applied sciences

MDPI

Article

Compact Fuzzy Systems Based on Boolean Relations

Helbert Espitia [1,*], José Soriano [1], Iván Machón [2] and Hilario López [2]

[1] Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas, 11021-110231588 Bogotá, Colombia; jairosoriano@udistrital.edu.co
[2] Departamento de Ingeniería Eléctrica, Electrónica de Computadores y Sistemas, Universidad de Oviedo, Campus de Viesques, 33204 Gijón/Xixón, Spain; machonivan@uniovi.es (I.M.); hilario@uniovi.es (H.L.)
* Correspondence: heespitiac@udistrital.edu.co

Abstract: This document presents some considerations and procedures to design a compact fuzzy system based on Boolean relations. In the design process, a Boolean codification of two elements is extended to a Kleene's of three elements to perform simplifications for obtaining a compact fuzzy system. The design methodology employed a set of considerations producing equivalent expressions when using Boole and Kleene algebras establishing cases where simplification can be carried out, thus obtaining compact forms. In addition, the development of two compact fuzzy systems based on Boolean relations is shown, presenting its application for the identification of a nonlinear plant and the control of a hydraulic system where it can be seen that compact structures describes satisfactory performance for both identification and control when using algorithms for optimizing the parameters of the compact fuzzy systems. Finally, the applications where compact fuzzy systems are based on Boolean relationships are discussed allowing the observation of other scenarios where these structures can be used.

Keywords: Boole; compact; control; fuzzy systems; identification; Kleene

Citation: Espitia, H.; Soriano, J.; Machón, I.; López, H. Compact Fuzzy Systems Based on Boolean Relations. *Appl. Sci.* **2021**, *11*, 1793. https://doi.org/10.3390/app11041793

Academic Editor: Youngchul Bae

Received: 29 December 2020
Accepted: 10 February 2021
Published: 18 February 2021

1. Introduction

Modeling, controlling, and performing any system is an essential task for the designer/user to apply the necessary corrections to make it more effective. Frequently, evaluating a system is done under fuzzy rules, for instance, by employing qualitative instead of numeric marks or incomplete numeric functions, etc. [1]. Thus, fuzzy systems theory is built on concrete and practical formation for many relevant applications, especially in decision-making processes [2,3].

Various real-life problems are problematic since they involve human beings, mechanical elements, and other issues. Consequently, cases like these or even simpler ones may include diverse uncertainty conditions of high relevance when achieving satisfactory outcomes. Uncertainty may include randomness, fuzziness, indistinguishability, and incompleteness [4,5]. The theory on fuzzy sets and fuzzy logic permits manipulating incomplete and imprecise information of objects belonging to a concept [6].

Zadeh proposed a Fuzzy Set (FS) as a focus to process and represent vagueness in the real world [7]. Fuzzy Inference Systems (FIS) are utilized to generate sets of fuzzy rules in problem solving in applications like detection [8,9], prediction [10,11], classification [12–14], and other tasks [15–17].

In [18], a review of the main theories involved in hybrid models based on neuro-fuzzy systems is presented. The basic concepts regarding the history of hybrid models, the features of the leading models, and their applications also are presented; some neuro-fuzzy structures observed are: fuzzy regression, adaptive neuro-based fuzzy inference systems, autoassociative neuro-fuzzy systems, and fuzzy cognitive maps. It also reviews fuzzification and defuzzification techniques and training algorithms. The authors exposed that fuzzy neural network models and their derivations are useful for the design of system with high degree of accuracy and a suitable level of interpretability allowing applications in different areas of science.

Using fuzzy sets provides a foundation for systematically handling vague and inaccurate concepts; mainly, fuzzy sets can be used when representing linguistic variables, which are values defined in a linguistic way [19]. Fuzzy logic sets are a suitable solutions alternative in cases with uncertainty, inaccuracy, and ambiguity in the presence of phenomena to be controlled or modeled [20–22].

Regarding fuzzy inference systems, the most widely known are Mamdani, Takagi-Sugeno, and Tsukamoto. Mamdani systems use fuzzy sets in inputs and outputs and include and aggregation of fuzzy sets acquired from each rule's implication. The result is obtained by defuzzification. Takagi-Sugeno systems employ functions that depend on the input value; the inference process result is accepted as a weighted addition whose value is given by each rule's inference process. On the other hand, Tsukamoto systems employ fuzzy sets for input and output. In the output, it uses injective membership functions to get the preimage of the value from the implication process; later, the results are weighted with the values obtained in implication process.

Concerning additional fuzzy inference developments, complex fuzzy sets are proposed to improve the design and modeling of real-life applications [23]. A Complex Fuzzy Set (CFS) [24] is an extension of a fuzzy set where the membership function consists of an amplitude term and one phase. Accordingly, paper [25] suggested a framework for logic reasoning called Complex Fuzzy Logic (CFL), employing the CFL theory since this is useful to make systems capable of managing different problems.

As a variation, the Mamdani Complex Inference System (M-CFIS) is proposed in [26]. Others are the CFIS like Neuro-complex, Adaptive Neuro-Complex Fuzzy Inferential System (ANCFIS) with higher-order TSK models [27], the Randomized Adaptive Neuro-Complex Fuzzy Inference System (RANCFIS) [28], and the Fast Adaptive Neuro-Complex Fuzzy Inference System (FANCFIS) [29]. However, a deficiency detected on the M-CFIS is that the base of rules may become redundant to a specific set of rules. In addition, reference [23] presents the analysis of a collection of complex data using properties from the fuzzy concept network and the complex soft set.

On the other hand, Mamdani Complex Fuzzy Inference System (M-CFIS) allows managing events not restricted only to values of a point in a specific time but including all the values inside some time intervals (phase term). In such scenarios, fuzzy complex theory permits one to see the range and phase values of an event, which means better performance; nevertheless, the set of rules may be redundant to a specific dataset [2].

Meanwhile, the fuzzy sets theory has been applied to manage uncertainty in numerous areas; even so, it shows limitations to face situations of inaccuracy when sources of vagueness appear [4]. Thus, the literary sources include different extensions of fuzzy sets to overcome such limitations:

- Intuitionistic Fuzzy Sets (IFS) proposed by Atanassov [30]. These consider the membership and no membership grade for each element simultaneously.
- Type 2 Fuzzy Sets (T2FS), incorporate uncertainty in defining the membership function through a fuzzy set to model the unitary interval. Type 2 Fuzzy Sets (T2FS) include uncertainties when defining the membership function using a fuzzy set to model it [31].
- Interval Valued Fuzzy Sets (IVFS). The membership grade of an element is given by a closed subinterval of the unitary interval; thus, the length of that interval is understood as a measurement of lack of certainty to build an element's accurate membership grade [32].
- Multiset-Based Systems, where the membership grade of an element is given by a subset of $[0, 1]$ [33].
- The theory of Hesitant Fuzzy Sets (HFS) is about managing situations where a set of values is possible to use in the process of defining the membership grade of an element [34].

Experts use intuitionist fuzzy sets to manage the evaluation of an alternative with specific values of membership and nonmembership. Meanwhile, hesitant fuzzy sets handle situations where experts doubt several possible membership values to evaluate an option [35].

The theory of fuzzy sets of Zadeh [7] has been applied in several fields. The idea of a fuzzy set is welcome since it manages uncertainty and vagueness that the Cantorian set cannot address. In fuzzy sets theory, the membership of an element is a unique value between zero and one. However, it may not always be true that the degree of nonmembership of an element in a fuzzy set is equal to one minus the degree of membership because there may be some degree of doubt; therefore, reference [30] introduced a generalization called intuitionistic fuzzy sets. This includes a margin of hesitation defined as one minus the sum of the degrees of membership and no membership [36].

On the other hand, hesitant fuzzy sets are useful to deal with group decision-making processes when the experts doubt possible memberships for an element. These possible memberships may include crisp values $[0, 1]$ and interval values [37]. Torra [34] uses the same approach introducing an extension for sets called hesitant fuzzy sets, given the standard difficulty that generally appears when it is necessary to define the grade of membership. Such a problem is not given by error range (as in IFS) or any distribution of possibilities (as in T2FS) but because there are possible values that make it difficult to decide which one is accurate. This is a common situation in the decision-making process for an expert regarding several grades of membership $\{0.43, 0.58, 0.79\}$ of element x in the set A. Thus, hesitant fuzzy sets become relevant since they make it easier to manage hesitation in typical real-life situations. Consequently, many developments have been carried out from different quantitative [37,38] and qualitative [39] perspectives, given that many doubts arise, which model the uncertainty in both directions. Besides, many HFS operators have been introduced to deal with this issue in various applications where decision-making is the most remarkable aspect [4,40].

Meanwhile, inference fuzzy systems based on Boolean relations are suitable when using Boole's algebra with sets whose membership values are given by $\{0, 1\}$, then a true table defines a model [41–43]. Nevertheless, these systems display low performance due to abrupt transitions. Consequently, a way to improve the performance is by replacing the Booleans for fuzzy sets [44]. When the concepts clearly define the membership in the range $[0, 1]$ which means that the limits of the concepts are "softened" (fuzzy) with a structure equivalent to $\{0, u, 1\}$ makes it possible to employ Kleene's algebra [45].

About related works, document [41] suggests the use of the Boolean algebra system design features. Conversely, paper [46] presents a practical approach of a methodology based on the design of automations employing fuzzy sets for implementing the process associated with defuzzification. Simultaneously, in [44] it is indicated that this proposal can be interpreted as a fuzzy inference system. This technique considers the interaction of sensors, actuators, and Boolean relations involved in control strategies of any industrial process [44,46].

As part of the design process, [47] employs Kleene's algebra to transform binary logic functions into fuzzy logic. Besides, paper [48] describes a method to simplify formulas using finite algebras. In addition, the work presented in [49] offers the vagueness modeling, ambiguity, and contrariness with trivalent and tetravalent fuzzy logic, employing Kleene and De Morgan algebraic structures, focusing on the tables and graphs. Thus, the design methodology based on these elements is presented in [50].

The proposed methodology considers a design using Boole's algebra, later extended to a fuzzy system, employing Kleene's algebra. Nevertheless, it is necessary to keep in mind that in Boole's algebra, the variable simplification is rooted in the law of excluded middle, which is not accomplished using Kleene's. However, a similar simplification with Kleene algebra may be achieved employing the absorption law.

Proposal Approach and Document Organization

This document offers a design proposal to fuzzy logic systems based on Boolean relations with compact structure. Previously in [50], a methodological design is proposed to implement fuzzy inference systems based on Boolean relations. This work is currently continued considering specific design cases of fuzzy systems with simplifications applied to equations in the inference process, providing compact systems useful in different applications as in identification and control of dynamic systems.

Related applications of these systems can be observed in [51] to predict chaotic time series while reference [52] identifies a variable charge in a distribution system; besides, document [53] displays a control of a synchronous generator of permanent magnets. Finally, the voltage control in a electrical grid is shown in [54]. Although in these works compact fuzzy structures are employed, these do not use the design methodology presented in [50] via Kleene's algebra that allow the formalization of these systems which is the main topic of this work.

The document is organized as follows: Section 2 describes the definitions employed for the design process of fuzzy inference systems based on Boolean relations and the simplification procedure with Kleene algebra used to determine the structures of compact systems. Section 3 shows the design process for specific cases of fuzzy inference systems based on Boolean relations called "compact systems", where Kleene algebra is employed for the simplification analysis. Section 4 introduces compact systems structures that are obtained considering two cases where the simplification of configurations that are represented in a truth table can be performed. Section 5 describes the application for the identification of a nonlinear plant, while Section 6 presents the application of a fuzzy PID (Proportional Integral Derivative) controller. Finally, Section 7 presents the discussion, and the conclusions are given in Section 8.

2. Framework

This section presents definitions and concepts necessary for the design of compact fuzzy systems based on Boolean relations. The methodology proposes the design in a Boolean table. Through design criteria, this table is converted into a Kleene's considering an intermediate variable u; then the equations that represent the true table are established to finally perform the implementation of the system.

The concepts shown in this section are used to determine the equations to implement compact fuzzy systems based on Boolean relations. For the design, the conversion of a truth table and the simplification of the terms are carried out using Kleene's algebra to establish compact structures.

First the definitions of De Morgan, Kleene, and Boole algebra are reviewed and also the Disjunctive Normal Form (DNF) in Kleene algebra obtained from a relations table. Subsequently, the description of fuzzy inference systems based on Boolean relationships is presented. Finally, a review of the design criteria used to convert a Boolean into a Kleene table is presented.

2.1. De Morgan, Kleene, and Boole Algebras

Boolean and Kleene algebras are used to carry out the design of compact fuzzy systems based on Boolean relations; therefore, the required concepts are detailed below. The following definitions are based on [45,55,56].

