

Article

Design Methodology for the Implementation of Fuzzy Inference 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, 111321 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

Received: 29 July 2019; Accepted: 24 October 2019; Published: 30 October 2019



Abstract: This paper proposes a methodology for the design of fuzzy inference systems based on Boolean relations. The approach using Boolean sets presents limited performance due to the abrupt transitions that occur during its functioning, therefore, fuzzy sets can be used aiming the improvement of the performance. In this approach, firstly, the design of a Boolean controller is performed, which is later extended into fuzzy under design guidelines proposed in this paper. The methodology uses Kleene algebra via truth tables for the fuzzy system design, allowing the simplification of the equations that implement the fuzzy system.

Keywords: Boolean relations; control; design methodology; fuzzy logic

1. Introduction

Fuzzy Inference Systems (FIS) have wide applicability in control systems due to their flexibility for control strategy implementation when ambiguity or imprecisions occur in a plant model [1–3]. On the other hand, Boolean algebra is a useful tool to design automation rules in Boolean Automation Systems (BAS) [4–6]. The design based on Boolean algebra allows to represent the knowledge of a system by means of specific cases established with Boolean-type variables; this way, an advantage of the control systems based on Boolean logic consists of a design methodology based on truth tables (widely known). Nevertheless, the performance of these systems is limited due to some abrupt transitions associated to different control actions. These abrupt transitions may be surpassed by using fuzzy sets instead of Booleans. However, more cases to consider appear in the system operation when using fuzzy sets; additionally, to design process several properties of Boolean algebra are not met for fuzzy sets; these issues can be addressed using Kleene algebra, which is an aspect to consider in this work. In this regard, it should be kept in mind that Boolean algebra (named \mathcal{B})-based tools have been developed using sets whose values are $\{0, 1\}$, while Kleene algebra (denoted \mathcal{K}) includes a third element u , having a structure of $\{0, u, 1\}$ [7].

In this line, some previous approaches (related works) have been developed, for example, Dougherty et al. [4] present a proposal to leverage the characteristics in automatism design based on Boolean algebra. This proposal is based on solving a minimization problem in FIS using BAS; meanwhile, in [8–11] a practical approach is developed where a BAS based methodology is offered to implement the associated defuzzification process. Particularly, in [8] is designed a fuzzy controller for a DC motor based on Boolean relations where a truth table and Karnaugh maps are used to find the simplified equations that implement the controller. Moreover, in [9], an active and reactive power flow fuzzy controller is proposed for a voltage source converter. This paper explores the possibility to employ fuzzy type-2 to implement fuzzy controllers using a simplified process for defuzzification,

meanwhile, Espitia et al. [10] show the design of a buck-boost voltage converter being important to show that the simplified and non-simplified equations when using fuzzy sets do not generate the same control surface; likewise, Espitia et al. [11] present a fuzzy temperature controller based on Boolean relationships where possible conditions on membership functions are proposed, so that the simplified and non-simplified equations display the same control surface, it is noteworthy that this approach is impractical because it turns out to be very restrictive, indicating that another approach should be addressed to convert a Boolean design to a fuzzy one. Meanwhile in [12,13] is exposed that the approach associated to design of fuzzy system using BAS methodology may be generating a fuzzy system called Fuzzy Inference System Based on Boolean Relations (FIS-BBR), where the concept of virtual actuator is a key aspect for obtaining the output of the fuzzy system. On the way to formalize the concepts involved in the FIS-BBR, it is important to note that none of these previous works explore the inclusion of Kleene algebra for the design process; in this respect, Salazar et al. [14] show an advance where Kleene algebra is used to convert binary logic functions into fuzzy logic. In this way, it is possible to address the situations which have been identified in previous works that exist when converting a Boolean design to fuzzy, so that the equations are equivalent. In this regard, this paper seeks to present a methodology that allows the extension of a Boolean controller into a fuzzy one for which Kleene algebra and regularity conditions are used so that the design equations with Boole and Kleene algebras are equivalent.

After this work, where the design principles are presented, the development of more complex applications would be expected, however, in a heuristic way, making an analogy with the operation of a discrete time dynamic system, a structure of an FIS-BBR could be obtained such as those shown in [15,16]; particularly, in [15] there is a neuro-fuzzy control system where plant identification and system control is performed using an equivalent FIS-BBR scheme, in [16], further progress is made by identifying the plant and optimizing the controller in an iterative way to have an adaptive control system. In these works, the similar FIS-BBR scheme is determined in a heuristic way; however, it could be formulated through the methodology proposed in this document, this aspect is expected to be addressed in future works. The above shows that the FIS-BBR can be used in more complex applications.

The fuzzy inference systems based on Boolean relations suggest an implementation method for fuzzy logic systems regarding automatism designs as those are widely and easily implemented in automatic process control [4,6,17]. This design technique considers both sensors and actuators and the Boolean relations involved in control strategies used in an industrial process [9,12,13]. Consequently, the FIS-BBR is proposed as an alternative for the implementation of fuzzy systems that present a direct interpretability for control systems with a compact inference mechanism.

Besides, Espitia et al. [10,11,13] present a basic proposal for an FIS-BBR, which considers the conversion from Boolean to fuzzy membership. Nevertheless, some issues need review and deeper analysis, which is the goal of this work, in particular, calculations with a three element algebra for propositional fuzzy logic [7]. The Kleene algebra permits to consider different cases that arise when a design is extended using Boolean sets to fuzzy sets; additionally, to achieve this extension, criteria are also required to obtain monotonic transitions between the actions defined in the Boolean design. Thus, an equivalence of Boolean and fuzzy design can be achieved, which consists of obtaining similar equations using Boolean and Kleene algebra, and under the criteria considered achieve the monotonic transitions between Boolean actions. Then, the Kleene algebra is used in the methodology design for an FIS-BBR, in this way, a table of two elements can be extended to one of three, taking into account some regularity conditions and also several design principles proposed in this paper. Using a three element table and Kleene algebra it is possible to establish relations in the inference process together with a simplification process. Another important aspect in this paper (to implement FIS-BBR systems) consists of the criteria to convert a Boolean membership into a fuzzy membership.

Considering that the main objective of this paper is to show the proposed design process, then a basic application case is used as a conceptual example showing that when using the proposed

procedure, it is possible to have an equivalence between the equations obtained with Boole and Kleene algebras. Other complex cases are considered for future developments for neuro-fuzzy systems and adaptive control.

Seeking to have an adequate order in the document, the theoretical concepts necessary to apply the design methodology are presented in the first part. Later, the operation process in an FIS-BBR is presented. Based on the above, the proposed methodology design is presented to finally develop a demonstrative example to observe the methodology in detail. The sections are organized as follows: it starts by giving a brief introduction to the Boolean design and previous works about fuzzy inference systems based on Boolean relations. Section 2 contains the framework, where Subsection 2.1 reviews Boolean and Kleene algebras, these concepts are used in the design process mainly for algebraic operations. Fuzzy sets are shown in Subsection 2.2, then Subsection 2.3 presents the well-known Mamdani and Takagi–Sugeno models, which are taken as a reference to show the inference process in an FIS-BBR system. Section 3 describes the inference system based on Boolean relations. Virtual actuator concept is here emphasized, which is the main aspect of FIS-BBR as this allows inclusions of feasible control actions; thus, it is directly applied on control systems. Subsection 3.2 shows the inference process for FIS-BBR considering the different steps. Taking in account the properties of Boole and Kleene algebras, and the conditions of regularity, the design methodology is described in Section 4. To show the methodology, in Section 5 the design for a tank fill system using Boolean and FIS-BBR controllers is presented, comparing the performance in each case. Finally, in Section 6, the discussion about characteristics of BBR systems is made. Conclusions are given in Section 7.

2. Framework

In the first part, the main aspects of the Boole and Kleene algebras are reviewed, which allow the manipulation of mathematical expressions; this aspect is relevant to achieve the transition between a Boolean design to a fuzzy one. The membership function definitions that are important in the proposed methodology to convert a Boolean set into a fuzzy one are also reviewed. Subsequently, the operation schemes for Mamdani and Takagi–Sugeno fuzzy systems are presented, so that they can be taken as a reference to observe the inference process in an FIS-BBR.

2.1. Boolean and Kleene Algebras

The design methodology for FIS-BBR considers the requirements to use Boolean and Kleene algebra; the design process starts in a Boolean domain and is then extended to Kleene; therefore, the concepts of Boolean and Kleene algebra are reviewed below.

Although algebra and logic can be considered as independent mathematical structures according to references [7,18,19], for Boolean and Kleene algebra, an equivalence with logic structures can be stated.

Definition 1 ([19]). *A De Morgan algebra is a bounded distributive lattice $\mathcal{M} = (A, \vee, \wedge, ', 0, 1)$, consisting of a set A together with two internal binary operations \wedge (called “meet” or “intersection” or “conjunction”), \vee (called “join” or “union” or “disjunction”), a unary operation $'$ (called “complement” or “not”) and two constants 0 and 1 or nullary operations. The operations \wedge and \vee are associative, commutative and distributive (\vee with regard to \wedge and vice-versa). Considering $\forall x, y \in A$ the De Morgan algebra satisfies:*

$$(x')' = x \tag{1}$$

$$(x \vee y)' = x' \wedge y' \tag{2}$$

$$(x \wedge y)' = x' \vee y' \tag{3}$$

Definition 2 ([19]). *A Kleene algebra $\mathcal{K} = (K, \vee, \wedge, ', 0, 1)$ is a De Morgan algebra that meets (4).*

$$x \wedge x' \leq y \vee y' \tag{4}$$

Definition 3 ([19]). A Boolean algebra $\mathcal{B} = (B, \vee, \wedge, ', 0, 1)$ is a Kleene algebra that meets (5) and (6).

$$x \vee x' = 1 \tag{5}$$

$$x \wedge x' = 0 \tag{6}$$

In the proposed design methodology for FIS-BBR is used the Boolean algebra of two elements, where $B = \{0, 1\}$, and also the three element Kleene algebra with $K = \{0, u, 1\}$, where the additional element u belongs to the interval $(0, 1)$. Boolean algebra has only two elements and it is used in logic applications where 0 means *false*, 1 *true*, \wedge *and*, \vee *or*, and $'$ *not*.

