chen, H.; Zhao, H. An adaptive differential evolution algorithm based on belief space and generalized opposition-based learning for resource allocation. *Appl. Soft Comput.* **2022**, *127*, 109419. [[CrossRef](#)]
29. Akhtar, M.; Manna, A.K.; Duary, A.; Bhunia, A.K. A hybrid tournament differential evolution algorithm for solving op-timisation problems and applications. *Int. J. Oper. Res.* **2022**, *45*, 300–343. [[CrossRef](#)]
30. Kharchouf, Y.; Herbazi, R.; Chahboun, A. Parameter’s extraction of solar photovoltaic models using an improved differential evolution algorithm. *Energy Convers. Manag.* **2021**, *251*, 114972. [[CrossRef](#)]
31. Zeng, Z.; Zhang, M.; Zhang, H.; Hong, Z. Improved differential evolution algorithm based on the sawtooth-linear population size adaptive method. *Inf. Sci.* **2022**, *608*, 1045–1071. [[CrossRef](#)]
32. De Athayde Prata, B.; Rodrigues, C.D.; Framinan, J.M. A differential evolution algorithm for the customer order scheduling problem with sequence-dependent setup times. *Expert Syst. Appl.* **2022**, *189*, 116097. [[CrossRef](#)]

33. Castillo, O.; Valdez, F.; Soria, J.; Yoon, J.H.; Geem, Z.W.; Peraza, C.; Ochoa, P.; Amador-Angulo, L. Optimal Design of Fuzzy Systems Using Differential Evolution and Harmony Search Algorithms with Dynamic Parameter Adaptation. *Appl. Sci.* **2020**, *10*, 6146. [[CrossRef](#)]
34. Ochoa, P.; Castillo, O.; Soria, J. Differential evolution algorithm using a dynamic crossover parameter with fuzzy logic applied for the CEC 2015 benchmark functions. In *North American Fuzzy Information Processing Society Annual Conference*; Springer: Cham, Switzerland, 2018; pp. 580–591. [[CrossRef](#)]
35. Ochoa, P.; Castillo, O.; Soria, J. Interval type-2 fuzzy logic dynamic mutation and crossover parameter adaptation in a fuzzy differential evolution method. In *Intuitionistic Fuzziness and Other Intelligent Theories and Their Applications*; Springer: Cham, Switzerland, 2018; pp. 81–94. [[CrossRef](#)]
36. Ochoa, P.; Castillo, O.; Soria, J. Differential Evolution Using Fuzzy Logic and a Comparative Study with Other Metaheuristics. In *Nature-Inspired Design of Hybrid Intelligent Systems*; Springer: Cham, Switzerland, 2017; pp. 257–268. [[CrossRef](#)]
37. Peraza, C.; Ochoa, P.; Castillo, O.; Geem, Z.W. Interval-Type 3 Fuzzy Differential Evolution for Designing an Interval-Type 3 Fuzzy Controller of a Unicycle Mobile Robot. *Mathematics* **2022**, *10*, 3533. [[CrossRef](#)]
38. Fukao, T.; Nakagawa, H.; Adachi, N. Adaptive tracking control of a nonholonomic mobile robot. *IEEE Trans. Robot. Autom.* **2000**, *16*, 609–615. [[CrossRef](#)]
39. Martinez, R.; Rodriguez, A.; Castillo, O.; Melin, P.; Aguilar, L.T. Optimization of type-2 fuzzy logic con-trollers for mobile robots using evolutionary methods. In *Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, USA, 11–14 October 2009; pp. 4764–4769.
40. Melin, P.; Urias, J.; Solano, D.; Soto, M.; Lopez, M.; Oscar Castillo, O. Voice Recognition with Neural Networks, Type-2 Fuzzy Logic and Genetic Algorithms. *Eng. Lett.* **2006**, *13*, 108–116.
41. Castillo, O.; Castro, J.R.; Melin, P. Interval Type-3 Fuzzy Control for Automated Tuning of Image Quality in Televisions. *Axioms* **2022**, *11*, 276. [[CrossRef](#)]
42. Castillo, O.; Castro, J.R.; Melin, P. Interval Type-3 Fuzzy Systems: Theory and Design. *Stud. Fuzziness Soft Comput.* **2022**, *418*, 1–100.
43. Mohammadzadeh, A.; Castillo, O.; Band, S.S.; Mosavi, A. A Novel Fractional-Order Multiple-Model Type-3 Fuzzy Control for Nonlinear Systems with Unmodeled Dynamics. *Int. J. Fuzzy Syst.* **2021**, *23*, 1633–1651. [[CrossRef](#)]
44. Castillo, O.; Melin, P. Towards Interval Type-3 Intuitionistic Fuzzy Sets and Systems. *Mathematics* **2022**, *10*, 4091. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.