Definition 1 ([56]). *A De Morgan algebra $\mathcal{M} = (M, \vee, \wedge, ^-, 0, 1)$, consists of a set M with two internal binary operations $\wedge$ (meet), $\vee$ (join), an unit operation $^-$ (complement) and the constants 0 and 1. Considering $A, B, C \in M$ the following properties are defined:*

$$A \wedge A = A \qquad A \vee A = A \tag{1}$$
$$A \wedge B = B \wedge A \qquad A \vee B = B \vee A \tag{2}$$
$$A \wedge (B \wedge C) = (A \wedge B) \wedge C \qquad A \vee (B \vee C) = (A \vee B) \vee C \tag{3}$$
$$A \wedge (A \vee B) = A \qquad A \vee (A \wedge B) = A \tag{4}$$
$$A \wedge (B \vee C) = (A \wedge B) \wedge (A \wedge C) \qquad A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C) \tag{5}$$
$$A \wedge 1 = A \qquad A \vee 0 = A \tag{6}$$
$$\overline{(A \wedge B)} = \bar{A} \wedge \bar{B} \qquad \overline{(A \vee B)} = \bar{A} \vee \bar{B} \tag{7}$$
$$\bar{0} = 1 \qquad \bar{1} = 0 \tag{8}$$
$$\bar{\bar{A}} = A \tag{9}$$

Properties (1)–(4) indicate that $\mathcal{M}$ is a lattice, also if it complies with (5) the lattice is distributive; thus, the operations $\wedge$ and $\vee$ are associative, commutative, and distributive ($\vee$ with respect to $\wedge$ and vice versa). The property (6) indicates that the lattice is bounded. Fulfilling properties (1)–(9) implies that $\mathcal{M}$ is a De Morgan algebra.

Definition 2 ([56]). *A Kleene algebra $\mathcal{K} = (K, \vee, \wedge, ^-, 0, 1)$ is a De Morgan algebra that meets (10).*

$$A \wedge \bar{A} \leq B \vee \bar{B} \tag{10}$$

Definition 3 ([56]). *A Boole algebra $\mathcal{B} = (B, \vee, \wedge, ^-, 0, 1)$ is a Kleene algebra that satisfies (11) y (12).*

$$A \vee \bar{A} \;=\; 1 \tag{11}$$
$$A \wedge \bar{A} \;=\; 0 \tag{12}$$

Compared to Boolean algebra, in Kleene algebra the relation (13) is satisfied.

$$\begin{aligned} A \vee \bar{A} &\leq 1 \\ A \wedge \bar{A} &\geq 0 \end{aligned} \tag{13}$$

2.2. Disjunctive Normal Form

An element a in a lattice is $\vee$-irreducible (*join-irreducible*) if it is different from 0 and cannot be written as the disjunction of strictly minor elements; then, in a finite distributive lattice each element is uniquely the disjunction of incomparable elements $\vee$-irreducible [55–57]. The disjunctive normal form for an element a is a representation of the type $a_0 \vee a_1 \vee \ldots \vee a_l$, where each a_i is incomparable respect to each a_j, $(0 \leq i, j \leq l)$ if $i \neq j$ and each a_i is $\vee$-irreducible.

Disjunctive Normal Form Obtained from Tables

The following three rules are used to obtain the disjunctive normal form in $\mathcal{K}$ of a function f from a table [56].

- For rows that have value 1 in the column of f: Elaborate the conjunction of variables that have value 1 with the complements of variables that have value 0 and the constant 1 for variables that have value u.
- For rows that have value u in the column of f: Construct the conjunction of the variables that have value 1 with the complements of variables that have value 0 and the conjunction of both the variable and its complement for those that have value u.

- Elaborate the disjunction of the conjunctions obtained with the two previous rules. If there are no conjunctions, an empty disjunction will be obtained since it represents the disjunctive normal form for 0.

Table 1 shows an example of the $\vee$-irreducible terms used to elaborate the disjunctive normal form (DNF).

Table 1. Example of terms use for elaborate DNF in $\mathcal{K}$.

A_1	A_2	f	DNF
0	0	0	-
0	u	u	$\bar{A}_1 \wedge (A_2 \wedge \bar{A}_2)$
0	1	1	$\bar{A}_1 \wedge A_2$
u	0	u	$(A_1 \wedge \bar{A}_1) \wedge \bar{A}_2$
u	u	u	$(A_1 \wedge \bar{A}_1) \wedge (A_2 \wedge \bar{A}_2)$
u	1	1	$1 \wedge A_2$
1	0	1	$A_1 \wedge \bar{A}_2$
1	u	1	$A_1 \wedge 1$
1	1	1	$A_1 \wedge A_2$

The equation associated with the DNF of the Table 1 is:

$$f = [\bar{A}_1 \wedge A_2 \wedge \bar{A}_2] \vee [\bar{A}_1 \wedge A_2] \vee [A_1 \wedge \bar{A}_1 \wedge \bar{A}_2] \vee [A_1 \wedge \bar{A}_1 \wedge A_2 \wedge \bar{A}_2] \vee A_2 \vee [A_1 \wedge \bar{A}_2] \vee A_1 \vee [A_1 \wedge A_2] \quad (14)$$

Using Boolean sets, the rules can easily be represented by a truth table; however, using only values 1 and 0 there are intermediate cases that cannot be represented; therefore, the possibility of representing intermediate concepts is enhanced when adding the variable u that represents a value between 1 and 0.

2.3. Formula Simplification

For expressions obtained in normal forms applying the distributive property together with the absorption property in $\mathcal{K}$, the number of literals and operators can be reduced. As an example, Table 2 shows a possible codification for the Y function.

Table 2. Codification example for a Kleene table.

A_1	A_2	Y
1	1	1
u	u	u
0	0	0
0	u	0
0	1	0
u	1	u
1	0	0
1	u	u
u	0	0

The respective equation for Y from Table 2 is:

$$Y_1 = (A_1 \wedge A_2) \vee (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge \bar{A}_2) \vee (A_1 \wedge \bar{A}_1 \wedge A_2) \vee (A_1 \wedge A_2 \wedge \bar{A}_2) \quad (15)$$

Using the distributive and absorption properties:

$$Y_1 = (A_1 \wedge A_2) \wedge [1 \vee (\bar{A}_1 \wedge \bar{A}_2) \vee \bar{A}_1 \vee \bar{A}_2] \quad (16)$$

Then the simplified equation for Y is:

$$Y_1 = A_1 \wedge A_2 \tag{17}$$

2.4. Fuzzy Inference Systems Based on Boolean Relations

The fuzzy inference systems based on Boolean relations (FIS-BBR) use coding and design tools of Boolean algebra. The structure of the inference process is given by the truth table associated with the system design. In order to implement the FIS-BBR the two-element Boolean encoding extends to a three-element Kleene encoding.

In the operation of a FIS-BBR, actions that contribute to carry out a total action on a system are considered. The actions are fully accomplished by weighting an actuator v_m (virtual actuator) using a function Y_m that operates on it (activation function), which is a fuzzy function with values between zero and one. In this way, the output in a FIS-BBR is calculated as the sum of the partial results $y_m = v_m Y_m$ (virtual outputs), as shown in Equation (20).

Table 3 shows the coding for the inputs and outputs of a FIS-BBR. The rows of left side represent the rules that provide an activation output. The columns on the right side correspond to the activation functions. In this way two parts are identified in Table 3, one corresponding to an encoding for the sets of each entry; the other corresponds to the activation outputs associated with virtual outputs.

Table 3. Truth table with rules and activation functions.

Sets Associated to the Input						Output Activation Functions					
A_1	A_2	$\cdots$	A_j	$\cdots$	A_p	Y_1	Y_2	$\cdots$	Y_m	$\cdots$	Y_w
$a_{1,1}$	$a_{1,2}$	$\cdots$	$a_{1,j}$	$\cdots$	$a_{1,p}$	$f_{1,1}$	$f_{1,2}$	$\cdots$	$f_{1,m}$	$\cdots$	$f_{1,w}$
$a_{2,1}$	$a_{2,2}$	$\cdots$	$a_{2,j}$	$\cdots$	$a_{2,p}$	$f_{2,1}$	$f_{2,2}$	$\cdots$	$f_{2,m}$	$\cdots$	$f_{2,w}$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$
$a_{k,1}$	$a_{k,2}$	$\cdots$	$a_{k,j}$	$\cdots$	$a_{k,p}$	$f_{k,1}$	$f_{k,2}$	$\cdots$	$f_{k,m}$	$\cdots$	$f_{k,w}$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$
$a_{q,1}$	$a_{q,2}$	$\cdots$	$a_{q,j}$	$\cdots$	$a_{q,p}$	$f_{q,1}$	$f_{q,2}$	$\cdots$	$f_{q,m}$	$\cdots$	$f_{q,w}$

In Table 3, variables $a_{k,j}$ and $f_{k,m}$ represent relationships between A_j and Y_m. Value of variables in the Boolean case are $\{0, 1\}$; in Kleene's case the values are $\{0, u, 1\}$. Table 3 can be a Boolean or Kleene truth table [56]. It is noteworthy that A_j can be a Boolean or fuzzy set associated with the input, and Y_m is an activation function associated with the output.

The combination of sets A_j when using a disjunctive normal form can be expressed as:

$$\bigwedge_{j=1}^{p} \widehat{A}(a_{k,j}, f_{k,m}) \tag{18}$$

where $\widehat{A}$ is a function that can be 0, 1, A_j, $\bar{A}_j$ or $A_j \wedge \bar{A}_j$, it depends on $a_{k,j}$ and $f_{k,m}$ respectively [22,45,56]. Table 4 shows the values of $\widehat{A}$ for the case of Kleene or Boole.

Table 4. Function values for $\widehat{A}$.

$a_{k,j}$	$f_{k,m}$	$\widehat{A}$
0	0	0
u	0	0
1	0	0
0	u	$\bar{A}_j$
u	u	$A_j \wedge \bar{A}_j$
1	u	A_j
0	1	$\bar{A}_j$
u	1	1
1	1	A_j

The m-th activation output Y_m, using the DNF can be represented as:

$$Y_m = \bigvee_{k=1}^{q} \bigwedge_{j=1}^{p} \widehat{A}(a_{k,j}, f_{k,m}) \tag{19}$$

Considering the activation outputs of Table 3, the total system output is calculated as:

$$y = \sum_{m=1}^{w} Y_m v_m \tag{20}$$

where v_m corresponds to m-th virtual actuator; then, it is obtained the m-th virtual output:

$$y_m = Y_m v_m \tag{21}$$

In Equation (21) $Y_m \in [0,1]$ is the activation function and $v_m \in \mathbb{R}$ is the value of the virtual actuator. In control systems, the actuator is used to apply the control action [22]; in some applications (with Boolean logic) it is possible to use several actuators for the control action (several batteries, valves, switches, etc.). However, when there is only one action element, this can be considered as the sum of several actuators called "virtual".

In control applications, the value of v_m is associated with the value (range) of the actuator utilized, for example, in the application of the hydraulic system described in [50], v_m corresponds to the valve's maximum flow for filling the tank. In general, for neuro-fuzzy applications the value of v_m can be an adaptation parameter when the optimization process of the fuzzy system is carried out.

It is noticeable that when using Kleene algebra in some cases it is possible to have a simplified activation Y_m with a smaller number of literals and connectors. This is the principle applied to have compact fuzzy systems based on Boolean relations.

2.5. Extension of Boolean Tables to Kleene Tables

For the conversion of a Boolean table to a Kleene table, two aspects must be considered. First, the regularity conditions that allow handling the ambiguity. The second are the aspects for encoding three elements table which are the monotonous transitions between Boolean actions.

2.5.1. Regularity Conditions

According to [58,59], regularity conditions allow handling ambiguity, from a ternary logic view. The regularity conditions indicate that if the ambiguity at the input increases, then the ambiguity at the output also increases.

Definition 4 ([58,59])**.** *Considering $E = \{0, u, 1\}$ a n-variable in ternary logic, then, a function f mapped from E^n to E such that $X \in E^n$ assigns $f(X) \in E$. Considering Figure 1 a relationship* *of ambiguity $\preceq$ corresponds to:*

$$0 \preceq u, \quad 1 \preceq u, \quad i \preceq i, \tag{22}$$

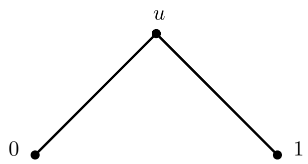

Figure 1. Ambiguity partial order $\preceq$ [59].

The above means that 0 and 1 are less ambiguous than u, if $i \preceq j$, then i is more ambiguous than j. In addition, for different elements $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ of E^n, then X is less ambiguous than Y if $x_i \preceq y_i$ for $i = 1, 2, \ldots, n$, namely $X \preceq Y$.

Considering that 0 and 1 are defined states and u is not state defined as 0 or 1. Then the partial order $\preceq$ describe ambiguity [59].

Definition 5 ([58,59])**.** *A function f is regular if it meets the condition of* ambiguity monotonicity *given by:*

$$f(A) \preceq f(B) \text{ when } A \preceq B \quad (A, B \in E^n). \tag{23}$$

This condition implies that if A is equal to or less ambiguous than B, then, $f(A)$ is equal to or less ambiguous than $f(B)$. For example, with (x_1, x_2) from Figure 2; for combinations $(1,1)$ and $(u,1)$, it is that $(1,1) \preceq (u,1)$; therefore, $f(1,1) \preceq f(u,1)$, in this way, if $f(1,1) = 1$ implies that $f(u,1)$ can be 1 or u in order to $f(x_1, x_2)$ be a regular function.

x_1 \ x_2	0	u	1
0	$f(0,0)$	$f(0,u)$	$f(0,1)$
u	$f(u,0)$	$f(u,u)$	$f(u,1)$
1	$f(1,0)$	$f(1,1)$	$f(1,1)$

Figure 2. Example of a three-element table.

2.5.2. Monotonous Transitions

For the extension of a Boolean table to a Kleene table, it is first necessary to meet the condition (23) to handle the ambiguity and secondly, the suitable configuration to achieve monotonous transitions between Boolean actions [50].

Definition 6. *The "Boolean actions" can be defined as the actions obtained by the activation function with the respective virtual actuator considering the respective Boolean encoding, thus the m-th Boolean action corresponds to:*

$$y_m = Y_m v_m \tag{24}$$

Definition 7. *The monotonic transitions between Boolean relations correspond to the condition to have a continuous and monotonic transition from y_m to y_n which are two Boolean actions.*

Kleene table is obtained by adding intermediate cases to the Boolean table, that is, u in the antecedent and $\{0, u, 1\}$ in the consequent according to the previous considerations.