In comparison with the Boolean algebra in Kleene algebra is observable that:

$$x \vee x' \leq 1 \tag{7}$$

$$x \wedge x' \geq 0 \tag{8}$$

For the FIS-BBR design process proposed, the distributive and absorption properties are especially used, Equations (9) and (10), respectively.

$$(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z) \tag{9}$$

$$x \wedge (x \vee y) = x \tag{10}$$

The main aspect to using Kleene algebra is that according to references [7,19], “different truth value algebras can give the same propositional logic”. In particular, using the standard operators of fuzzy propositional calculus “the propositional calculus for three-valued logic and the calculus for fuzzy logic are the same” (the formulas are equivalent) [19]. It is noticeable that Kleene trivalent logic has a finite number of truth values, while fuzzy logic has an infinite number of truth values (multivalent). It is also worth highlighting that the expression can be handled using normal forms and truth tables. Then a fuzzy propositional logic system can be represented using truth tables [7].

Between \mathcal{K} and \mathcal{B} algebras there exists a relation which can be observed when rows having the variable u are suppressed in Table 1. In this way, Table 1 becomes one of \mathcal{B} [7].

Table 1. Table of truth for the disjunction, conjunction and complement in \mathcal{K} algebra.

Disjunction		Conjunction				Complement	
\vee	\wedge	$'$					
0	0	0	0	0	0	1	
u	u	u	u	u	u	u	
1	1	1	1	1	1	0	

Arithmetic operations for disjunction, conjunction and complement could be used for propositional calculus. Maximum, minimum and complement as $x' = 1 - x$ operations are called Kleene and Boolean standard algebra operations and are used for propositional calculus [7].

2.2. Fuzzy Sets and Operators

In this section the basic concepts about fuzzy sets and fuzzy operators used in FIS-BBR are reviewed. The first definition is that of a fuzzy set [20,21].

Definition 4 ([20]). Let X be a set, a fuzzy set A in X is given by the membership function:

$$A : X \rightarrow [0, 1]$$

A fuzzy set can be denoted by $A = \{(x, A(x)) | x \in X\}$.

Another common notation used for membership function of the fuzzy set A of X is $\mu_A(x) : X \rightarrow [0, 1]$, then, $A(x)$ and $\mu_A(x)$ are equivalents. In this article both definitions are used, mainly $A(x)$ when algebraic manipulations are performed, and $\mu_A(x)$ when it is considered the shape of the membership function.

In the case of a Boolean set, the value of the membership $\mu_A(x)$, is 1 when the element fully belongs to set A and 0 when it does not [22]. According to [23–25], Zadeh suggested that element membership may not necessarily have a 0 value or 1 uniquely but could have a value of the interval $[0, 1]$ to consequently obtain a fuzzy set defined as a set of tuples $\{\mu_A(x), x\}$, where $\mu_A(x)$ represents the membership of set A of x .

For practical implementations of FIS-BBR, it is possible the use of general connectors via triangular norms, conorms (t-norms and t-conorms respectively), and complements. These definitions are presented below:

Definition 5 ([26]). A triangular norm t-norm is a binary operation $\odot : [0, 1] \times [0, 1] \rightarrow [0, 1]$ used in multi-valued logic (fuzzy logic). The t-norm is a general form of conjunction operation in logic, it is commutative, associative, and with 1 as neutral element. The t-norm for any $a, b, c \in [0, 1]$ satisfies:

- Commutative: $a \odot b = b \odot a$.
- Associative: $a \odot (b \odot c) = (a \odot b) \odot c$.
- Identity: $1 \odot a = a$.
- Increasing in each variable: $a \odot b \leq a \odot c$ if $b \leq c$.

Some t-norms are minimum, product, drastic, Lukasiewicz, Frank, Yager, Schweizer-Sklar, Hamacher and Sugeno-Weber [27].

Definition 6 ([26]). A dual function called t-conorm $\oplus : [0, 1] \times [0, 1] \rightarrow [0, 1]$, is commutative, associative and neutral with 0 element. The t-conorm for $a, b, c \in [0, 1]$ satisfies:

- Commutative: $a \oplus b = b \oplus a$.
- Associative: $a \oplus (b \oplus c) = (a \oplus b) \oplus c$.
- Identity: $0 \oplus a = a$.
- Increasing in each variable: $a \oplus b \leq a \oplus c$ if $b \leq c$.

Taking in account [27] some t-conorms are maximum, probabilistic (algebraic sum), Lukasiewicz, Drastic, Frank, Yager, Schweizer-Sklar, Hamacher and Sugeno-Web.

Definition 7 ([28]). A fuzzy complement is defined by a function $\gamma : [0, 1] \rightarrow [0, 1]$. This function for $a, b \in [0, 1]$ satisfies:

- Boundary conditions: $\gamma(1) = 0$ and $\gamma(0) = 1$.
- Monotonicity: if $a \leq b$, then $\gamma(a) \geq \gamma(b)$.
- Continuity: $\gamma(a)$ is a continuous function.
- Involution: $\gamma(\gamma(a)) = a$.

Some complements are standard, Sugeno and Yager.

An important aspect in FIS-BBR consists of the relations between Boolean and Fuzzy sets, in this way the α -cut concept is used. The α -cut can be expressed as follows.

Definition 8 ([29]). Considering a value $\alpha \in [0, 1]$, the α -cut in a fuzzy set A has the elements of X with a membership value μ_A equal to or greater than α when having:

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\} \quad (11)$$

2.3. Fuzzy Systems

Taking in account [30,31] fuzzy logic allows the modeling of a system using fuzzy sets and rules that describe the system behavior. Fuzzy systems permit to model and control non-linear processes; after designing a fuzzy system (based on knowledge), optimization algorithms can be used to achieve better performance [32,33]. Moreover, relations among variables in fuzzy systems are presented using rules as:

If antecedent is linguistic label then consequent is linguistic label.

The linguistic labels are represented using fuzzy sets and allow described characteristics of the antecedent and consequent [34,35]. For the design of fuzzy control systems, there are two major methods: Mamdani and Takagi–Sugeno models, which are differentiated by their capability to present the collected information [36].

2.3.1. Mamdani Model

In Mamdani fuzzy systems both antecedent and consequent are linguistic terms and the output corresponds to the superposition of individual outputs given by each rule [37]. Figure 1 shows the basic structure of a Mamdani-type system.

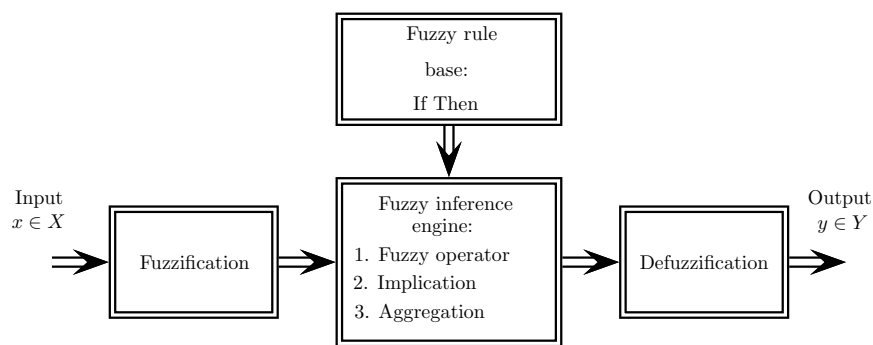


Figure 1. Mamdani systems.

In a Mamdani fuzzy system, the fuzzification, the inference fuzzy engine, and the defuzzification process are distinguished. In the fuzzification process a specific value enters the system and is converted in a membership level to each rule.

The base of the rules consists of the methodology in which the fuzzy system stores linguistic knowledge (*If-Then* type rules). A rule as rules-base or knowledge base has two parts: an antecedent and a consequent. In a Mamdani-type system both rules are given by linguistic expressions which are represented by membership functions.

The output obtained by the inference mechanism corresponds to a fuzzy set, then in order to be interpreted as numerical data, it is necessary to convert the fuzzy output inference mechanism into a specific value; this process is carried out by the defuzzification process through multiple options, such as centroid and centers average, among others [38,39].

Figure 2 shows a Mamdani-type fuzzy system operation, the following steps must be taken into account. In the first instance in the fuzzification process a concrete value is converted into a membership one for each rule, then the fuzzy operators are applied for each rule, next the implication method is performed; the results of such implication are aggregated to finally obtain the output by applying the defuzzification method.

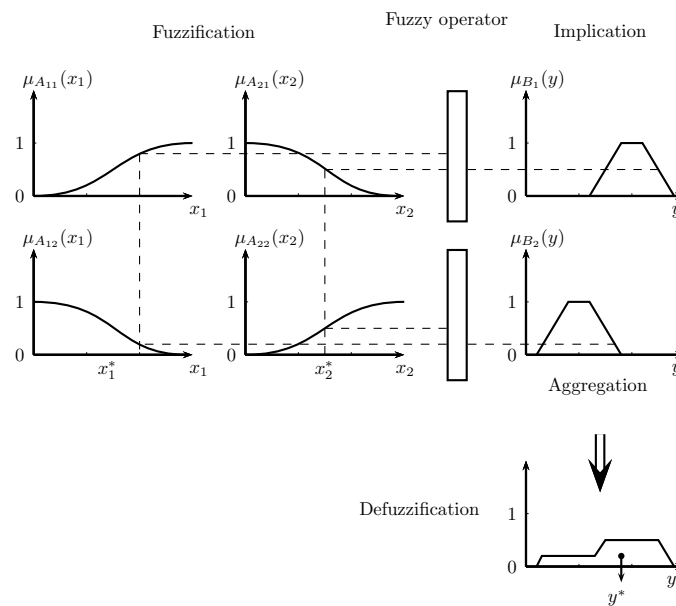


Figure 2. Operation of a fuzzy Mamdani-type inference system.