An example of the extension of a table of two elements to one of three elements considering the regularity conditions can be seen in Figure 3. In order to meet the condition (23), two options are possible; however; the last configuration is used to achieve monotonous transitions between Boolean actions.

x_1	x_2	f
0	0	1
1	0	0
0	1	0
1	1	0

$\rightarrow$

$x_1 \backslash x_2$	0	1
0	1	0
1	0	0

$\rightarrow$

$x_1 \backslash x_2$	0	u	1
0	1	u	0
u	u	u	$0, u$
1	0	$0, u$	0

$\rightarrow$

$x_1 \backslash x_2$	0	u	1
0	1	u	0
u	u	u	0
1	0	0	0

Figure 3. Example of the extension from bivalent to trivalent tables meeting regularity conditions.

3. Compact Fuzzy Systems Based on Boolean Relations

Compact fuzzy systems based on Boolean relations can be established when applying the procedure of terms simplification employing Kleene's algebra. Thus, compact fuzzy systems are obtained when an activation function has a single inference rule since disjunction operation is eliminated. It should be noted that by applying the respective simplifications through Kleene algebra, an expression equivalent to the simplification performed with Boolean algebra can be obtained.

According to the possible configurations in the truth table (Boolean case) the following cases are considered:

- Y_U: Activation function with unitary dependence. The coding is presented in Table 5.
- Y_T: Activation function with total dependence. Respective coding in Table 6.

On Tables 5 and 6 the cells marked with X indicate that the Boolean variable can be 1 or 0. In Table 5, for the function Y_U the action is found when A_m is "activated", that is, it only depends on a set of the input. Meanwhile, in Table 6 for Y_T there is an action when A_1, $A_2 \dots A_n$ are activated simultaneously; therefore, it depends on all sets of the input.

Table 5. Coding for activation with a set at the input.

A_1	$\cdots$	A_m	$\cdots$	A_{n-1}	A_n	Y_U
X	$\cdots$	1	$\cdots$	X	X	1
X	$\cdots$	0	$\cdots$	X	X	0

Table 6. Encoding for activation with all sets on input.

A_1	A_2	$\cdots$	A_{n-1}	A_n	Y_T
1	1	$\cdots$	1	1	1
0	X	$\cdots$	X	X	0
X	0	$\cdots$	X	X	0
X	X	$\cdots$	0	X	0
X	X	$\cdots$	X	0	0

In order to establish the equations for the cases considered, the procedure to obtain the terms of a function in the disjunctive normal form is used. Depending on the case, the simplification is done using Kleene or Boole algebras.

3.1. Equations with Boolean Algebra

Considering the procedure for expressing the functions in disjunctive normal form and using Boolean algebra, the equation for Y_U corresponds to:

$$Y_U = A_1 \wedge (A_2 \vee \bar{A}_2) \wedge \cdots \wedge (A_{n-1} \vee \bar{A}_{n-1}) \wedge (A_n \vee \bar{A}_n) \tag{25}$$

$$Y_U = A_1 \tag{26}$$

meanwhile, the expression for Y_T corresponds to:

$$Y_T = A_1 \wedge A_2 \wedge \cdots \wedge A_{n-1} \wedge A_n \quad (27)$$

3.2. Boolean Table Extension to Kleene Table

To extend a Boolean table into a Kleene table, it is noteworthy that the cases marked with X can be 1, 0, or u. In order to comply with the conditions of regularity and monotonous transitions between Boolean actions, the following considerations are taken:

- For the cases that are added where there is no transition of the output, the value must be kept 1 or 0 according to the case.
- For additional cases where there is a transition between the output states, the value of u is placed in the output.

3.3. Activation Function with Unitary Dependence

In the case of an activation function with unit dependency, Table 7 shows the extent of the table from Boole to Kleene, where X can be $\{1, u, 0\}$.

Table 7. Kleene extension of coding for activation with a set on input.

A_1	$\cdots$	A_m	$\cdots$	A_{n-1}	A_n	Y_U
X	$\cdots$	1	$\cdots$	X	X	1
X	$\cdots$	u	$\cdots$	X	X	u
X	$\cdots$	0	$\cdots$	X	X	0

In this extension, to have a regular table and maintain a monotonous transition Y_U is 1 for any case that $A_m = 1$, it is also u when $A_m = u$ and 0 for $A_m = 0$. As observed, when there is a transition of Y_U between 0 and 1 is assigned u.

Considering the rules to establish the disjunctive normal form, it can be seen that A_m appears in the terms when Y_U is 1 or u; therefore, the equation can be written as:

$$Y_U = (A_m \wedge F_B) \vee (A_m \wedge F_E) \vee A_m = A_m \quad (28)$$

where:

- F_B: corresponds to the disjunction of all the conjunctions obtained in the truth table for which Y_U is 1 eliminating from these the variable A_m.
- F_E: corresponds to the disjunction of all the conjunctions obtained in the truth table for which Y_U is u eliminating from these the variable A_m.

As seen, when having a value of 1 in the column for Y_U, the conjunction where the variable A_m is 1 and the rest of variables u it is obtained the variable A_m, hence when this variable occurs in the rest of the terms, an absorption process happens since it appears in all the terms. That is, for all conjunctions when Y_U is 1 always appears A_m and when Y_U is u it is always found the conjunction $A_m \wedge \bar{A}_m$.

3.3.1. Example for a Two-Variable Table

Table 8, which a truth table of two variables, serves as a first example to show the design of these systems.

Table 8. Boolean relations for two variables.

A_1	A_2	f_U
0	0	0
0	1	0
1	0	1
1	1	1

The simplification using Boolean algebra is:

$$f_U = (A_1 \wedge A_2) \vee (A_1 \wedge \bar{A}_2) = A_1 \wedge (A_2 \vee \bar{A}_2) = A_1 \tag{29}$$

It is noteworthy that the previous procedure is not fulfilled when employing Kleene algebra since $A_2 \vee \bar{A}_2 \leq 1$; however, with Kleene's algebra by including the variable u can be obtained the equivalent result. By using the codification of Table 9, there is also an equivalence with the formula employing Boolean algebra; besides, it also meets the regularity condition.

Table 9. Kleene extension coding for two variables.

A_1	A_2	f_U
0	0	0
0	u	0
0	1	0
1	1	1
1	0	1
1	u	1
u	0	u
u	1	u
u	u	u

Another way to represent the extended table is shown in Figure 4.

A_2 \ A_1	0	u	1
0	0	u	1
u	0	u	1
1	0	u	1

Figure 4. Alternative Kleene encoding representation.

Applying Kleene algebra is obtained:

$$\begin{aligned} f_U &= (A_1 \wedge A_2) \vee (A_1 \wedge \bar{A}_2) \vee A_1 \\ &\quad \vee (A_1 \wedge \bar{A}_1 \wedge \bar{A}_2) \vee (A_1 \wedge \bar{A}_1 \wedge A_2) \vee (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge \bar{A}_2) \\ &= A_1 \wedge (A_2 \vee \bar{A}_2) \vee A_1 \vee A_1 \wedge [(\bar{A}_1 \wedge \bar{A}_2) \vee (\bar{A}_1 \wedge A_2) \vee (\bar{A}_1 \wedge A_2 \wedge \bar{A}_2)] \\ &= A_1 \end{aligned} \tag{30}$$

This expression can be written as:

$$f_U = [A_1 \wedge F_B(A_2, \bar{A}_2)] \vee [A_1 \wedge F_E(\bar{A}_1, A_2, \bar{A}_2)] \vee A_1 = A_1 \tag{31}$$

3.3.2. Example for a Table of Three Variables

Table 10 is employed as a second example that corresponds to a truth table of three elements.

Table 10. Boolean relations for three variables.

A_1	A_2	A_3	f_U
0	X	X	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

The simplification using Boolean algebra is:

$$\begin{aligned} f_U &= (A_1 \wedge \bar{A}_2 \wedge \bar{A}_3) \vee (A_1 \wedge A_2 \wedge \bar{A}_3) \vee \\ &\quad (A_1 \wedge \bar{A}_2 \wedge A_3) \vee (A_1 \wedge A_2 \wedge A_3) \\ &= A_1 \wedge [(\bar{A}_2 \wedge \bar{A}_3) \vee (A_2 \wedge \bar{A}_3) \vee (\bar{A}_2 \wedge A_3) \vee (A_2 \wedge A_3)] \\ &= A_1 \wedge [(\bar{A}_3 \wedge (\bar{A}_2 \vee A_2)) \vee (A_3 \wedge (\bar{A}_2 \vee A_2))] \\ &= A_1 \wedge (\bar{A}_3 \vee A_3) \wedge (\bar{A}_2 \vee A_2) \\ &= A_1 \end{aligned} \tag{32}$$

Considering the third element u when building the Kleene table, the configuration displayed in Table 11 is found.

Table 11. Kleene relations for three variables.

A_1	A_2	A_3	f_U	A_1	A_2	A_3	f_U	A_1	A_2	A_3	f_U
1	0	0	1	u	0	0	u	0	0	0	0
1	0	1	1	u	0	1	u	0	0	1	0
1	1	0	1	u	1	0	u	0	1	0	0
1	1	1	1	u	1	1	u	0	1	1	0
1	0	u	1	u	0	u	u	0	0	u	0
1	1	u	1	u	1	u	u	0	1	u	0
1	u	0	1	u	u	0	u	0	u	0	0
1	u	1	1	u	u	1	u	0	u	1	0
1	u	u	1	u	u	u	u	0	u	u	0

Alternatively, the relationship between the variables can be shown using the coding in Figure 5.

$A_3 = 0$

A_2 \ A_1	0	u	1
0	0	u	1
u	0	u	1
1	0	u	1

$A_3 = u$

A_2 \ A_1	0	u	1
0	0	u	1
u	0	u	1
1	0	u	1

$A_3 = 1$

A_2 \ A_1	0	u	1
0	0	u	1
u	0	u	1
1	0	u	1

Figure 5. Three-variable relationships for f_U.

Applying Kleene algebra for f_U is obtained:

$$\begin{aligned} f_U &= (A_1 \wedge \bar{A}_2 \wedge \bar{A}_3) \vee (A_1 \wedge \bar{A}_2 \wedge A_3) \vee (A_1 \wedge A_2 \wedge \bar{A}_3) \vee (A_1 \wedge A_2 \wedge A_3) \vee \\ &\quad (A_1 \wedge \bar{A}_2) \vee (A_1 \wedge A_2) \vee (A_1 \wedge \bar{A}_3) \vee (A_1 \wedge A_3) \vee A_1 \vee \\ &\quad (A_1 \wedge \bar{A}_1 \wedge \bar{A}_2 \wedge \bar{A}_3) \vee (A_1 \wedge \bar{A}_1 \wedge \bar{A}_2 \wedge A_3) \vee (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge \bar{A}_3) \vee (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge A_3) \vee \\ &\quad (A_1 \wedge \bar{A}_1 \wedge \bar{A}_2 \wedge A_3 \wedge \bar{A}_3) \vee (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge A_3 \wedge \bar{A}_3) \vee \\ &\quad (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge \bar{A}_2 \wedge \bar{A}_3) \vee (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge \bar{A}_2 \wedge A_3) \vee (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge \bar{A}_2 \wedge A_3 \wedge \bar{A}_3) \\ f_U &= A_1 \wedge [(\bar{A}_2 \wedge \bar{A}_3) \vee (\bar{A}_2 \wedge A_3) \vee (A_2 \wedge \bar{A}_3) \vee (A_2 \wedge A_3) \vee \bar{A}_2 \vee A_2 \vee \bar{A}_3 \vee A_3] \vee A_1 \vee \\ &\quad A_1 \wedge [(\bar{A}_1 \wedge \bar{A}_2 \wedge \bar{A}_3) \vee (\bar{A}_1 \wedge \bar{A}_2 \wedge A_3) \vee (\bar{A}_1 \wedge A_2 \wedge \bar{A}_3) \vee (\bar{A}_1 \wedge A_2 \wedge A_3) \vee \\ &\quad (\bar{A}_1 \wedge \bar{A}_2 \wedge A_3 \wedge \bar{A}_3) \vee (\bar{A}_1 \wedge A_2 \wedge A_3 \wedge \bar{A}_3) \vee \\ &\quad (\bar{A}_1 \wedge A_2 \wedge \bar{A}_2 \wedge \bar{A}_3) \vee (\bar{A}_1 \wedge A_2 \wedge \bar{A}_2 \wedge A_3) \vee (\bar{A}_1 \wedge A_2 \wedge \bar{A}_2 \wedge A_3 \wedge \bar{A}_3)] \end{aligned} \tag{33}$$

For this case the expression can be written as:

$$f_U = (A_1 \wedge F_B) \vee (A_1 \wedge F_E) \vee A_1 = A_1 \tag{34}$$

3.4. Activation Function with Total Dependence

In this case, activation is performed by all input sets. Table 12 shows the extent of the table from Boole to Kleene where X can be $\{1, u, 0\}$ and $X_{u,1}$ the elements $\{u, 1\}$.

Table 12. Kleene extension of the encoding for activation with all sets on input.

A_1	A_2	$\cdots$	A_{n-1}	A_n	Y_T
1	1	$\cdots$	1	1	1
u	$X_{u,1}$	$\cdots$	$X_{u,1}$	$X_{u,1}$	u
$X_{u,1}$	u	$\cdots$	$X_{u,1}$	$X_{u,1}$	u
$X_{u,1}$	$X_{u,1}$	$\cdots$	u	$X_{u,1}$	u
$X_{u,1}$	$X_{u,1}$	$\cdots$	$X_{u,1}$	u	u
0	X	$\cdots$	X	X	0
X	0	$\cdots$	X	X	0
X	X	$\cdots$	0	X	0
X	X	$\cdots$	X	0	0

In this table, if any of the variables A_m is 0 then $Y_T = 0$, if all variables A_m are 1, then, $Y_T = 1$; finally, if any of the variables A_m is u and the other ones u or 1, then $Y_T = u$. Thus, the cases where there is a transition from Y_T between 0 and 1 they are assigned u.

Using the rules to determine the disjunctive normal form, it is observed that the term $A_1 \wedge A_m \wedge \ldots \wedge A_n$ appears for the expressions when Y_T is 1 or u. In general Y_T can be written as:

$$Y_T = (A_1 \wedge A_m \wedge \ldots \wedge A_n) \vee (A_1 \wedge A_m \wedge \ldots \wedge A_n \wedge F_E) \tag{35}$$

$$Y_T = A_1 \wedge A_m \wedge \ldots \wedge A_n$$

where F_E corresponds to the disjunction of all the conjunctions obtained in the truth table for which Y_T is u eliminating from these the conjunction $A_1 \wedge A_m \wedge \ldots \wedge A_n$.

It is important to note the term $A_1 \wedge A_m \wedge \ldots \wedge A_n$ is presented in the terms calculated for Y_T when is u, hence when the absorption process is carried out, it is obtained the expression for Y_T.

Example for a Table of Three Variables

In order to show the case when having an activation function with total dependency, it is used the coding in Table 13.

Table 13. Boolean relations for three variables.

A_1	A_2	A_3	f_T
0	X	X	0
X	0	X	0
X	X	0	0
1	1	1	1

The respective expression using Boolean algebra is:

$$f_T = A_1 \wedge A_2 \wedge A_3 \tag{36}$$

Table 14 displays the configuration considering the third element u to build the Kleene table.

Alternatively, the relationship between variables is described by the configuration present in Figure 6.

Table 14. Kleene relations for three variables.

A_1	A_2	A_3	f_T	A_1	A_2	A_3	f_T	A_1	A_2	A_3	f_T
1	0	0	0	u	0	0	0	0	0	0	0
1	0	1	0	u	0	1	0	0	0	1	0
1	1	0	0	u	1	0	0	0	1	0	0
1	1	1	1	u	1	1	u	0	1	1	0
1	0	u	0	u	0	u	0	0	0	u	0
1	1	u	u	u	1	u	u	0	1	u	0
1	u	0	0	u	u	0	0	0	u	0	0
1	u	1	u	u	u	1	u	0	u	1	0
1	u	u	u	u	u	u	u	0	u	u	0

$A_3 = 0$

$A_2 \backslash A_1$	0	u	1
0	0	0	0
u	0	0	0
1	0	0	0

$A_3 = u$

$A_2 \backslash A_1$	0	u	1
0	0	0	0
u	0	u	u
1	0	u	u

$A_3 = 1$

$A_2 \backslash A_1$	0	u	1
0	0	0	0
u	0	u	u
1	0	u	1

Figure 6. Three-variable relationships for f_T.

Applying Kleene algebra for f_T is obtained:

$$\begin{aligned} f_T &= (A_1 \wedge A_2 \wedge A_3) \vee (A_1 \wedge A_2 \wedge A_3 \wedge \bar{A}_3) \vee (A_1 \wedge A_2 \wedge \bar{A}_2 \wedge A_3) \vee (A_1 \wedge A_2 \wedge \bar{A}_2 \wedge A_3 \wedge \bar{A}_3) \vee \\ &\quad (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge A_3) \vee (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge A_3 \wedge \bar{A}_3) \vee (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge \bar{A}_2 \wedge A_3) \vee \\ &\quad (A_1 \wedge \bar{A}_1 \wedge A_2 \wedge \bar{A}_2 \wedge A_3 \wedge \bar{A}_3) \\ f_T &= (A_1 \wedge A_2 \wedge A_3) \vee [(A_1 \wedge A_2 \wedge A_3) \wedge [\bar{A}_3 \vee \bar{A}_2 \vee (\bar{A}_2 \wedge \bar{A}_3) \vee \\ &\quad \bar{A}_1 \vee (\bar{A}_1 \wedge \bar{A}_3) \vee (\bar{A}_1 \wedge \bar{A}_2) \vee (\bar{A}_1 \wedge \bar{A}_2 \wedge \bar{A}_3)]] \end{aligned} \tag{37}$$

In this case, the equation is written as:

$$f_T = (A_1 \wedge A_2 \wedge A_3) \vee (A_1 \wedge A_2 \wedge A_3 \wedge F_E) \tag{38}$$

therefore:

$$f_T = A_1 \wedge A_2 \wedge A_3 \tag{39}$$

4. Compact Fuzzy System Architectures Based on Boolean Relations

Considering the analysis carried out in Section 3, the compact configurations are obtained for two particular cases similar to classical neuro-fuzzy structures [22]; however, structures ad hoc can also be obtained with the proposed design methodology.

For the development of compact fuzzy systems based on Boolean relations, the following configurations of the activation functions are considered:

- Architecture I: Employing activation functions with total dependence. Activation of each virtual actuator is performed by all input sets. Similar to the scheme used in radial-based neural networks.
- Architecture II: Using activation functions with unit dependency. Each virtual actuator is activated by a single input set. It can be directly related to the structure of a dynamic discrete time system.

Regarding possible applications, architecture I allows relationships between inputs to obtain the output; therefore, it can be used in identification of dynamic systems where the existence of products between the input variables is observed, as shown in Section 5. On the other hand, structure II permits the definition of direct analogies with linear controllers to increase the adaptation capacity, as shown in the example of Section 6.

4.1. Compact System with Architecture I

In order to establish a system with architecture I, the coding shown in Table 15 is used.

Table 15. Coding for compact systems with architecture I.

$A_{1,m}(x_1)$	$A_{2,m}(x_2)$	$\cdots$	$A_{p,m}(x_p)$	Y_m
1	1	$\cdots$	1	1
0	X	$\cdots$	X	0
X	0	$\cdots$	X	0
X	X	$\cdots$	0	0

According to the previous analysis, from Table 15 is found that:

$$Y_m = A_{1,m}(x_1) \wedge A_{2,m}(x_2) \wedge \cdots \wedge A_{p,m}(x_p) \tag{40}$$

For the implementation using fuzzy sets the conjunction is made with the product t-norm and each set $A_{p,m}$ has a membership function associated $\mu_{p,m}$. Taking $m = 1$ a partial virtual output of this inference process can be seen in Figure 7, where v_1 corresponds to the respective virtual actuator.

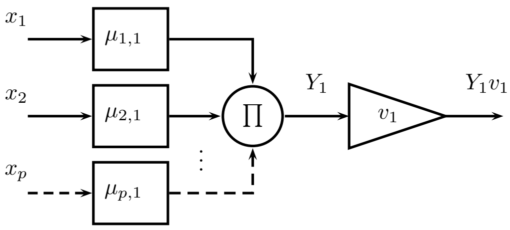

Figure 7. Partial virtual output of the inference process.

Using M virtual actuators with their respective activation functions, it is found the scheme shown in Figure 8. A constant virtual actuator is used to complement the design v_c which corresponds to an encoding with an activation function Y_c always active, that is, with values of 1 in their respective column.

In this way, the expression for the inference process of this system is:

$$f = v_c + \sum_{m=1}^{M} v_m \left[\prod_{p=1}^{N} \mu_{p,m}(x_p) \right] \tag{41}$$

This scheme can be employed to systems identification since it has a structure similar to that conventionally used in radial-based neural networks when used to $\mu_{p,m}$ Gaussian membership functions [60].

The structure of Figure 8 can be interpreted as a neuro-fuzzy network. The respective membership value for each input is obtained in the first layer. The second layer corresponds to the product of the values obtained from the membership functions, that is, the calculation of Y_m, while $Y_m v_m$ (virtual actuators multiplied by the respective activation functions) is obtained in the third layer. Finally, in layer four it is performed the summation of all $Y_m v_m$ obtaining the inference output f.

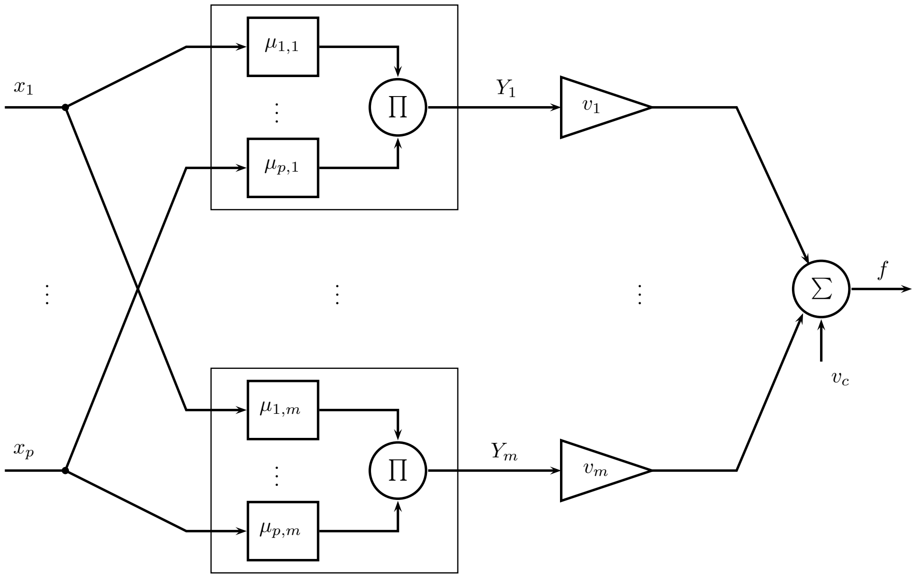

Figure 8. Representation of a compact system with architecture I.

Application for Dynamic Systems

In order to carry out the application of this configuration for dynamic systems, delays can be used for the input signal and the feedback of the system output. As example, Figure 9 shows a configuration with one input delay and two for the output feedback. This fuzzy system can be used for identification or control.

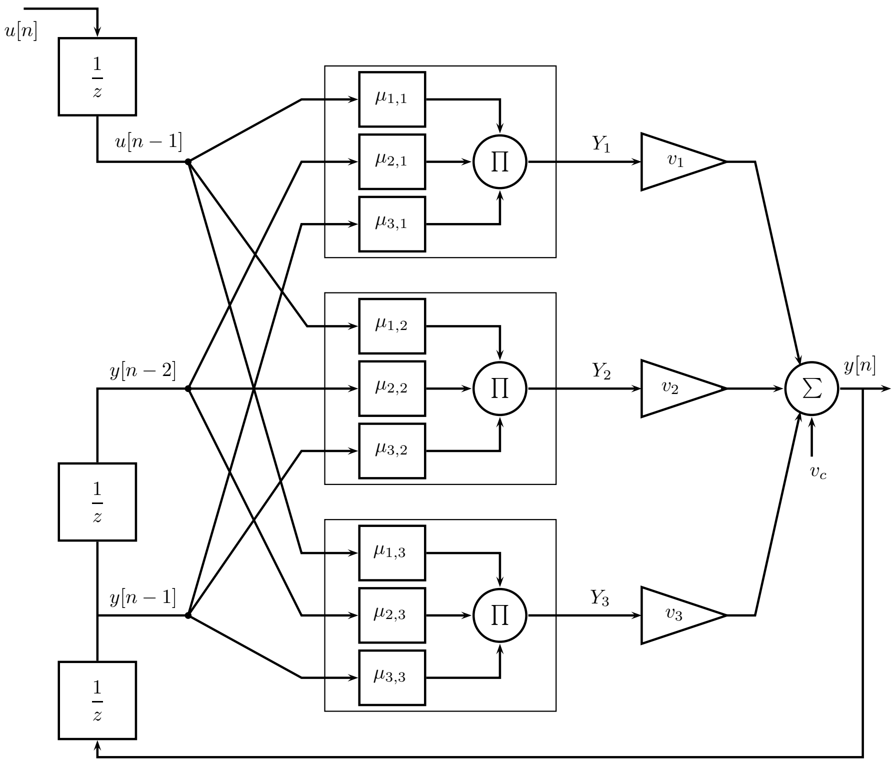

Figure 9. Configuration of the neuro-fuzzy systems for identification and control.

4.2. Compact System with Architecture II

In this case, the coding of Table 16 is employed to find a fuzzy system where the activation functions depend on a single input set.

Table 16. Partial coding for a compact system with architecture II.

$A_{1,1}(x_1)$	$A_{1,2}(x_1)$	$\cdots$	$A_{1,i}(x_1)$	$Y_{1,1}$
1	X	$\cdots$	X	1
0	X	$\cdots$	X	0

From Table 16 is obtained $Y_{1,1} = A_{1,1}(x_1)$. Generally increasing the columns, the activation outputs are written as:

$$Y_{1,i} = A_{1,i}(x_1) \tag{42}$$

Having for each set $A_{1,i}$ a membership function $\mu_{1,i}$, and the respective virtual actuator $v_{1,i}$, then it is obtained the partial output y_1 shown in Figure 10.