2.3.2. Takagi–Sugeno Model

This model consists of logic rules (regression rules) containing fuzzy antecedents associated to the input and functional consequents in the output. Takagi–Sugeno (TSK) model permits both represent qualitative knowledge and the use of quantitative data for the training process [40,41]. In a TSK model a fuzzy rule is shown as:

$$\text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \text{ and } \dots \text{ and } x_n \text{ is } A_n \text{ then } y_1 = f_1(x_1, x_2, \dots, x_n)$$

In the antecedents A_1, A_2, \dots, A_n are fuzzy sets, and $y_1 = f_1(x_1, x_2, \dots, x_n)$ corresponds to a function in the consequent. In many applications f_1 may be a polynomial, in this way some alternatives to f_1 are:

- TSK zero order, when f is a constant value, then $f_1 = c$.
- TSK first order, when f is a polynomial as $f_1 = a_1x_1 + a_2x_2 + \dots + a_nx_n + c$.

In Takagi–Sugeno systems, values given by activated rules are already numeric data, in this way, it is considered that the processes of aggregation and defuzzification occur in one operation [30]. Figure 3 shows the elements that define a TSK fuzzy system.

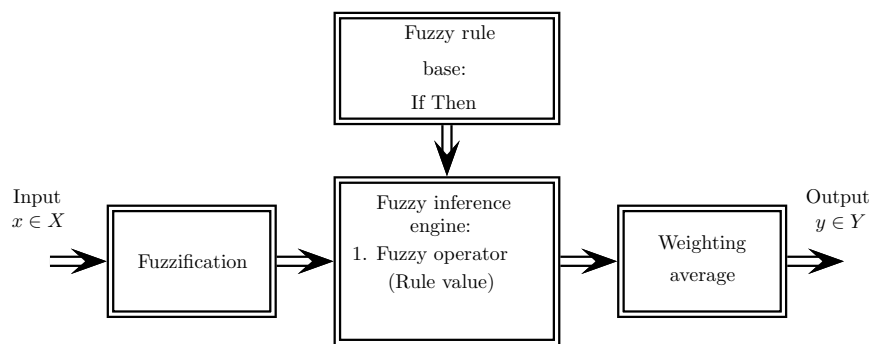


Figure 3. TSK fuzzy system.

Depending on the input variables (n), each rule (m) has a function associated:

$$\begin{aligned} y_1 &= f_1(x_1, x_2, \dots, x_n) \\ y_2 &= f_2(x_1, x_2, \dots, x_n) \\ &\vdots \\ y_m &= f_m(x_1, x_2, \dots, x_n) \end{aligned}$$

For calculating the output in a TSK system the different functions are weighted considering the value by which the antecedent associated to each rule was activated; the output of the system may be calculated as:

$$y = \frac{w_1 y_1 + w_2 y_2 + \dots + w_m y_m}{w_1 + w_2 + \dots + w_m} \tag{12}$$

A TSK order zero system may be considered a particular case of the Mamdani type where each consequent rule is modeled using a fuzzy set singleton-type [41].

Figure 4 shows the functioning of a Takagi–Sugeno fuzzy system; the following steps are observed: In the first place, an input value is converted into a membership level for each rule using the fuzzification process, then fuzzy operators are applied obtaining truth values for each rule (rule value), finally the output is calculated via weighting average.

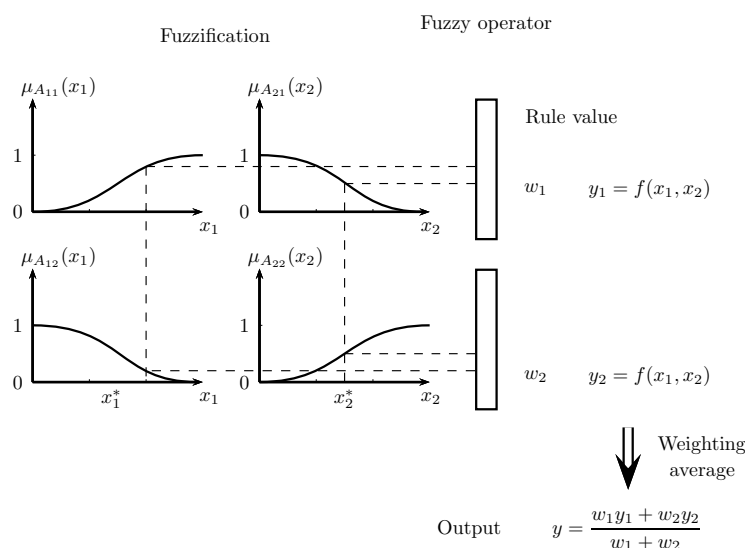


Figure 4. Operation of a fuzzy Sugeno-type inference system.

3. Fuzzy Inference Systems Based on Boolean Relations

This is a mechanism designed to implement fuzzy inference systems focused on control systems. The offered methodology with BBR is based on the possible transition from Boolean to fuzzy logic by using mechanisms of Boolean and Kleene logic design.

The concept of FIS-BBR allows performing a methodology to fuzzy inference systems design based on truth tables. The basis of BBR corresponds to an extrapolation from Boolean sets to fuzzy sets [13].

The control process is achieved via virtual outputs using activation functions established in a truth table. Disjunctive Normal Form (DNF) is used to obtain the inference equations for activation functions, Figure 5 shows how the process is performed, the simplification of inference rules using Kleene algebra is also considered for the design process.

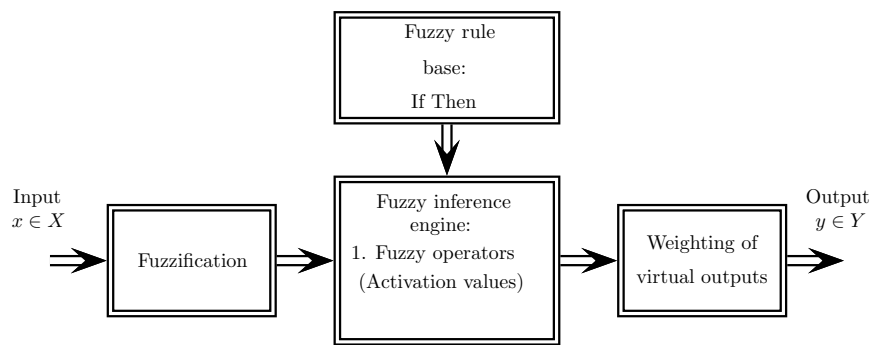


Figure 5. Fuzzy systems Based on Boolean Relations.

FIS-BBR becomes an alternative to implement systems of control based on automation, thus, it is rooted in automation design principles (such as that one applied to model digital circuits) but extended to fuzzy systems. The extension from Boolean to fuzzy systems can be developed by trivalent systems (Kleene algebra), which can be directly extended to fuzzy systems.

From the automatism viewpoint there is an action of one or several actuators starting from values determined by sensors; this action is usually ON/OFF (as in classic sets or crisp), generating in the process abrupt transitions. FIS-BBR aims to avoid these abrupt transitions when converting Boolean sets to fuzzy, and at the same time exploits the methodology in automatism development from the Kleene domain. The notion of virtual output is introduced in the following subsection.

3.1. Virtual Output

In control systems the actuator is the element used for applying the control action [2]; for automatism design it is possible to employ several actuators for the control action (several batteries, valves, switches...). However, when there is only one element of action, this can be considered as the sum of several actuators called “virtual”. Those at the same time have an associated virtual output. From this viewpoint the actuator may be considered as the discourse universe in the consequent. According to the application, the value of virtual actuators may be positive or negative, and also the total output may be higher than partial virtual actions [13].

The output corresponds to the sum of the respective virtual outputs:

$$y = y_{v_1} + y_{v_2} + \dots + y_{v_m} + \dots + y_{v_w}$$

$$y = \sum_{m=1}^w y_{v_m} \quad (13)$$

where $y_{v_m} = Y_m v_m$ being $Y_m \in [0, 1]$ the activation function, and $v_m \in \mathbb{R}$ is the respective value of virtual actuator. The concept of virtual output was conceived under a real physical correspondence in control applications to the total output in the inference system [13].

Example

Looking at the action of a valve of $6 \text{ cm}^3/\text{s}$ filling a tank, this valve may be considered as the contribution present in three different valves, which are given by virtual actuators:

- Small flow: $v_1 = 1 \text{ cm}^3/\text{s}$.
- Medium flow: $v_2 = 2 \text{ cm}^3/\text{s}$.
- Large flow: $v_3 = 3 \text{ cm}^3/\text{s}$.

Figure 6 shows the filling scheme of a tank with three virtual valves.

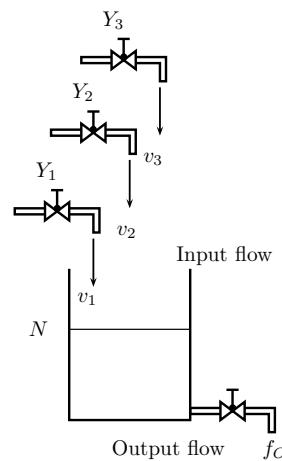


Figure 6. Tank system with three virtual valves.

Using these virtual valves make possible the actions displayed in Table 2 where ON is the valve activation. In this table “Conf.” means configuration.

Table 2. Different actions for virtual actuators.