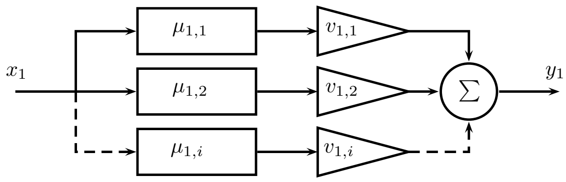

Figure 10. Configuration example for a partial output.

Extending Table 16 for more input variables x_1, x_2,..., x_j, it is determined the coding of Table 17.

Table 17. General coding for a compact system with architecture II.

$A_{j,1}(x_j)$	$A_{j,2}(x_j)$	$\cdots$	$A_{j,i}(x_j)$	$Y_{j,1}$	$Y_{j,2}$	$\cdots$	$Y_{j,i}$
1	X	$\cdots$	X	1	0	$\cdots$	0
X	1	$\cdots$	X	0	1	$\cdots$	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
X	X	$\cdots$	1	0	0	$\cdots$	1

In this way, each activation function $Y_{j,i}$ directly depends on a set $A_{j,i}$. The resulting expression is:

$$Y_{j,i} = A_{j,i}(x_j) \tag{43}$$

Using the respective membership functions $\mu_{j,i}$ associated with sets $A_{j,i}$. Figure 11 shows the general scheme obtained.

The output of the inference process can be calculated as:

$$f = \sum_{j=1}^{M} \sum_{i=1}^{N} v_{j,i}\mu_{j,i}(x_j) \tag{44}$$

As seen in Figure 11, the compact fuzzy system scheme can be interpreted as a neuro-fuzzy network since input and output relationships are obtained through membership functions and virtual actuators. The membership value for each input is obtained in the first layer. Then, this value is multiplied using respective virtual actuator $v_{j,i}$ in the second layer. In the third layer, the functions f_j associated to each input x_j are calculated. Finally, in this same layer performing the summation of all f_j the inference output f is determined.

In this way, the adaptation parameters correspond to the virtual actuators and the configuration values for the membership functions. This architecture can be used in identification and control since it allows for the representation of nonlinear relationships between the input and the output that can be associated with a dynamic system.

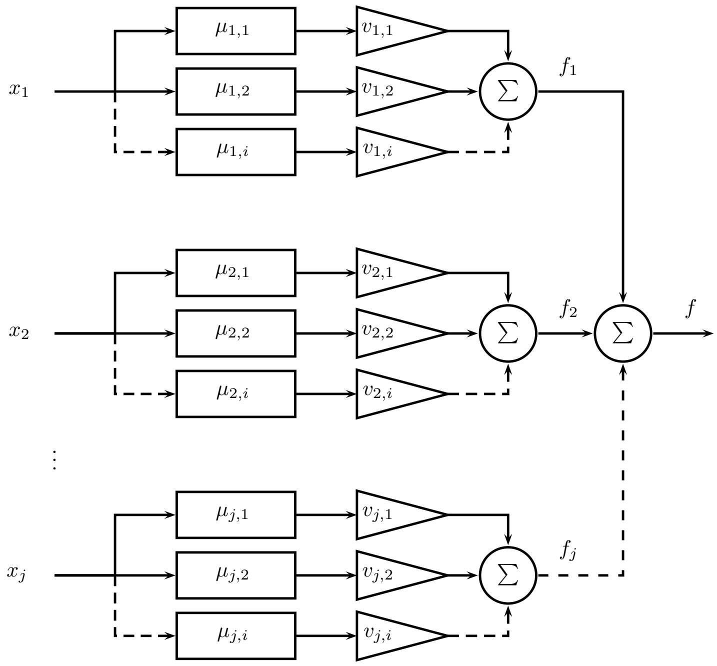

Figure 11. General diagram of a compact system with architecture II.

Application for Dynamic Systems

In this case, it is observed the direct relationship that a compact FIS-BBR has to represent a nonlinear discrete time dynamic system. This equivalence is observed when modifying a linear system of discrete time using fuzzy sets to represent the relationships of the system. The model is obtained by replacing the linear relationships present for each of the delays of both the input and the output feedback. Figure 12 shows the input and output of the linear and fuzzy systems.

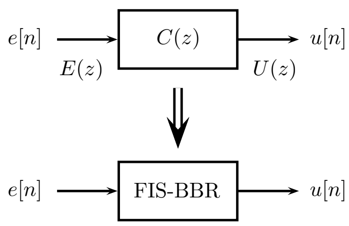

Figure 12. Linear system and fuzzy system.

In the first place, a linear system in discrete time is considered where its transfer function is:

$$C(z) = \frac{U(z)}{E(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{N_e} z^{-N_e}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + b_{N_u} z^{-N_u}} \tag{45}$$

Performing the respective operations:

$$\begin{aligned} U(z) + a_1 z^{-1} U(z) + a_2 z^{-2} U(z) + \cdots + b_{N_u} z^{-N_u} U(z) = \\ b_0 E(z) + b_1 z^{-1} E(z) + b_2 z^{-2} E(z) + \cdots + b_{N_e} z^{-N_e} E(z) \end{aligned} \tag{46}$$

The discrete time equation of this system is:

$$\begin{aligned} u[n] = b_0 e[n] + b_1 e[n-1] + b_2 e[n-2] + \cdots + b_p e[n-p] + \cdots + b_{N_e} e[n-N_e] \\ - a_1 u[n-1] - a_2 u[n-2] - \cdots - a_q u[n-q] - \cdots - a_{N_u} u[n-N_u] \end{aligned} \tag{47}$$

where the respective coefficients a_q, b_p are constant. In the fuzzy system these constants are replaced by nonlinear relationships, obtaining:

$$u[n] = f_{e,0}(e[n]) + f_{e,1}(e[n-1]) + f_{e,2}(e[n-2]) + \cdots + f_{e,N_e}(e[n-N_e]) \\ - f_{u,1}(u[n-1]) - f_{u,2}(u[n-2]) - \cdots - f_{u,N_u}(u[n-N_u]) \quad (48)$$

As example, Figure 13 is proposed for setting the fuzzy system with different membership functions; besides, system output is calculated as:

$$u = \sum_{j=1}^{M}\sum_{i=1}^{N} v_{j,i}\mu_{j,i}(x_j) = \sum_{p=0}^{N_e} f_{e,p}(e_p) + \sum_{q=0}^{N_u} f_{u,q}(u_q) \quad (49)$$

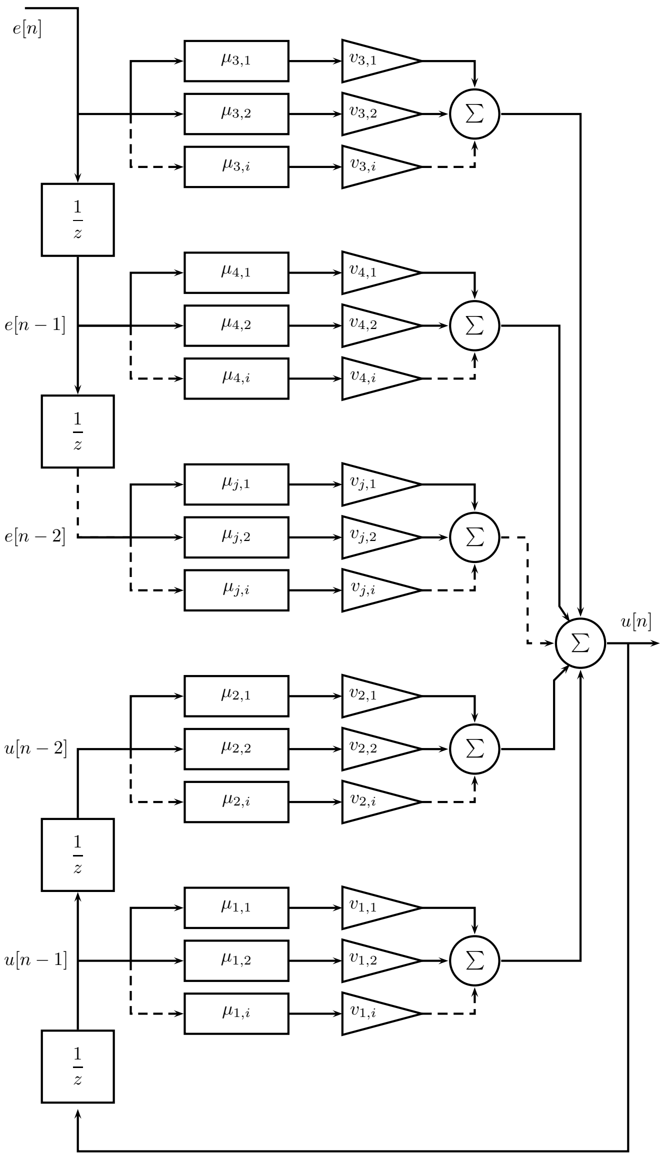

Figure 13. General scheme of the neuro-fuzzy system.

In order to consider nonlinearities (including saturation) for the functions $f_{e,p}$ and $f_{u,q}$, the relationship shown in the Figure 14 can be used.

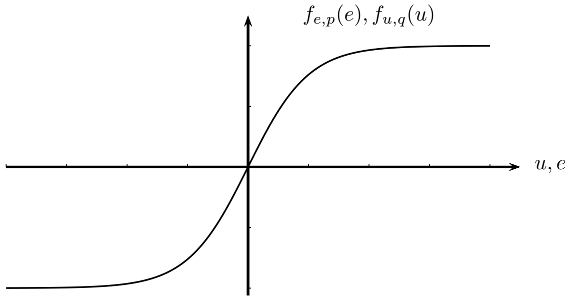

Figure 14. Nonlinear function to implement a discrete time dynamic system.

5. Application Example of Architecture I: Nonlinear Plant Identification

By the simplification procedure using Kleene's algebra described in Sections 2 and 3, the compact structures are established in Section 4; therefore, this section shows the application of structure I for the identification of a nonlinear system.

The identification process builds mathematical models of dynamic systems based on the real input observations, outputs, and disturbances. It also requires selecting the structure of the model [61,62].

The system given by Equation (50) is taken as a case study, which has been described in [22]. A stable functioning is guaranteed with $u[n] \in [-2, 2]$. The calculation of the training data is made using $y[0] = 0$, and $y[1] = 0$.

$$y[n+1] = \frac{y[n](y[n-1]+2)(y[n]+2.5)}{8.5 + y[n]^2 + y[n-1]^2} + u[n] \tag{50}$$

Figure 15 shows the architecture employed to identify the plant with one delay in the input and two for feedback; u represents the input while the output is y and z^{-1} represents a delay.

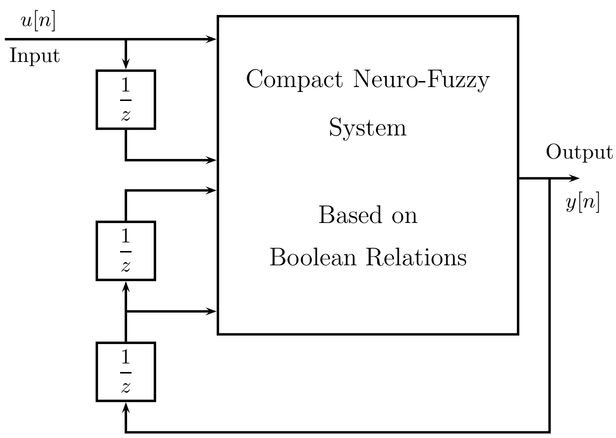

Figure 15. Model employed for the identification.

5.1. Compact Neuro-Fuzzy System

In this case, Gaussian memberships are used for the configuration in each $\mu_{i,l}$. Figure 16 is an example where $x_1, \ldots, x_i$ are inputs and $\mu_{1,l}, \ldots, \mu_{i,l}$, are the membership functions.

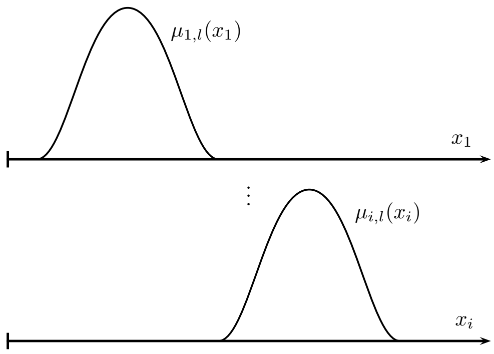

Figure 16. Gaussian sets for each input.

Considering the input, output, and delays, the structure of Figure 15 used in the identification process is determined.

The system output is calculated as:

$$f(x) = v_c + \sum_{l=1}^{M} v_l \left[\prod_{i=1}^{N} \exp\left(-\left(\frac{x_i - \delta_{i,l}}{\sigma_{i,l}} \right)^2 \right) \right] \tag{51}$$

the resulting activation outputs are:

$$Y_l = \prod_{i=1}^{N} \exp\left(-\left(\frac{x_i - \delta_{i,l}}{\sigma_{i,l}} \right)^2 \right) \tag{52}$$

System output is:

$$f = v_c + \sum_{l=1}^{M} v_l Y_l \tag{53}$$

where M represents the number of fuzzy rules, N number of inputs, v_l the actuator, v_c a constant action of the input, x_i is the system's input datum with $i = 1, 2, 3, \ldots, N$, while, $\delta_{i,l}$ and $\sigma_{i,l}$ are the center and the standard deviation of the Gaussian fuzzy set; finally, $f = y[n]$ is the fuzzy system's output.

The rules that implement the estructure of Figure 17 can be represented as shown in Boole Table 18. In general, for $l = 1, \ldots, 4$ these rules are described as:

- *If* $u[n]$ is $\mu_{1,l}$ and $u[n-1]$ is $\mu_{2,l}$ and $y[n-2]$ is $\mu_{3,l}$ and $y[n-1]$ is $\mu_{4,l}$ *Then* the activation function is Y_l.

Table 18. Rules associated with the estructure of Figure 17.

$\mu_{1,1}$	$\mu_{2,1}$	$\mu_{3,1}$	$\mu_{4,1}$	$\mu_{1,2}$	$\mu_{2,2}$	$\mu_{3,2}$	$\mu_{4,2}$	$\mu_{1,3}$	$\mu_{2,3}$	$\mu_{3,3}$	$\mu_{4,3}$	$\mu_{1,4}$	$\mu_{2,4}$	$\mu_{3,4}$	$\mu_{4,4}$	Y_1	Y_2	Y_3	Y_4	Y_c
1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	1	0	0	0	1
X	X	X	X	1	1	1	1	X	X	X	X	X	X	X	X	0	1	0	0	1
X	X	X	X	X	X	X	X	1	1	1	1	X	X	X	X	0	0	1	0	1
X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	0	0	0	1	1

In this way, the output is calculated using Equation (54), where v_c has associated an activation function always active $Y_c = 1$.

$$f = v_1 Y_1 + v_2 Y_2 + v_3 Y_3 + v_4 Y_4 + v_c \tag{54}$$

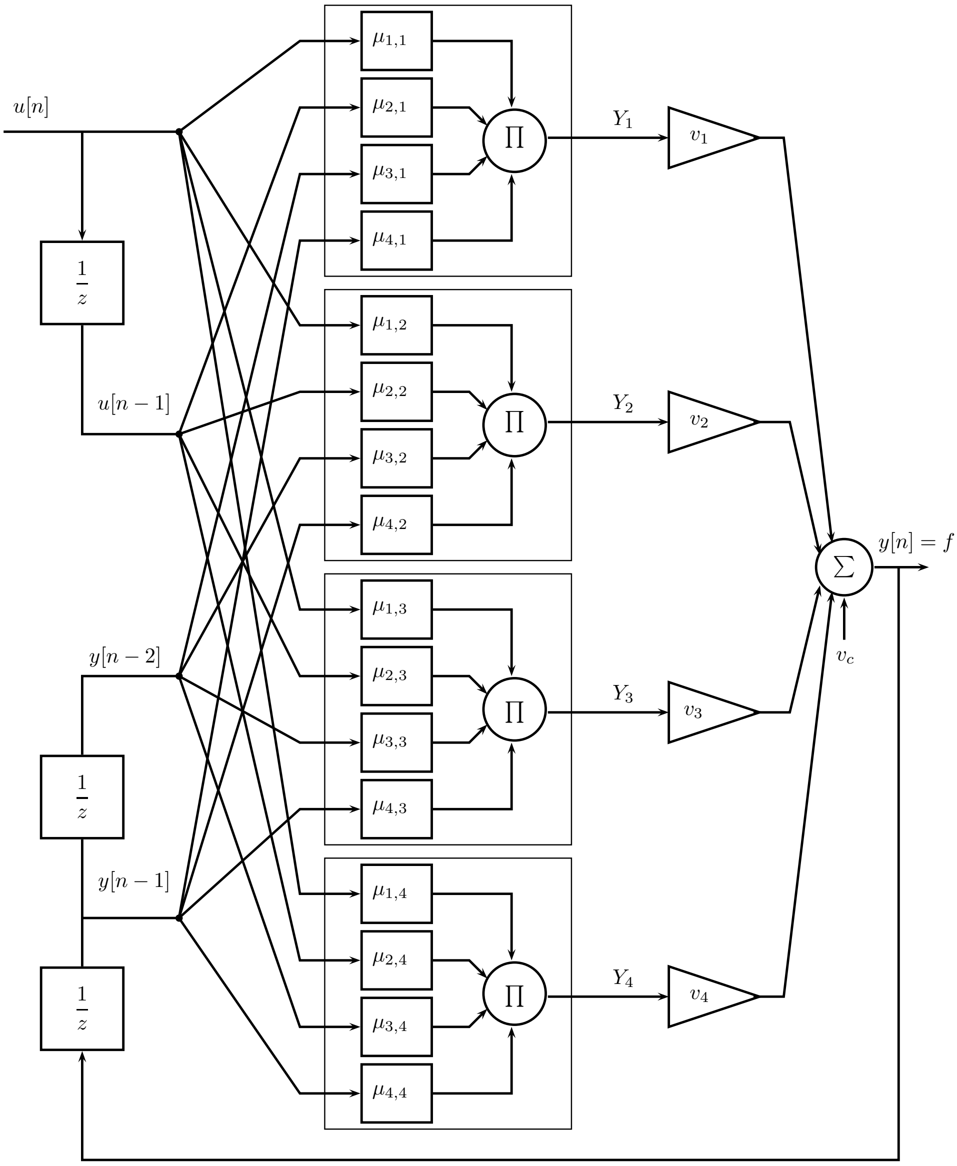

Figure 17. Representation of the neuro-fuzzy system employed.

5.2. Training Algorithm

For the system training, it is employed the Back Propagation (BP) algorithm that uses the gradients calculated in relation to the parameters [51]. This approach aims at minimizing the J_e error defined as:

$$J_e = \frac{1}{2}[y_r(p) - f(x(p))]^2 \tag{55}$$

where y_r is the real data, f the data from the fuzzy system, and p the variable associated with each training data. The equations used in the training process are:

$$v_c(q+1) = v_c(q) - \alpha \frac{\partial J_e}{\partial v_c}\bigg|_p \tag{56}$$

$$v_l(q+1) = v_l(q) - \alpha \frac{\partial J_e}{\partial v_l}\bigg|_p \tag{57}$$

$$\delta_{i,l}(q+1) = \delta_{i,l}(q) - \alpha \frac{\partial J_e}{\partial x_{i,l}}\bigg|_p \tag{58}$$

$$\sigma_{i,l}(q+1) = \sigma_{i,l}(q) - \alpha \frac{\partial J_e}{\partial \sigma_{i,l}}\bigg|_p \tag{59}$$

Thus, the parameters are upgraded with:

$$v_c(q+1) = v_c(q) - \alpha(f - y_r) \tag{60}$$

$$v_l(q+1) = v_l(q) - \alpha(f - y_r)Y_l \tag{61}$$

$$\delta_{i,l}(q+1) = \delta_{i,l}(q) - \alpha(f - y_r)v_l(q)Y_l \frac{2(x_i[p] - \delta_{i,l}(q))}{\sigma_{i,l}(q)^2} \tag{62}$$

$$\sigma_{i,l}(q+1) = \sigma_{i,l}(q) - \alpha(f - y_r)v_l(q)Y_l \frac{2(x_i[p] - \delta_{i,l}(q))^2}{\sigma_{i,l}(q)^3} \tag{63}$$

According to [51,63], parameters adaptation of the neuro-fuzzy system involves the following steps:

1. The neuro-fuzzy system is chosen considering M and N. According to [63], M is increased when having more parameters and, consequently, more calculations and higher accuracy.
2. Output calculations for an input-output $(x[p], y_r[p])$ in the q-th training stage.
3. Parameter upgrade $v_l(q+1)$, $\delta_{i,l}(q+1)$ and $\sigma_{i,l}(q+1)$. Where α is the learning rate.
4. Return to step 2 taking $q = q+1$, until the error $e = f - y_r[p]$ is smaller than a value ε, or until a maximum value of q.
5. Return to step 2 taking $p = p+1$, taking the next input-output $(x[p+1], y_r[p+1])$.

5.3. Training Process Results

Figure 18 displays the random data used (1000 samples). The training takes 70% of the data and to validate 30%. Figure 19 shows the result after training where the Mean Squared Error (MSE) is equal to 9.3592×10^{-4} using 200 iterations, while Figure 20 shows the validation process result with a MSE of 1.0×10^{-3}.

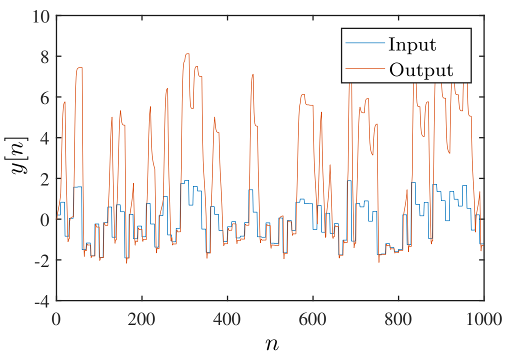

Figure 18. Plant data used.

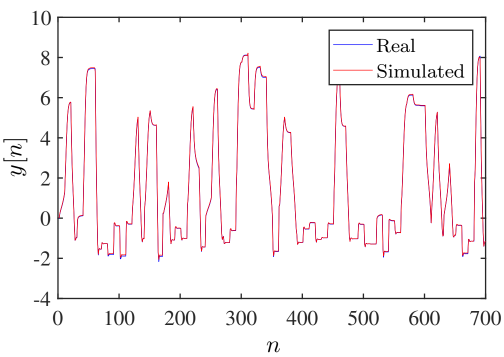

Figure 19. Training result.

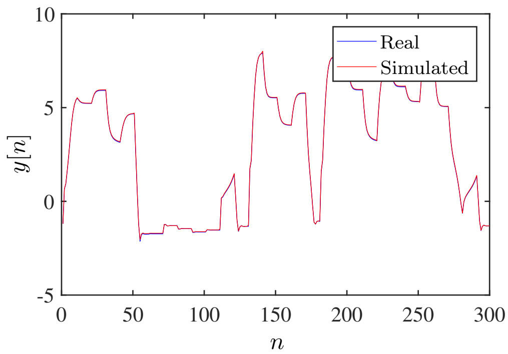

Figure 20. Validation result.

6. Application Example of Architecture II: PID Control

Fuzzy logic systems allow link-up system inputs and outputs using linguistic terms in the description [63]. This feature allows using fuzzy logic in control systems development [22]; in addition, the FIS systems allows handling nonlinearities like products between variables, saturations, and power functions, among others present in nonlinear dynamic systems [64]. This section aims to show that the II architecture established in Section 4 through the procedure of Section 3 and the fundamentals of Section 2 can be used to control systems. As a study case, it is shown the application of a hydraulic control system employing compact FIS-BBR with architecture II.

6.1. Hydraulic System

The considered model is like the one used in [50], consisting of a hydraulic system displayed in Figure 21. The system variables and parameters values are shown in Table 19. As can be observed in Equation (64), the model has a square root that incorporates a nonlinear component in the system behavior. In addition, there is a limitation in the maximum tank inflow.

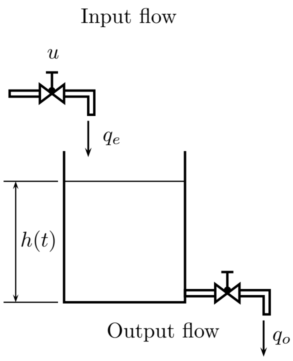

Figure 21. Tank filling system.

Table 19. System parameter values.

Parameter	Symbol	Value
Inflow	$q_e(t)$	0.25 L/s or 2.5×10^{-4} m^3/s (max)
Height of the liquid	$h(t)$	1 m (max)
Tank cross-sectional area	A	0.126 m^2
Output valve cross-sectional area	A_0	0.317 cm^2 (3.17×10^{-5} m^2)
Constant value of the valve	$C_v = A_0\sqrt{2g}$	1.4×10^{-4} $m^{5/2}/s$

The model of the hydraulic system is given by equation:

$$q_e(t) = A\frac{dh(t)}{dt} + C_v\sqrt{h(t)} \tag{64}$$

In addition, a delay time t_0 of 0. s is associated with the valve operation; therefore, the relation between control action u and the input flow is $q_e(t) = 2.5 \times 10^{-4} u(t - t_0)$ where $0 \leq u \leq 1$. Figure 22 shows the plant response for different values of the action $u(t - t_0)$ considering that the maximum value of h is 1 m.

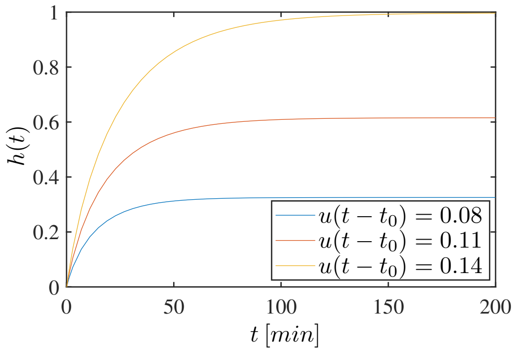

Figure 22. Plant response.

6.2. Controller

The controller is defined from a discrete-time linear controller transformed into a compact fuzzy shape when employing fuzzy sets. Then, the optimization of the controller parameters takes place with the plant and controller. Such optimization happens with the BFGS Quasi-Newton (Broyden–Fletcher–Goldfarb–Shanno) that explains the relevance of the initial search point [65].

A PID controller in discrete time is described by the transfer function:

$$C(z) = K_p + \frac{K_i}{1 - z^{-1}} + K_d(1 - z^{-1}) \tag{65}$$

performing the respective operations is obtained:

$$C(z) = \frac{(K_p + K_i + K_d) - (K_p + 2K_d)z^{-1} + K_d z^{-2}}{1 - z^{-1}} \tag{66}$$

in general, it can be written like:

$$C(z) = \frac{b_0 - b_1 z^{-1} + b_2 z^{-2}}{1 - z^{-1}} \tag{67}$$

The respective equation in discrete time for this controller is

$$u[n] = b_0 e[n] - b_1 e[n-1] + b_2 e[n-2] + u[n-1] \tag{68}$$

For the fuzzy controller, the expression is:

$$u[n] = f_1(e[n]) + f_2(e[n-1]) + f_3(e[n-2]) + f_4(u[n-1]) \tag{69}$$

For implementing the f_i sigmoidal functions are utilized to represent negative $\mu_{N,i}$ and positive $\mu_{P,i}$ values, as shown in Figure 23. Thus, Figure 24 shows the fuzzy compact PID controller.