Actuator	Conf. 1	Conf. 2	Conf. 3	Conf. 4	Conf. 5	Conf. 6	Conf. 7	Conf. 8
v_1	OFF	ON	OFF	OFF	ON	ON	OFF	ON
v_2	OFF	OFF	ON	OFF	ON	OFF	ON	ON
v_3	OFF	OFF	OFF	ON	OFF	ON	ON	ON
Flow	0 cm ³ /s	1 cm ³ /s	2 cm ³ /s	3 cm ³ /s	3 cm ³ /s	4 cm ³ /s	5 cm ³ /s	6 cm ³ /s

In this way it is conceivable to define the “Boolean actions” as the possible actions produced using Boolean sets.

3.2. Fuzzy Inference Process for FIS-BBR

Considering that fuzzy inference systems based on Boolean relations use design tools and codification of Boolean systems, the structure of the inference process is given by the truth table associated with the Boolean design. For the implementation of the FIS-BBR the Boolean codification can be extended to Kleene algebra.

Table 3 shows a possible codification for inputs and control actions of a system. According to [19], considering the normal forms for Boolean and Kleene algebras in general terms the conjunction (\wedge) is applied to the corresponding rows of inputs in Table 3, which represent the rules that provide an activation output. The disjunction (\vee) is applied over the columns of the activation outputs in Table 3.

In Table 3 variables $a_{k,j}$ and $f_{k,m}$ are used to represent relations between A_j (set associated to input) and Y_m (activation function associated to output) while for a Boolean case it has values $\{0, 1\}$ and in a Kleene case values $\{0, u, 1\}$. Table 3 can be a truth table Boolean or Kleenean, some examples are in [19]. It is noteworthy that A_j can be a Boolean or fuzzy set associated to input and Y_m is an activation function associated to output.

The implementation of FIS-BBR using Kleene algebra is made according to [19], where it is established that *fuzzy propositional logic is the same as the three valued Kleene propositional logic*.

Table 3. Truth table with activation rules.

Input Sets						Activation Outputs					
A_1	A_2	...	A_j	...	A_p	Y_1	Y_2	...	Y_m	...	Y_w
$a_{1,1}$	$a_{1,2}$...	$a_{1,j}$...	$a_{1,p}$	$f_{1,1}$	$f_{1,2}$...	$f_{1,m}$...	$f_{1,w}$
$a_{2,1}$	$a_{2,2}$...	$a_{2,j}$...	$a_{2,p}$	$f_{2,1}$	$f_{2,2}$...	$f_{2,m}$...	$f_{2,w}$
\vdots	\vdots	...	\vdots	...	\vdots	\vdots	\vdots	...	\vdots	...	\vdots
$a_{k,1}$	$a_{k,2}$...	$a_{k,j}$...	$a_{k,p}$	$f_{k,1}$	$f_{k,2}$...	$f_{k,m}$...	$f_{k,w}$
\vdots	\vdots	...	\vdots	...	\vdots	\vdots	\vdots	...	\vdots	...	\vdots
$a_{q,1}$	$a_{q,2}$...	$a_{q,j}$...	$a_{q,p}$	$f_{q,1}$	$f_{q,2}$...	$f_{q,m}$...	$f_{q,w}$

The combination of concepts using a Disjunctive Normal Form (DNF) in general can be expressed as:

$$\bigwedge_{j=1}^p \widehat{A}(a_{k,j}, f_{k,m}) \tag{14}$$

where \widehat{A} is a function that can be 0, 1, A_j , A'_j or $A_j \wedge A'_j$ and depends on $a_{k,j}$ and $f_{k,m}$ respectively (for more information about function \widehat{A} in Boolean and Kleene case see [2,7,19]). Table 4 shows the values of \widehat{A} for a Kleene or Boolean case.

Table 4. Values of \widehat{A} function.

A_j	Y_m	\widehat{A}
0	0	0
u	0	0
1	0	0
0	u	A'_j
u	u	$A_j \wedge A'_j$
1	u	A_j
0	1	A'_j
u	1	1
1	1	A_j

The m -th output activation function Y_m , using DNF can be expressed as:

$$Y_m = \bigvee_{k=1}^q \bigwedge_{j=1}^p \widehat{A}(a_{k,j}, f_{k,m}) \tag{15}$$

In [19,42] some aspects about *fuzzy logic with t-norms and t-conorms* are presented, where a logic based on t-norms and t-conorms is considered. Regarding the accomplishment of the algebraical properties used in the design (for example the distributive property [43]), it is possible to use t-norms and t-conorms obtaining:

$$Y_m = \bigoplus_{k=1}^q \bigodot_{j=1}^p \mu_{\widehat{A}} \tag{16}$$

where \oplus indicates the fuzzy operation to t-conorm (maximum, algebraic sum, etc); and \odot represents the fuzzy operation of t-norm the fuzzy operation (minimum, product, etc). Additionally, $\mu_{\hat{A}}$ is the membership function associated with \hat{A} .

Considering each combination in the table, the system global output can be calculated as:

$$y = \sum_{m=1}^w Y_m v_m \tag{17}$$

where v_m corresponds to the m -th virtual actuator. Then, the m -th virtual output is:

$$y_{v_m} = Y_m v_m \tag{18}$$

In addition, 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.

Equation 16 may be associated to Table 3 as this scheme identifies two parts: one corresponding to a codification for membership functions to each input of discourse universe; the other corresponds to the activation outputs associated to virtual outputs. Regarding the inputs, each line represents an inference rule. In output sections each column represents one activation function.

Considering Table 3 as a first step for output calculation (without simplifying), conjunction is applied (t-norm) among the sets alongside the lines whose activation output is active in the truth table (inference rules). The second step consists of applying disjunction (t-conorm) among all the results obtained in the first step to each column, which corresponds to the activation outputs.

Finally, given the fuzzy activation functions, a weighting is made to each one using the value obtained by the virtual actuator, which is called *virtual output*. A unique specific output action is obtained by summing these products.

Figure 7 shows a graphic example for the fuzzy inference process based on Boolean relations. The associated Kleene Table 5 can be an extension from Boolean Table 6, the considerations to take a relation between Boolean and Kleene Table are shown in Section 4. Considering the Kleene Table 5, the respective inference equations are:

$$Y_1 = (A_1 \wedge A_2) \vee (A_1 \wedge A'_1 \wedge A_2 \wedge A'_2) \vee (A_1 \wedge A'_1 \wedge A_2) \vee (A_1 \wedge A_2 \wedge A'_2) \tag{19}$$

$$Y_2 = (A_1 \wedge A'_1 \wedge A_2 \wedge A'_2) \vee (A'_1 \wedge A_2 \wedge A'_2) \vee (A'_1 \wedge A_2) \vee (A_1 \wedge A'_1 \wedge A_2) \vee (A_1 \wedge A'_2) \vee (A_1 \wedge A_2 \wedge A'_2) \vee (A_1 \wedge A'_1 \wedge A'_2) \tag{20}$$

Using distributive and absorption properties,

$$Y_1 = (A_1 \wedge A_2) \wedge [1 \vee (A'_1 \wedge A'_2) \vee A'_1 \vee A'_2] \tag{21}$$

$$Y_2 = (A'_1 \wedge A_2) \wedge [(A_1 \wedge A'_2) \vee A'_2 \vee 1 \vee A_1] \vee (A_1 \wedge A'_2) \wedge [1 \vee A_2 \vee A'_1] \tag{22}$$

Then the inference equations for Y_1 and Y_2 are:

$$Y_1 = A_1 \wedge A_2 \tag{23}$$

$$Y_2 = (A'_1 \wedge A_2) \vee (A_1 \wedge A'_2) \tag{24}$$

Table 5. Example of Kleene Table for FIS-BBR.

A_1	A_2	Y_1	Y_2
1	1	1	0
u	u	u	u
0	0	0	0
0	u	0	u
0	1	0	1
u	1	u	u
1	0	0	1
1	u	u	u
u	0	0	u

Table 6. Example of Boolean Table for FIS-BBR.

A_1	A_2	Y_1	Y_2
1	1	1	0
0	0	0	0
0	1	0	1
1	0	0	1

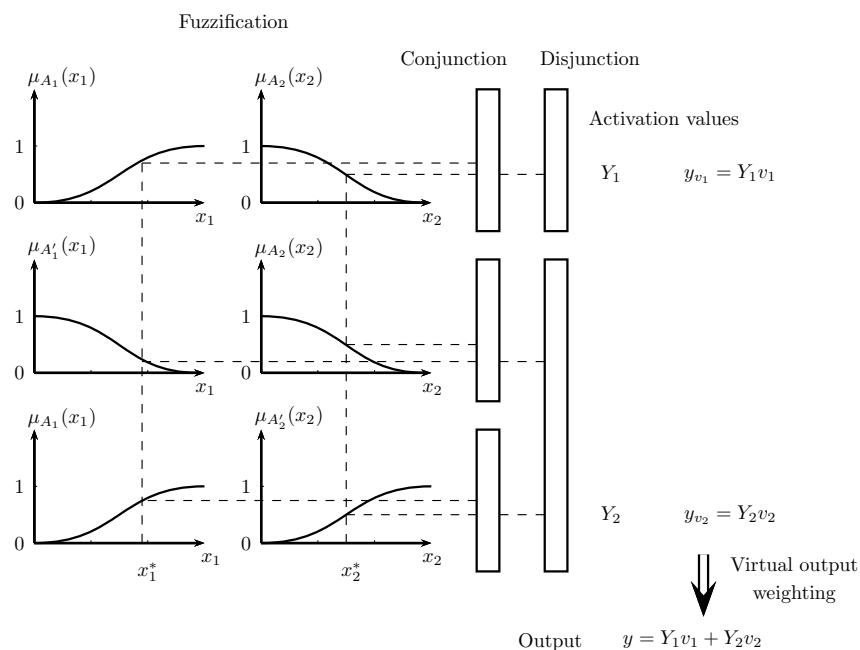


Figure 7. Operation of a fuzzy inference system Based on Boolean Relations.

4. Design Methodology

The methodology is developed considering that FIS-BBR consists of a definition of Boolean sets in the antecedent which have a membership function associated that represents concepts that can be described in a Boolean-type truth Table. It is noteworthy that the design methodology considers two main aspects:

- Establish Boolean actions.
- Monotonic transitions between Boolean actions.