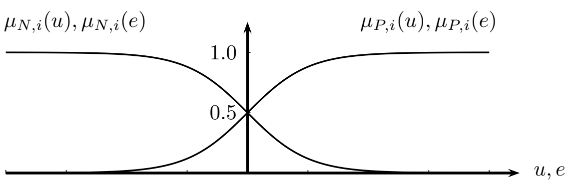

Figure 23. Membership functions employed.

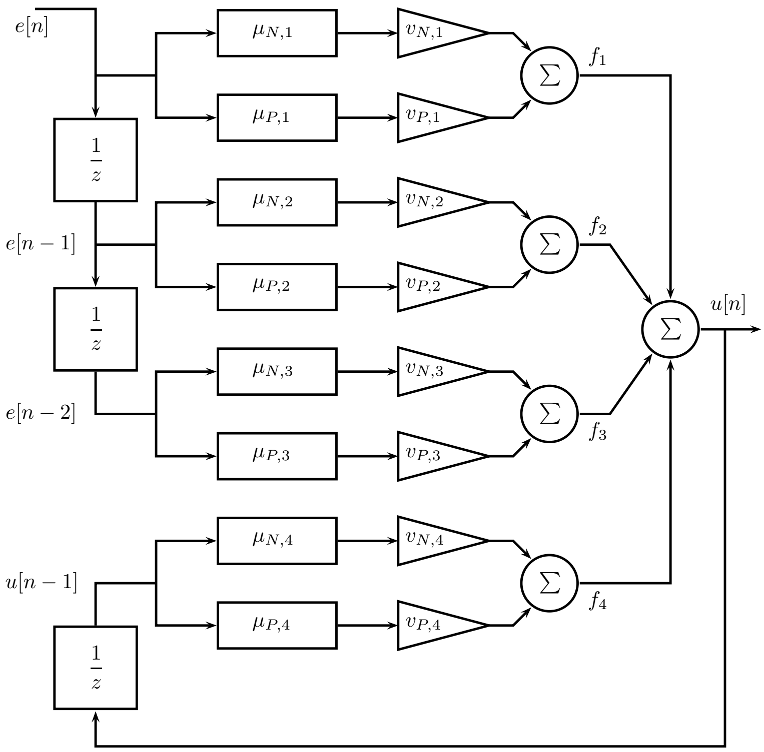

Figure 24. Compact fuzzy PID controller.

The f_i functions are given in the form:

$$f_i(x_i) = v_{N,i}\left(1 + e^{-\sigma_{N,i}(x_i - \gamma_{N,i})}\right)^{-1} + v_{P,i}\left(1 + e^{-\sigma_{P,i}(x_i - \gamma_{P,i})}\right)^{-1} \tag{70}$$

The parameters employed are the following: for the centers $\gamma_{N,i} = 0$, $\gamma_{P,i} = 0$, for the slope $\sigma_{N,i} = -0.08$, $\sigma_{P,i} = 0.08$; and the virtual actuators $v_{N,i} = -25$, $v_{P,i} = 25$. In this way, Figure 25 shows the respective functions f_i.

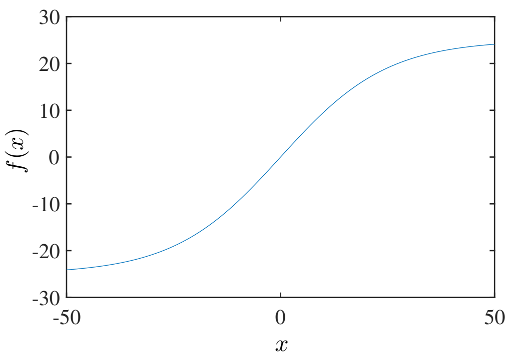

Figure 25. Input-output relationship for each f_i.

The rules used to implement the estructure of Figure 24 can be represented as shown in Boole Table 20. In general, these rules are described as:

- *If* x_i is $\mu_{N,i}$ *Then* the activation function is $Y_{N,i}$.
- *If* x_i is $\mu_{P,i}$ *Then* the activation function is $Y_{P,i}$.

Table 20. Rules associated with the estructure of Figure 24.

$\mu_{N,1}$	$\mu_{P,1}$	$\mu_{N,2}$	$\mu_{P,2}$	$\mu_{N,3}$	$\mu_{P,3}$	$\mu_{N,4}$	$\mu_{P,4}$	$Y_{N,1}$	$Y_{P,1}$	$Y_{N,2}$	$Y_{P,2}$	$Y_{N,3}$	$Y_{P,3}$	$Y_{N,4}$	$Y_{P,4}$
1	X	X	X	X	X	X	X	1	0	0	0	0	0	0	0
X	1	X	X	X	X	X	X	0	1	0	0	0	0	0	0
X	X	1	X	X	X	X	X	0	0	1	0	0	0	0	0
X	X	X	1	X	X	X	X	0	0	0	1	0	0	0	0
X	X	X	X	1	X	X	X	0	0	0	0	1	0	0	0
X	X	X	X	X	1	X	X	0	0	0	0	0	1	0	0
X	X	X	X	X	X	1	X	0	0	0	0	0	0	1	0
X	X	X	X	X	X	X	1	0	0	0	0	0	0	0	1

Then, the controller output is calculated as:

$$u = v_{N,1}Y_{N,1} + v_{P,1}Y_{P,1} + v_{N,2}Y_{N,2} + v_{P,2}Y_{P,2} + v_{N,3}Y_{N,3} + v_{P,3}Y_{P,3} + v_{N,4}Y_{N,4} + v_{P,4}Y_{P,4} \quad (71)$$

6.3. Optimization Process

The optimization process includes an operation that permits the calculation of the control system dynamics. The optimization parameters X defines the response of the control system $y_{out}(X)$. The reference output y_{ref} (desired behavior) is found from the reference model given by the transfer function display in Equation (72). Figure 26 shows the reference model response for different values of $r(t)$ where the settling time is the 1800 s namely 30 min (time scale in minutes). The reference model is proposed to reduce the settling time of the open-loop plant presented in Figure 22.

$$\frac{Y_{ref}(s)}{R_{ref}(s)} = \frac{1}{450\,\mathrm{s} + 1} \quad (72)$$

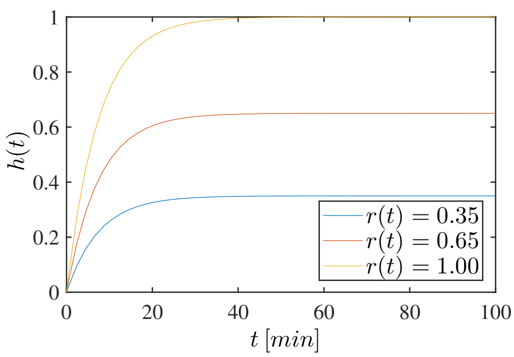

Figure 26. Reference model response.

Figure 27 shows the scheme used for the optimization process. The result of y_{out} and y_{ref} are employed to calculate the performance index to optimize $J(X)$. Thus, the optimization process employs the performance index function. In addition, X corresponds to the controller's set of parameters, n is the discrete-time variable, and N is the number of data employed. In this way, the performance function is obtained, which corresponds to mean squared error:

$$J = \frac{1}{N}\sum_{n=1}^{N}(y_{ref}(n) - y_{out}(n, X))^2 \quad (73)$$

The Quasi-Newton method of BFGS (Broyden–Fletcher–Goldfarb–Shanno) is used in the optimization process; this method performs a successive Hessian approximation [65].

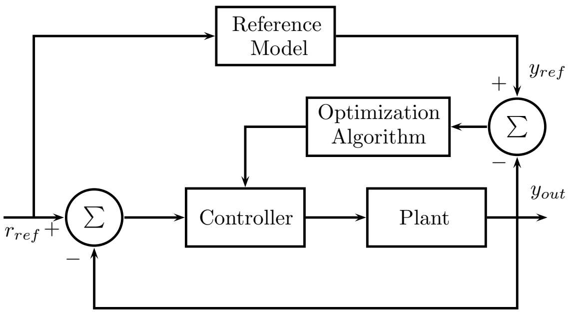

Figure 27. Scheme of the optimization process.

6.4. Results

This section displays the results after optimizing a compact fuzzy controller; the optimization variables are the parameters of the membership function $\mu_{N,i}$, $\mu_{P,i}$, and the actuators $v_{N,i}$, $v_{P,i}$. For this case, (PID controller), the action of feedback is not optimized since it is the integral component. The value before the optimization is 0.0876085 for the objective function. After the process, the value of the objective function is 4.75608×10^{-5} using 100 iterations. Figure 28 shows the optimization process results. It is also seen the response of the system with the training data; meanwhile, Figure 29 shows the controller's behavior for different references employed in the training process.

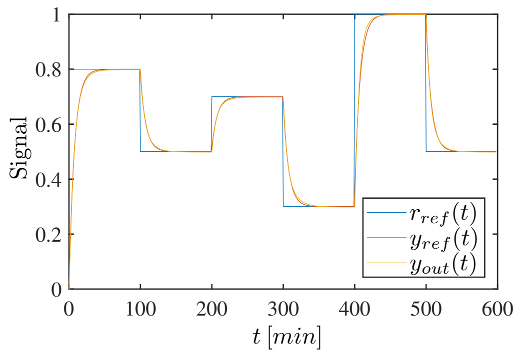

Figure 28. Optimization results with training data.

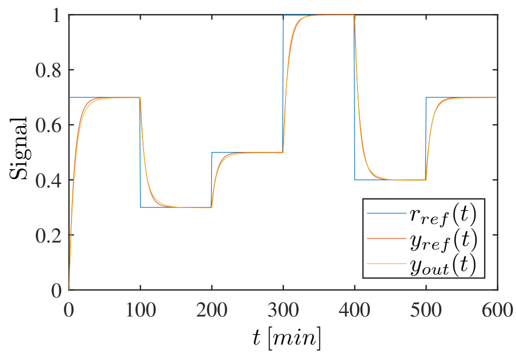

Figure 29. Controller results with different references.

7. Discussion

In this work, two schemes of compact fuzzy systems based on Boolean relationships were formalized, which can be used in identification and control; however, other architectures can be established, such as when an activation is used with partial dependence on the input sets or ad hoc structures for dynamic systems. Additionally, more elaborate applications can be carried out for the prediction of chaotic series such as energy consumption in electrical systems, as well as the development of nonlinear controllers and adaptive control techniques for thermal, hydraulic, and generators among others.

8. Conclusions

This work reviewed the methodology for designing fuzzy inference systems applied to specific cases to have compact fuzzy inference systems based on Boolean relations. The advantage of this methodology consists of presenting the rules of the system as a truth table and establishing the equations for system implementation using Boole and Kleene algebras.

Possible application cases of the concepts used for the design of compact fuzzy systems based on Boolean relations were presented. These systems can be generalized to have more complex structures without losing interpretability. Through the cases under study, the equivalence is achievable to simplify expressions using Boole and Kleene algebras, which permits various configurations of compact inference systems based on Boolean relations.

The application examples considered to identify and control nonlinear dynamic systems allow a straightforward application of fuzzy systems based on Boolean relations with interpretability and adaptability. Architecture I allowed for the identification of the system that presents multiplications between the variables that describe its behavior; meanwhile, architecture II allowed for the expansion of the adaptation capacity of a PID controller achieving the desired behavior of this control system considering the desired reference model.

For the architecture I used to plant identification, the value of the objective function before training is 0.79851; the result after training is equal to 9.3592×10^{-4} using 200 iterations. Meanwhile, for architecture II applied to control, the value before the optimization is 0.0876085 for the objective function; after the process, the value obtained is 4.75608×10^{-5} using 100 iterations. These results show that the proposed architectures allow the adaptation of their parameters using optimization algorithms.

The initial configuration to perform the optimization is essential; in this way, the proposal allows for the easy establishment of the initial configurations of the proposed systems. For architecture I, the initial configuration is obtained systematically, distributing the sets in the range of the inputs. For the control application, the sigmoidal fuzzy sets were configured considering the value of the desired output. With these initial configurations, an adequate optimization of the fuzzy systems used for identification and control is achieved.

The researchers expect to employ compact fuzzy systems to design adaptive neuro-fuzzy control systems in further work. Given the compact structure, these systems make possible the use of Back Propagation and Dynamic Back Propagation algorithms for adaptive applications. Researchers will also consider carrying out MIMO applications using compact subsystems based on Boolean relationships in such a way that different MIMO configurations can be used for identification and control.

Author Contributions: Conceptualization, H.E. and J.S.; Methodology, H.E. and J.S.; Project administration, H.L.; Supervision, H.L.; Validation, I.M.; Writing—original draft, H.E.; Writing—review and editing, H.E. and I.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: This work does not use external data.

Acknowledgments: The authors express gratitude to the Universidad de Oviedo, and the Universidad Distrital Francisco José de Caldas.