The Boolean actions are defined as the action obtained by the activation function and the virtual actuator in the Boolean design (codification). Using Boolean membership the m -th Boolean action corresponds to virtual output $y_m = Y_m v_m$.

The monotonic transitions between Boolean actions correspond to a condition for a continuous transition, from y_m to y_n Boolean actions. The continuous transition is obtained using fuzzy sets instead of Boolean.

For the design methodology, the respective Boolean sets and actions are defined, then Boolean sets are converted into fuzzy using a set of criteria (based on the fuzzy sets and the extended Kleene Table). Then the formula for each activation function is determined, if necessary, a simplification process is also applied. Finally, the inference process using fuzzy sets is performed. Figure 8 shows the design methodology.

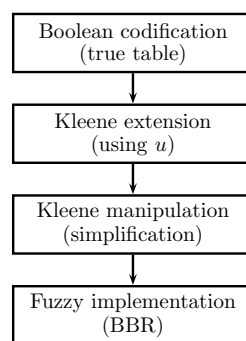


Figure 8. Design methodology.

The variable u is used for extending the Boolean Table to Kleene truth Table considering the transitions between Boolean codifications. The Kleene codification is given by the designer using regularity condition (restriction) to obtain a regular Table.

A particular case of implementation is represented when the reduced formula obtained in the ternary extended table is the same as in Boolean Table. In this case, the formula can be directly obtained from the binary table when all conditions are accomplished achieving equivalence; then, the Kleene formula is equivalent to Boolean.

4.1. Regularity Conditions

According to [44], the regularity condition for regular functions allows to handle the ambiguity the regular functions treat ambiguity from a standpoint of ternary logic. The regularity condition indicates that if the ambiguity in the input increases, then the ambiguity in the output also increases.

Definition 9 ([44]). Considering $E = \{0, u, 1\}$ a n -variable ternary logic function f is a mapping from E^n to E which to $X \in E^n$ assigns $f(X) \in E$. Based on Figure 9 an ambiguity relation \preceq corresponds to:

$$0 \preceq u, \quad 1 \preceq u, \quad i \preceq i, \tag{25}$$

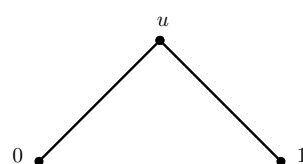


Figure 9. Partial order \preceq of ambiguity.

That means 0 and 1 are less ambiguous than u , generally if $i \preceq j$ then i is less ambiguous than j . Moreover, for distinct elements $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_n)$ of E^n , then X is less ambiguous than Y if $x_i \preceq y_i$ for $i = 1, 2, \dots, n$, namely $X \preceq Y$.

Considering that 0 and 1 are definite states and u for a state not definable as 0 or 1. Then, the partial ordering \preceq describes the ambiguity [44].

Definition 10 ([44]). A function f is regular if it meets the condition of “ambiguity monotonicity” given by:

$$f(A) \preceq f(B) \text{ whenever } A \preceq B \quad (A, B \in E^n). \tag{26}$$

This condition implies that if A is less or equally ambiguous than B then $f(A)$ is less or equally ambiguous than $f(B)$, for example considering $(x_1 = 1, x_2 = 1)$ and $(x_1 = u, x_2 = 1)$ then $(x_1 = 1, x_2 = 1) \preceq (x_1 = u, x_2 = 1)$ therefore $f(x_1 = 1, x_2 = 1) \preceq f(x_1 = u, x_2 = 1)$, then if $f(x_1 = 1, x_2 = 1) = 1$ this implies that $f(x_1 = u, x_2 = 1)$ can be 1 or u to be a regular function.

4.2. Extension from Boolean to Kleene Table

For the extension from Boolean to Kleene Table, firstly the condition (26) is necessary to handle the ambiguity; secondly, the suitable configuration to achieve monotonic transitions between Boolean actions must be taken into account.

The Kleene Table is obtained adding intermediate cases in the Boolean Table using the variable u in antecedent, and $\{0, u, 1\}$ in the consequent according to the above considerations.

An example of extension from bivalent to trivalent table achieving conditions of regularity can be seen in Figure 10; for condition (26) several options are possible; however, for achieving monotonic transitions between Boolean partitions the configuration is shown in Figure 11.

			x_1			
			x_2	0	u	1
x_1	x_2	0	1	0	u	1
0	0	0	1	u	u	$\{1, u\}$
1	1	1	1	$\{1, u\}$	1	1

Figure 10. Extension from bivalent to trivalent tables achieving regularity conditions.

			x_1		
			x_2	0	u
0	0	u	1	u	1
u	u	u	1	1	1
1	1	1	1	1	1

Figure 11. Extension for monotonic transitions between Boolean actions.

4.3. Considerations for Fuzzy Sets Selection

As the suggested system is based on automatism designs, Boolean sets are determined in consideration of conceivable sensors which divide the input discourse universes in different Boolean actions that have associated control actions. In this way, fuzzy sets are built from Boolean previously categorized by the designer; thus, the established guidelines are explained below.

4.3.1. Kleene Table Correspondence Criteria

Membership functions in fuzzy sets can be defined considering values $\delta_L, \delta_U \in [0, 1]$ with $\delta_L \leq \delta_U$ (defined by the designer). Thus, the membership function $\mu(x)$ admits to establish a set $v(x)$ with the form (27).

$$v(x) = \begin{cases} 1, & \text{if } \mu(x) \geq \delta_U; \\ u, & \text{if } \delta_L < \mu(x) < \delta_U; \\ 0, & \text{if } \mu(x) \leq \delta_L. \end{cases} \quad (27)$$

The criterion states that when using $v(x)$ it is mandatory to maintain the same codification in the Kleene Table.

4.3.2. Convexity Criteria

Since one of the issues in BBR systems consists of the monotonous transition among Boolean actions, then, it is proposed that fuzzy sets obtained from Booleans must be convex. Considering the transition from Boolean to a fuzzy set A then $\forall x_1, x_2 \in X$ with $\lambda \in [0, 1]$, it is necessary to accomplish (28) (see [29]).

$$\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min[\mu_A(x_1), \mu_A(x_2)] \quad (28)$$

Figure 12 presents an example of membership function associated to convex and non-convex fuzzy sets.

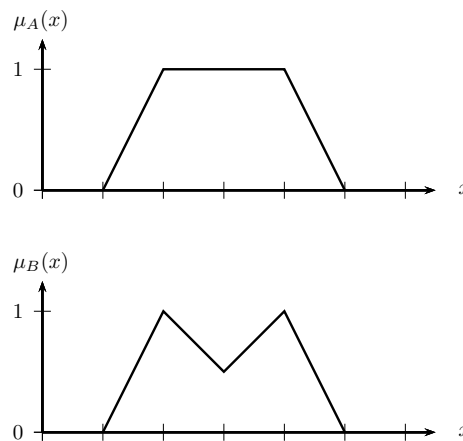


Figure 12. Example of a convex and non-convex fuzzy set.

4.3.3. Boolean Table Correspondence Criteria

This conversion criteria is based on α -cut, where considering a value $\alpha \in [0, 1]$, the α -cut for a fuzzy set A has the elements of X with a membership value μ_A equal to or greater than α [29], then $A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\}$.

Thus, the criterion proposed is based on a level α (given by the designer) where a Boolean set B is established with the membership function $\mu(x)$ using equation (29), where the Boolean set $B(x)$ must match for codification in the Boolean table.

$$B(x) = \begin{cases} 1, & \text{if } \mu(x) \geq \alpha; \\ 0, & \text{if } \mu(x) < \alpha. \end{cases} \quad (29)$$

A case of conversion from a Boolean to a fuzzy set using a level α is shown in Figure 13 producing the memberships $\mu_1(x)$ and $\mu_2(x)$.

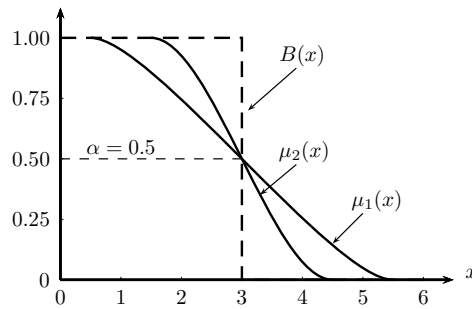


Figure 13. Example of conversion from a Boolean to a fuzzy set using a level α .

5. Conceptual Design Example: Tank Level Control

The purpose of this section is to show the design methodology using a basic application case in such a way that the steps involved in the design process can be clearly observed, being an important aspect the manipulation done with Boolean and Kleene algebras. Therefore, a SISO (Single Input Single Output) hydraulic system is considered as application case.

In hydraulic systems, level control is a classic problem that has been managed using fuzzy logic as observed in the references [45–47], specially, the tank model is based on [48].

In this example, the start point is a Boolean design in which the segmentation of the discourse universe is achieved. The control scheme considers three possible actions based on automatism for filling the tank.

5.1. System Model

The considered dynamic model is similar to that presented in reference [48]. The scheme of the hydraulic system is shown in Figure 14.

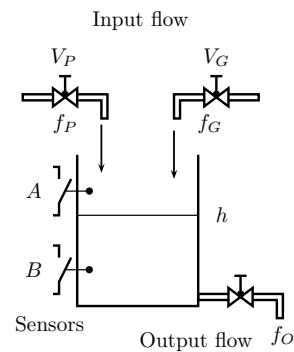


Figure 14. Scheme for a tank filling.

The variables and parameters used are:

- $q_e(t)$: Input flow.
- $h(t)$: Height of liquid level in the tank.
- A_T : Cross-sectional area of the tank.
- A_0 : Cross-sectional area of the output valve.
- $C_v = A_0\sqrt{2g}$: Output valve constant.

The hydraulic system model is described by equation (30), the left part corresponds to input flow, on the right, the first term corresponds to accumulation, and the second to the output flow.

$$q_e(t) = A_T \frac{dh(t)}{dt} + A_0 \sqrt{2gh(t)} \tag{30}$$

Equation (31) presents another form to represent the hydraulic system.

$$q_e(t) = A_T \frac{dh(t)}{dt} + C_v \sqrt{h(t)} \tag{31}$$

For the considered case the parameters are: Height of the tank: 1m; referred height to filling the tank 80 cm; maximum input flow 0.25 L/s or ($2.5 \times 10^{-4} \text{m}^3/\text{s}$); transverse tank area 0.126 m², output pipes diameter 1/4 in or (0,635 cm); transversal area of the output valve 0.317 cm² or ($3.17 \times 10^{-5} \text{m}^2$), constant value for the output valve is $1.4 \times 10^{-4} \text{m}^{5/2}/\text{s}$.

A delay time $t_0 = 0.1\text{s}$ associate to the valve is also considered; therefore, the relation between control action u and input flow is $q_e(t) = u(t - t_0)$. In this model the cross-sectional area of the output valve A_0 is considered constant. Finally, each valve V_G and V_P has a maximum flow associated, which is:

- $V_G = 1.5 \text{ L/s}$
- $V_P = 1.0 \text{ L/s}$

5.2. Boolean Design

A first step in the design consists of analyzing the system dynamic transitions. In this regard, different levels in the tank are considered, their transitions as well as the actions performed in the valves. The levels considered are:

- Low level: L .
- Medium level: M .
- High level: H .

For actions of control there are two valves V_P and V_G , which can be real or virtual depending on hardware implementation. Table 7 is obtained considering the valve actions.

Table 7. Table of events regarding status and valves.

Event	Estate	Consequence
1	Level low L	V_P and V_G activated
2	Level intermediate M	V_P disabled and V_G activated
3	Level high H	V_P activated and V_G disabled

As Boolean actions are already defined, then some inferences can be made:

- **If** level h is low **Then** input flow is large.
- **If** level h is intermediate **Then** input flow is medium.
- **If** level h is high **Then** input flow is low.

For the action valves (V_G large valve and V_P small valve), inferences can be defined as well:

- **If** level h is low **Then** V_G and V_P activated
- **If** level h is intermediate **Then** V_G activated and V_P disabled.
- **If** level h is high **Then** V_G disabled and V_P activated.

For implementing the actions on the valves two activation functions are defined, allowing the opening and closing of valves in such a way that the total action in the tank is:

$$u = Y_{VG} V_G + Y_{VP} V_P \tag{32}$$

5.3. FIS-BBR Design

In this case Boolean sets are replaced by fuzzy sets under certain considerations. Likewise, with fuzzy sets defined, the implementation of relations found in the Boolean design is possible, using specific operations as complement (standard), conjunction (t-norm: minimum), and disjunction (t-conorm: maximum) to fuzzy sets.

5.4. Boolean Sets

Figure 14 considers possible sensors that are activated or diminished when the liquid is lower or higher in a defined level to employ in the control system. Then the Boolean membership functions proposed for implementing the system of control can be seen in Figure 15.

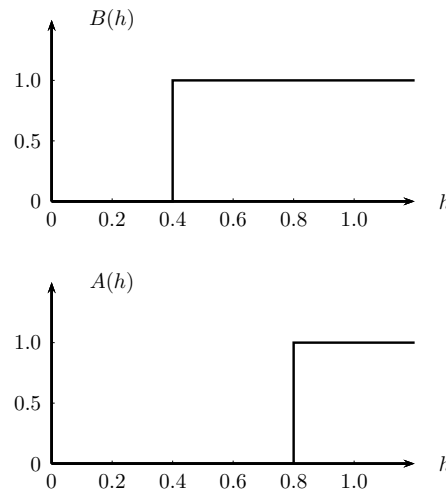


Figure 15. Boolean sets.

The observable regions in the previous Boolean sets are:

- Low: $\mu_B(h) = 0$ and $\mu_A(h) = 0$.
- Medium: $\mu_B(h) = 1$ and $\mu_A(h) = 0$.
- High: $\mu_B(h) = 1$ and $\mu_A(h) = 1$.

Table 8 represents the respective Boolean encoding for the states and control actions. Where X represents the “don’t care” cases; the variable X can be 0 or 1, according to convenience for obtaining a simplified expression.

Table 8. Boolean encoding of actions and states.

B	A	Y_{VG}	Y_{VP}
0	0	1	1
1	0	1	0
1	1	0	1
0	1	X	X

Considering DNF for Boolean algebra [19], from Table 8 (taken $X = 0$) the equations for the activation functions are:

$$Y_{VG} = (B' \wedge A') \vee (B \wedge A') \tag{33}$$

$$Y_{VP} = (B' \wedge A') \vee (B \wedge A) \tag{34}$$

When applying synthesis and considering the case “don’t care” ($B = 0, A = 1$), Table 9 is used to obtain a better simplification. The case “don’t care” is used only to simplify Y_{VP} .

Table 9. Boolean encoding using values for X.

B	A	Y_{VG}	Y_{VP}
0	0	1	1
1	0	1	0
1	1	0	1
0	1	0	1

Considering Table 9, the following equations are obtained:

$$Y_{VG} = A' \tag{35}$$

$$Y_{VP} = B' \vee A \tag{36}$$

For the extension from a Boolean to a Kleenean Table, a value u is considered corresponding to the transition between values 0 and 1 (see Figure 16). For Y_{VG} and Y_{VP} from Boolean Table 9 to Kleene table, first is considered the regularity condition obtaining the Table 10. In the second stage, Table 11 is stated considering the monotonic transitions between Boolean actions. Table 12 shows another way to represent Table 11.

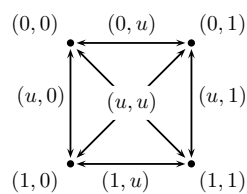


Figure 16. Boolean transitions using u .

Table 10. Possible Kleenean codification for states and actions.

B	A	Y_{VG}	Y_{VP}
0	0	1	1
u	0	$\{1, u\}$	u
1	0	1	0
1	u	u	u
1	1	0	1
u	1	$\{0, u\}$	$\{1, u\}$
0	1	0	1
0	u	u	$\{1, u\}$
u	u	u	u

Table 11. Final Kleenean codification for states and actions.

<i>B</i>	<i>A</i>	<i>Y_{VG}</i>	<i>Y_{VP}</i>
0	0	1	1
<i>u</i>	0	1	<i>u</i>
1	0	1	0
1	<i>u</i>	<i>u</i>	<i>u</i>
1	1	0	1
<i>u</i>	1	0	1
0	1	0	1
0	<i>u</i>	<i>u</i>	1
<i>u</i>	<i>u</i>	<i>u</i>	<i>u</i>

Table 12. Alternative Kleenean codification for states and actions.

	<i>Y_{VG}</i>			<i>Y_{VP}</i>			
<i>B</i> \ <i>A</i>	0	<i>u</i>	1	<i>B</i> \ <i>A</i>	0	<i>u</i>	1
0	1	<i>u</i>	0	0	1	1	1
<i>u</i>	1	<i>u</i>	0	<i>u</i>	<i>u</i>	<i>u</i>	1
1	1	<i>u</i>	0	1	0	<i>u</i>	1

Applying Kleene algebra properties (see [7]), the following equation results for *Y_{VG}*.

$$\begin{aligned}
 Y_{VG} &= (B' \wedge A') \vee A' \vee (B \wedge A') \vee (B \wedge A' \wedge A) \\
 &\quad \vee (B' \wedge A' \wedge A) \vee (B' \wedge B \wedge A' \wedge A) \\
 Y_{VG} &= A' \wedge [B' \vee 1 \vee B \vee (B \wedge A) \vee (B' \wedge A) \\
 &\quad \vee (B' \wedge B \wedge A)] \\
 &= A'
 \end{aligned}
 \tag{37}$$

This is obtained due to absorption properties. Likewise, for *Y_{VP}* is obtained:

$$\begin{aligned}
 Y_{VP} &= (B' \wedge A') \vee (B' \wedge B \wedge A') \vee (B \wedge A' \wedge A) \\
 &\quad \vee (B \wedge A) \vee A \vee (B' \wedge A) \vee B' \\
 &\quad \vee (B' \wedge B \wedge A' \wedge A) \\
 Y_{VP} &= B' \wedge [A' \vee (B' \wedge A') \vee A \vee 1] \\
 &\quad \vee A \wedge [(B \wedge A') \vee B \vee 1 \vee (B' \wedge B \wedge A')] \\
 Y_{VP} &= B' \vee A
 \end{aligned}
 \tag{38}$$

As observed, this focus brings equal expressions to the Boolean case when simplification is applied. It is remarkable that the proposed design methodology allows to obtain this equivalence between Boole and Kleene cases, which is an important aspect of fuzzy systems based on Boolean relations because the design starts in a Boolean domine.

In this paper, it is possible to observe that even though the property used to simplify two elements with the table (third excluded) is different from that used for the three elements table (absorption), it is possible to have the same expressions considering the respective intermediate transitions. The algebraic characteristics that do not meet the fuzzy sets in regard of those defined for Boolean sets are given

by Equations (5) and (6), in this way, is important the employment of Kleene algebra to develop the design process when using fuzzy sets.

The equivalence between Boolean and fuzzy design is appreciable when similar expressions are obtained using Boolean and Kleene algebras, and under the considered criteria the monotonic transitions between Boolean actions are achieved.

5.5. FIS-BBR Implementation

According to considerations for fuzzy set selection (Section 4.3), values $\delta_U = 1$ and $\delta_L = 0$, as well as a level of 0.5 (α -cut) are taken aiming to find an equivalence with the Kleene Table. In the same way, Figure 17 proposes fuzzy sets for sensors that measure the level of liquid present in the tank by considering a linear-saturated behavior.