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

References

1. Voskoglou, M.G. Methods for Assessing Human-Machine Performance under Fuzzy Conditions. *Mathematics* **2019**, *7*, 230. [CrossRef]
2. Tuan, T.M.; Lan, L.T.H.; Chou, S.-Y.; Ngan, T.T.; Son, L.H.; Giang, N.L.; Ali, M. M-CFIS-R: Mamdani Complex Fuzzy Inference System with Rule Reduction Using Complex Fuzzy Measures in Granular Computing. *Mathematics* **2020**, *8*, 707. [CrossRef]
3. Chrysafis, K.A.; Papadopoulos, B.K. Decision Making for Project Appraisal in Uncertain Environments: A Fuzzy-Possibilistic Approach of the Expanded NPV Method. *Symmetry* **2021**, *13*, 27. [CrossRef]

4. Rodríguez, R.M.; Martínez, L.; Torra, V.; Xu, Z.S.; Herrera, F. Hesitant Fuzzy Sets: State of the Art and Future Directions. *Int. J. Intell. Syst.* **2014**, 29, 495–524. [CrossRef]
5. Roszkowska, E.; Jefmanski, B. Interval-Valued Intuitionistic Fuzzy Synthetic Measure (I-VIFSM) Based on Hellwig's Approach in the Analysis of Survey Data. *Mathematics* **2021**, *9*, 201. [CrossRef]
6. Peng, Y.; Tao, Y.; Wu, B.; Wang, X. Probabilistic Hesitant Intuitionistic Fuzzy Linguistic Term Sets and Their Application in Multiple Attribute Group Decision Making. *Symmetry* **2020**, *12*, 1932. [CrossRef]
7. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
8. Troussas, C.; Chrysafiadi, K.; Virvou, M. An intelligent adaptive fuzzy-based inference system for computer-assisted language learning. *Expert Syst. Appl.* **2019**, *127*, 85–96. [CrossRef]
9. Tiwari, L.; Raja, R.; Sharma, V.; Miri, R. Fuzzy Inference System for Efficient Lung Cancer Detection. In *Computer Vision and Machine Intelligence in Medical Image Analysis*; Springer: Singapore, 2020; pp. 33–41.
10. Sagir, A.M.; Sathasivam, S. A Novel Adaptive Neuro Fuzzy Inference System Based Classification Model for Heart Disease Prediction. *Pertanika J. Sci. Technol.* **2017**, *25*, 43–56.
11. Afriyie-Mensah, R.; Xiao, J.; Das, O.; Jiang, L.; Xu, Q.; Alhassan, M.O. Application of Adaptive Neuro-Fuzzy Inference System in Flammability Parameter Prediction. *Polymers* **2020**, *12*, 122. [CrossRef] [PubMed]
12. Bakhshipour, A.; Zareiforoush, H.; Bagheri, I. Application of decision trees and fuzzy inference system for quality classification and modeling of black and green tea based on visual features. *J. Food Meas. Charact.* **2020**, *14*, 1402–1416. [CrossRef]
13. Manogaran, G.; Varatharajan, R.; Priyan, M.K. Hybrid recommendation system for heart disease diagnosis based on multiple kernel learning with adaptive neuro-fuzzy inference system. *Multimed. Tools Appl.* **2018**, *77*, 4379–4399. [CrossRef]
14. Handoyo, S.; Kusdarwati, H. Implementation of Fuzzy Inference System for Classification of Dengue Fever on the villages in Malang. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2019; Volume 546, p. 052026.
15. Shastry, K.A.; Sanjay, H.A. Adaptive Neuro-Fuzzy Inference System in Agriculture. In *Fuzzy Expert Systems and Applications in Agricultural Diagnosis*; IGI Global: Hershey, PA, USA, 2020.
16. Abdolkarimi, E.S.; Mosavi, M.R. Wavelet-adaptive neural subtractive clustering fuzzy inference system to enhance low-cost and high-speed INS/GPS navigation system. *GPS Solut.* **2020**, *24*, 36. [CrossRef]
17. de Campos Souza, P.V.; Lughofer, E. Identification of Heart Sounds with an Interpretable Evolving Fuzzy Neural Network. *Sensors* **2020**, *20*, 6477. [CrossRef] [PubMed]
18. de Campos Souza, P.V. Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Appl. Soft Comput.* **2020**, *92*, 106275. [CrossRef]
19. Hartono. Optimization of Tsukamoto Fuzzy Inference System using Fuzzy Grid Partition. *IJCSN Int. J. Comput. Sci. Netw.* **2016**, *5*, 786–791.
20. Babuska, R. *Fuzzy Modeling for Control*; Kluwer Academia Publishers: Boston, MA, USA, 1998.
21. Nobre, F.S.M.; Nakasone, J.J.; Palhares, A.G.B.; Madrid, M.K.; Roy, R. Fuzzy logic in management control: A case study. In Proceedings of the 2000 IEEE In Engineering Management Society, Albuquerque, NM, USA, 15 August 2000; pp. 414–419.
22. Nguyen, H.T.; Nadipuram, R.P.; Walker, C.L.; Walker, E.A. *A First Course in Fuzzy and Neural Control*; Chapman & Hall/CRC: Boca Raton, FL, USA, 2003.
23. Al-Qudah, Y.; Hassan, M.; Hassan, N. Fuzzy Parameterized Complex Multi-Fuzzy Soft Expert Set Theory and Its Application in Decision-Making. *Symmetry* **2019**, *11*, 358. [CrossRef]
24. Ramot, D.; Milo, R.; Friedman, M.; Kandel, A. Complex fuzzy sets. *IEEE Trans. Fuzzy Syst.* **2002**, *10*, 171–186. [CrossRef]
25. Ramot, D.; Friedman, M.; Langholz, G.; Kandel, A. Complex fuzzy logic. *IEEE Trans. Fuzzy Syst.* **2003**, *11*, 450–461. [CrossRef]
26. Selvachandran, G.; Quek, S.G.; Hong-Lan, L.T.; Son, L.H.; Giang, N.L.; Ding, W.; Abdel-Basset, M.; Albuquerque, V.H.C. A New Design of Mamdani Complex Fuzzy Inference System for Multi-attribute Decision Making Problems. *IEEE Trans. Fuzzy Syst.* **2019**. [CrossRef]
27. Chen, Z.; Aghakhani, S.; Man, J.; Dick, S. ANCFIS: A neurofuzzy architecture employing complex fuzzy sets. *IEEE Trans. Fuzzy Syst.* **2010**, *19*, 305–322 [CrossRef]
28. Liu, Y.; Liu, F. An adaptive neuro-complex-fuzzy-inferential modeling mechanism for generating higher-order TSK models. *Neurocomputing* **2019**, *365*, 94–101. [CrossRef]
29. Yazdanbakhsh, O.; Dick, S. FANCFIS: Fast adaptive neuro-complex fuzzy inference system. *Int. J. Approx. Reason.* **2019**, *105*, 417–430. [CrossRef]
30. Atanassov, K.T. Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **1986**, *20*, 87–96. [CrossRef]
31. Dubois, D.; Prade, H. Fundamentals of fuzzy sets. In *The Handbooks of Fuzzy Sets*; Springer: Berlin/Heidelberg, Germany, 2000; Volume 7.
32. Bustince, H.; Fernandez, J.; Kolesarov, A.; Mesiar, R. Generation of linear orders for intervals by means of aggregation functions. *Fuzzy Sets Syst.* **2013**, *220*, 69–77. [CrossRef]
33. Yager, R.R. On the theory of bags. *Int. J. Gen. Syst.* **1986**, *13*, 23-37. [CrossRef]
34. Torra, V. Hesitant fuzzy sets. *Int. J. Intell. Syst.* **2010**, *25*, 529–539. [CrossRef]
35. Beg, I.; Rashid, T. Group Decision Making Using Intuitionistic Hesitant Fuzzy Sets. *Int. J. Fuzzy Log. Intell. Syst.* **2014**, *14*, 181–187. [CrossRef]
36. Ejegwa, P.A.; Akowe, S.O.; Otene, P.M.; Ikyule, J.M. An Overview on Intuitionistic Fuzzy Sets. *Int. J. Sci. Technol. Res.* **2014**, *3*, 142–145.

37. Qian, G.; Wang, H.; Feng, X. Generalized hesitant fuzzy sets and their application in decision support system. *Knowl.-Based Syst.* **2013**, *37*, 357–365. [CrossRef]
38. Chen, N.; Xu, Z.; Xia, M. Interval-valued hesitant preference relations and their applications to group decision making. *Knowl. Based Syst.* **2013**, *37*, 528–540. [CrossRef]
39. Rodríguez, R.M.; Martínez, L.; Herrera, F. Hesitant fuzzy linguistic term sets for decision making. *IEEE Trans. Fuzzy Syst.* **2012**, *20*, 109–119. [CrossRef]
40. Xu, Z. Hesitant Fuzzy Sets Theory. In *Studies in Fuzziness and Soft Computing*; Springer International Publishing: Cham, Switzerland, 2014; Volume 314.
41. Dougherty, E.; Giardina, C. *Mathematical Methods for Artificial Intelligence and Autonomous Systems*; Prentice Hall: Englewood, CO, USA, 1988.
42. Silveira, L. Diseño e Implementación de un Motor de Reglas Dinámicas Usando Especificaciones GeneXus. Master's Thesis, Universidad de la República, Montevideo, Uruguay, 2010.
43. Barco, C. *Álgebra Booleana. Aplicaciones Tecnológicas*; Editorial; Universidad de Caldas: Manizales, Colombia, 2005.
44. Espitia, H.; Soriano, J. Sistema de inferencia difusa basado en relaciones Booleanas. *Rev. Ing.* **2010**, *15*, 52–66.
45. Nguyen, H.T.; Walker, E.A. *A First Course in Fuzzy Logic*, 3rd ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2006.
46. Soriano, J.J.; González, O.L.; Munar, F.V.; Ramos, A.A. Propuesta de concresor basado en relaciones Booleanas. *Ingeniería* **2001**, *6*, 42–50.
47. Salazar, O.; Soriano, J.; Serrano, H. From binary logic functions to fuzzy logic functions. *Appl. Math. Sci.* **2013**, *7*, 5129–5138. [CrossRef]
48. Salazar, O.; Soriano, J. *Método de Simplificación de Fórmulas por Medio de Álgebras Finitas*; Editorial UD; Universidad Distrital Francisco José de Caldas: Bogotá, Colombia, 2018.
49. Soriano, J. *Álgebra Abstracta Aplicada en Ingeniería. Casos de Aplicación en Sistemas Difusos tipo 1 y tipo 2*; Editorial UD; Universidad Distrital Francisco José de Caldas: Bogotá, Colombia, 2019.
50. Espitia, H.; Soriano, J.; Machón, I.; López, H. Design Methodology for the Implementation of Fuzzy Inference Systems Based on Boolean Relations. *Electronics* **2019**, *8*, 1243. [CrossRef]
51. Morales, L.; Espitia, H.; Soriano, J. Predicción de la serie de tiempo Lorenz usando un sistema neuro-DBR. *Rev. Cienc. Ing. Neogranadina* **2010**, *20*, 31–51. [CrossRef]
52. Espitia, H.; Díaz, G.; Díaz, S. Adaptive model for a variable load in a distribution network using a neuro-fuzzy system. In Proceedings of the IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA), Bogotá, Colombia, 31 May–2 June 2017.
53. Espitia, H.; Díaz, G.; Díaz, S. Control of a Permanent Magnet Synchronous Generator Using a Neuro-Fuzzy System. *Appl. Comput. Sci. Eng.* **2018**, *915*, 89–101.
54. Espitia, H.; Machón, I.; López, H.; Díaz, G. Proposal of an Adaptive Neurofuzzy System to Control Flow Power in Distributed Generation Systems. *Complexity* **2019**. [CrossRef]
55. Gehrke, M.; Walker, C.; Walker, E. A mathematical setting for fuzzy logic. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **1997**, *5*, 223–238. [CrossRef]
56. Gehrke, M.; Walker, C.; Walker, E. Normal forms and truth tables for fuzzy logics. *Fuzzy Sets Syst.* **2003**, *138*, 25–51. [CrossRef]
57. Gehrke, M.; Walker, C.; Walker, E. Some comments on fuzzy normal forms. In Proceedings of the Ninth IEEE International Conference on Fuzzy Systems, San Antonio, TX, USA, 7–10 May 2000; Volume 2, pp. 593–598.
58. Mukaidono, M. Regular ternary logic function-Ternary logic functions suitable for treating ambiguity. *IEEE Trans. Comput.* **1986**, *C-35*, 183. [CrossRef]
59. Yamamoto, Y.; Mukaidono, M. P-Functions-Ternary Logic Functions Capable of Correcting Input Failures and Suitable for Treating Ambiguities. *IEEE Trans. Comput.* **1992**, *41*, 28–35. [CrossRef]
60. Del Brio, B.; Sanz, A. *Redes Neuronales y Sistemas Difusos*; Editorial; Alfaomega, Segunda Edición: Mexico DF, Mexico, 2005.
61. Recio-Colmenares, R.; Gurubel-Tun, K.J.; Zúñiga-Grajeda, V. Optimal Neural Tracking Control with Metaheuristic Parameter Identification for Uncertain Nonlinear Systems with Disturbances. *Appl. Sci.* **2020**, *10*, 7073. [CrossRef]
62. Piltan, F.; Kim, J.-M. Bearing Fault Identification Using Machine Learning and Adaptive Cascade Fault Observer. *Appl. Sci.* **2020**, *10*, 5827. [CrossRef]
63. Wang, L. *A Course in Fuzzy Systems and Control*; Prentice Hall: Englewood, CO, USA, 1997.
64. Khalil, H. *Nonlinear Systems*, 3rd ed.; Prentice Hall: Englewood, CO, USA, 2002.
65. Mora, H. *Optimización no Lineal y Dinámica*; Editorial; Universidad Nacional de Colombia: Bogotá, Colombia, 2001.