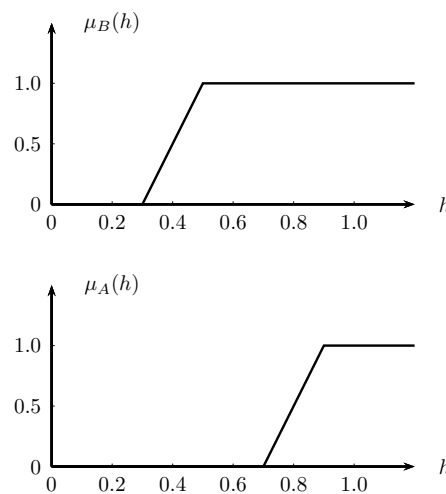


Figure 17. Proposed fuzzy sets.

The simplified equations to system implementation are:

$$Y_{VG} = \mu_{A'}$$

$$Y_{VP} = \mu_{B'} \oplus \mu_A$$

Finally, taken the Equation (32) and using “min” for t-norm, “max” for t-conorm and standard complement, then the control action corresponds to:

$$u = V_G (1 - \mu_A) + V_P \max[(1 - \mu_B), \mu_A] \tag{39}$$

In order to show the processes involved in the operation of the Boolean and BBR controllers, Figures 18 and 19 show the respective block diagrams. In Figure 18 is presented the associated Boolean sets for Boolean controller, meanwhile Figure 19 shows the membership functions used for the BBR controller.

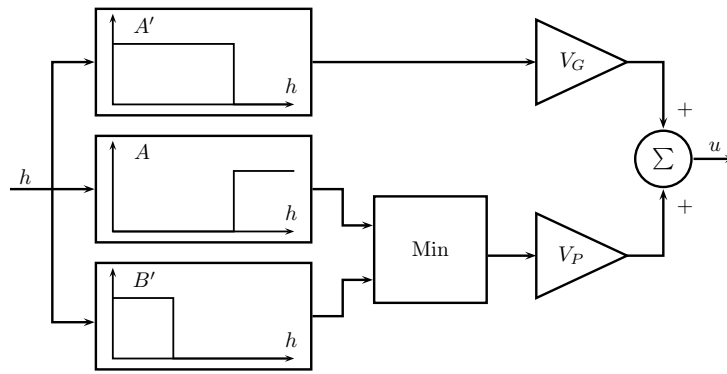


Figure 18. Block diagram associated to Boolean controller.

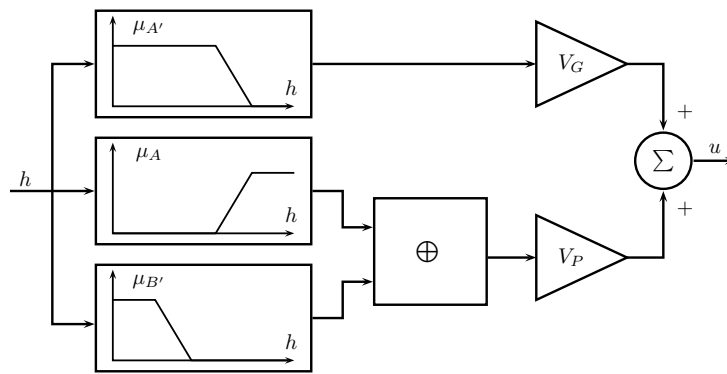


Figure 19. Respective block diagram for BBR controller.

5.6. Results

The noteworthy results correspond to the characteristic transfer shape of the control surface (curve in this case) to implementing Boolean and BBR controller. These transfer curves are observable in Figure 20. Another relevant result is related to the control system dynamic response for the controllers; this can be seen in Figure 21.

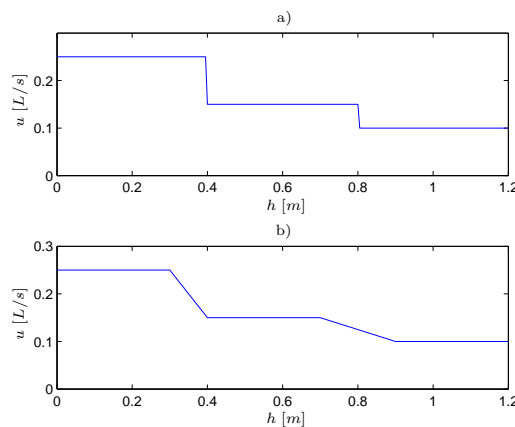


Figure 20. Transfer curve for: (a) Boolean, (b) BBR.

Likewise, Figure 20 shows the characteristic transfer curve of the BBR controller, displaying monotonous behavior among Boolean actions according to design.

Figure 21 shows a Boolean controller typical switching behavior, BBR controller shows a softer behavior than the Boolean. A detail of these characteristics is shown in Figure 22, where an oscillatory behavior is observed for the Boolean controller while the BBR controller presents a continuous and

smooth behavior. In control systems where the actuator is sensitive to sudden changes this is a factor of importance in the design of the system. According to [49] the oscillatory behavior associated with cycle limit is common in ON-OFF control systems.

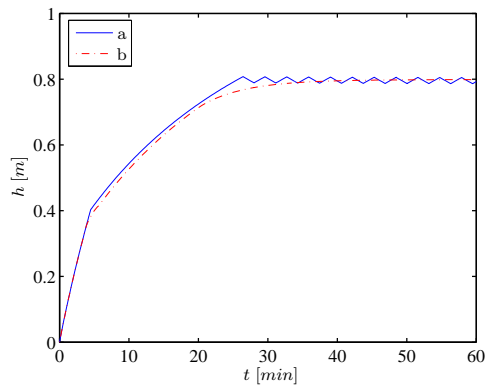


Figure 21. System dynamic responses for a reference of 80 cm: (a) Boolean, (b) BBR.

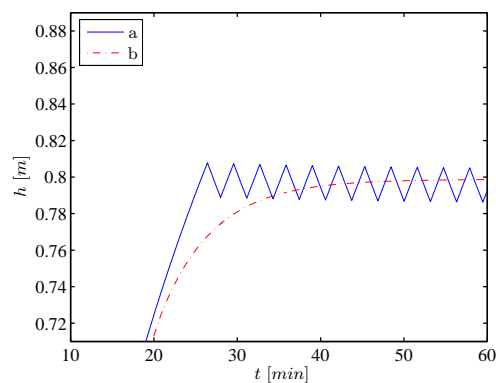


Figure 22. Detail responses: (a) Boolean, (b) BBR.

Figure 23 shows the control actions for Boolean and fuzzy designs. The fuzzy controller exhibits continuous transitions in comparison with Boolean. The implementation of a Boolean controller uses ON-OFF sensors and actuators; however, when carrying out the fuzzy implementation, these sensors and actuators are required to have a continuous behavior according to the design.

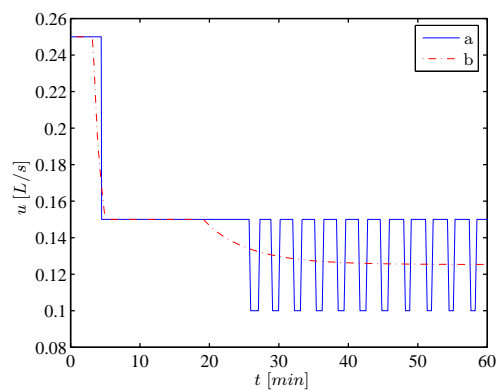


Figure 23. Control actions: (a) Boolean, (b) BBR.

The system performance indicators are contained in Table 13, being $\%e_{ss}$ equal to the steady state error, T_s settling time and finally the oscillation feature. The T_s value is calculated when the

signal is within a range with 5% of the reference. The reference considered for simulations is 80 cm, from Table 13 the minor steady-state error is for the BBR system; the lower T_s is achieved using the Boolean controller and not oscillation for BBR system. It should be noted that if the oscillations of the Boolean controller exceed the band of 5%, the settling time cannot be calculated.

Table 13. System indicators of performance taking a reference of 80 cm.

Feature \ Control	Boolean	BBR
$\%e_{ss}$	–	0.25%
T_s	25.5 min	31.5 min
Oscillation	Yes	Not

6. Discussion

Concerning BBR systems, the most relevant characteristic lies in their automatism designs for control systems; under this feature, an extension of a Boolean to a Kleene Table with a further implementation of fuzzy sets is made. Concerning the structure, this is at times similar to non-normalized Takagi–Sugeno fuzzy system; however, BBR systems include the respective design methodology shown in section 4. Moreover, including those characteristics in BBR systems offers the following advantages:

- Design methodology rooted in tables of truth which is obtained taking Table 3 as reference, described in Section 3.2. This also allows to simplify the design by using Kleene algebra.
- A simplified inference process, which is seen in Figures 1, 3 and 5 where the fuzzy systems of Mamdani, Takagi–Sugeno and BBR appear.
- Direct application in control systems, which is observed in Section 3, including the concept of virtual actuator shown in Section 3.1. This concept grants a compact inference process oriented toward control.
- Applicability in neuro-fuzzy supervised systems using the structure in Figure 7, and Equations (16) and (17). Training parameters in a supervised form are achieved when using continuous membership functions, and fuzzy operators as both, the t-norm product and the algebraic sum to t-conorm.

This paper is focused on presenting the design methodology. The possibility of deepening and studying other related aspects is open. Some of the topics to study in later developments are:

- Implementation of BBR using CNF (Conjunctive Normal Form) considering a similar methodology as it is employed for the design of digital circuits.
- Refinement of criteria to convert Boolean sets to fuzzy.
- Deepen in the considerations to use general forms of t-norm, t-conorm and complement. In this paper, we considered the standard operators.
- Taking this work as a reference, the development of more complex cases for systems with multiple poles, zeros, and non-linear behaviors can be performed, considering also the development of supervised neuro-fuzzy systems.

This document presents the methodology with a basic application case seeking to show the algebraic operations used in the design process. For more complex applications related to the FIS-BBR, can be considered [15] where a neuro-fuzzy control system for the Control of a Permanent Magnet Synchronous Generator is presented. Additionally, [16] presents an adaptive scheme to the Control Flow Power in Distributed Generation Systems. These works use a compact fuzzy scheme equivalent to an FIS-BBR, which is determined in a heuristic way taking a discrete linear time system which is converted to a compact fuzzy system. This systems could be formulated using the methodology proposed in this document which is expected to be addressed in a further work.

7. Conclusions

This paper introduces some issues for system implementation of fuzzy inference based on Boolean relations, whose approach aims to transform Boolean sets into fuzzy for the implementation of fuzzy inference systems. In this regard, regularity conditions and the criteria to have continuous (monotonic) transitions between Boolean actions are used to transform a Boolean Table into a Kleene Table, as well as to define the shape of fuzzy sets.

Using the set of criteria proposed, together with Boolean and Kleene algebras, we observed the possibility to achieve an equivalence between Boolean and fuzzy designs obtaining continuous transitions between Boolean actions.

The proposal considers Boolean algebra tools to design automatic control systems based on truth tables, allowing the possibility of simplifying equations. This methodology allows the design of fuzzy compact systems which are expected to be studied in future work.

In an FIS-BBR implementation, to achieve smooth transitions between Boolean actions, sensors and actuators are required to have a continuous behavior according to the design that is carried out. Considering the application case, the sensors and the actuators must present an analog behavior for tank level control.

The design example was made for a specific set of parameters (tank model). However, when parameter variations occur, it is convenient to implement an adaptive system. In this way, the application of control systems using Boolean relations in neuro-fuzzy adaptive control is expected in future studies.

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 & editing, H.E. and I.M.

Funding: This research received no external funding.

Acknowledgments: A recognition for Universidad de Oviedo and the Universidad Distrital Francisco José de Caldas. An additional acknowledgment to Ph.D. Irene Díaz and Ph.D. Omar Salazar for the review of some mathematical aspects.

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

References

1. Piegat, A. Fuzzy modeling and control. *Stud. Fuzziness Soft Comput.* **2001**, *69*, 1–728.
2. Nguyen, H.T.; Nadipuram, R.P.; Walker, C.L.; Walker, E.A. *A First Course in Fuzzy and Neural Control*, 1st ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2002.
3. Cerman, O. Fuzzy model reference control with adaptation mechanism. *Expert Syst. Appl.* **2013**, *40*, 5181–5187. [[CrossRef](#)]
4. Dougherty, E.; Giardina, C. *Mathematical Methods for Artificial Intelligence and Autonomous Systems*; 1st ed.; Prentice Hall: Bergen County, NJ, USA; 1988.
5. Garnier, R.; Taylor, J. *Discrete Mathematics: Proofs, Structures and Applications*, 3rd ed.; CRC Press: Boca Raton, FL, USA, 2009.
6. Whitesitt, J.E. *Boolean Algebra and Its Applications*; Dover Publications: Mineola, NY, USA; 2012.
7. Nguyen, H.T.; Walker, E.A. *A First Course in Fuzzy Logic*, 3rd ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2006.
8. Espitia, H.; Morales, L.; Soriano, J. Design and simulation of a fuzzy controller for a DC motor based on boolean relations. *Visión Electrónica* **2010**, *4*, 12–22.
9. Chamorro, H.R.; Díaz, N.L.; Soriano, J.J.; Espitia, H.E. Active and reactive power flow fuzzy controller for VSC HVDC using DBR and DBR type 2. In Proceedings of the IEEE Annual Meeting of the North American Fuzzy Information Processing Society, El Paso, TX, USA, 18–20 March 2011.
10. Espitia, H.E.; Chamorro, H.R.; Soriano, J.J. Fuzzy controller design using concretion based on boolean relations (CBR). In Proceedings of the 12th UK Workshop on Computational Intelligence (UKCI), Edinburgh, UK, 5–7 September 2012.

11. Espitia, H.E.; Soriano, J.J. Design and simulation of a fuzzy temperature controller using the concesor based on boolean relationship. *Tecnura* **2012**, *16*, 29–40.
12. Espitia, H.E. Aplicación del Concesor Basado en Relaciones Booleanas Para Sistemas de Lógica Difusa Tipo Dos. Tesis de Maestría en Ingeniería Industrial, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia, 2009.
13. Espitia, H.E.; Soriano, J.J. Sistema de inferencia difusa basado en relaciones booleanas. *Ingeniería* **2010**, *15*, 52–66.
14. Salazar, O.; Soriano, J.; Serrano, H. From binary logic functions to fuzzy logic functions. *Appl. Math. Sci.* **2013**, *7*, 5129–5138. [[CrossRef](#)]
15. Espitia, H.; Díaz, G.; Díaz, S. Control of a Permanent Magnet Synchronous Generator Using a Neuro-Fuzzy System. *Commun. Comput. Inf. Sci. Appl. Comput. Sci. Eng.* **2018**, *915*, 89–101.
16. Espitia, H.; Machón, I.; López, H.; Díaz, G. Proposal of an Adaptive Neurofuzzy System to Control Flow Power in Distributed Generation Systems. *Complexity* **2019**, *2019*, 1–16. [[CrossRef](#)]
17. Ruiz, A.; Molina, J.M. *Automatización y Telecontrol de Sistemas de Riego*; Marcombo S.A.: Barcelona, Spain, 2010.
18. Gehrke, M.; Walker, C.; Walker, E. A mathematical setting for fuzzy logic. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **1997**, *5*, 223–238. [[CrossRef](#)]
19. Gehrke, M.; Walker, C.; Walker, E. Normal forms and truth tables for fuzzy logics. *Fuzzy Sets Syst.* **2003**, *138*, 25–51. [[CrossRef](#)]
20. Klir, G.; Wheeler, B. *Fuzzy Sets and Fuzzy Logic*; Prentice Hall: Bergen County, NJ, USA, 1995.
21. Kabir, S.; Papadopoulos, Y. A review of applications of fuzzy sets to safety and reliability engineering. *Int. J. Approx. Reason.* **2018**, *100*, 29–55. [[CrossRef](#)]
22. Goodstein, R.L. *Boolean Algebra*; Dover Publications: Mineola, NY, USA, 2007.
23. Zadeh, L.A. Fuzzy sets. *Inf. Control.* **1965**, *8*, 338–353. [[CrossRef](#)]
24. Ngan, S.C. Revisiting fuzzy set operations: A rational approach for designing set operators for type-2 fuzzy sets and type-2 like fuzzy sets. *Expert Syst. Appl.* **2018**, *107*, 255–284. [[CrossRef](#)]
25. Kashyap, P.K.; Kumar, S.; Dohare, U.; Kumar, V.; Kharel, R. Green Computing in Sensors-Enabled Internet of Things: Neuro Fuzzy Logic-Based Load Balancing. *Electronics* **2019**, *8*, 384. [[CrossRef](#)]
26. Klement, E.; Mesiar, R.; Pap, E. Triangular Norms. *Trends Log.* **2000**, *8*, 1–387.
27. Troiano, L.; Rodríguez, L.J.; Marinaro, P.; Díaz, I. Statistical analysis of parametric t-norms. *Inf. Sci.* **2014**, *257*, 138–162. [[CrossRef](#)]
28. Clark, T.D.; Larson, J.M.; Mordeson, J.N.; Potter, J.D.; Wierman, M. *Applying Fuzzy Mathematics to Formal Models in Comparative Politics*; Studies in Fuzziness and Soft Computing; Springer: Berlin/Heidelberg, Germany, 2008.
29. Zhang, R.; Phillis, Y.; Kouikoglou, V. *Fuzzy Control of Queuing Systems*; Springer: London, UK, 2005.
30. Wang, L. *A Course on Fuzzy Systems and Control*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 1997.
31. Li, J.; Gong, Z. SISO Intuitionistic Fuzzy Systems: IF-t-Norm, IF-R-Implication, and Universal Approximators. *IEEE Access* **2019**, *7*, 70265–70278. [[CrossRef](#)]
32. Fayaz, M.; Ullah, I.; Kim, D. An Optimized Fuzzy Logic Control Model Based on a Strategy for the Learning of Membership Functions in an Indoor Environment. *Electronics* **2019**, *8*, 132. [[CrossRef](#)]
33. Kofinas, P.; Dounis, A.I. Online Tuning of a PID Controller with a Fuzzy Reinforcement Learning MAS for Flow Rate Control of a Desalination Unit. *Electronics* **2019**, *8*, 231. [[CrossRef](#)]
34. Siddique, N.; Adeli, H. *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*; Wiley: Hoboken, NJ, USA, 2013.
35. Cueva, G.; Espada, J.; García, V.; Gonzalez, R. Fuzzy decision method to improve the information exchange in a vehicle sensor tracking system. *Appl. Soft Comput.* **2015**, *35*, 708–716. [[CrossRef](#)]
36. Zaheeruddin; Jain, V.K. A fuzzy expert system for noise-induced sleep disturbance. *Expert Syst. Appl.* **2006**, *30*, 761–771. [[CrossRef](#)]
37. Mandani, E.H. Application of fuzzy algorithms for control of simple dynamic plant. *Proc. Inst. Electr. Eng.* **1974**, *121*, 1585–1588. [[CrossRef](#)]
38. Keshwani, D.R.; Jones, D.D.; Meyer, G.E.; Brand, R.M. Rule-based Mamdani-type fuzzy modeling of skin permeability. *Appl. Soft Comput.* **2008**, *8*, 285–294. [[CrossRef](#)]

39. Matía, F.; Al-Hadithi, B.M.; Jiménez, A.; Segundo, P.S. An affine fuzzy model with local and global interpretations. *Appl. Soft Comput.* **2011**, *11*, 4226–4235. [[CrossRef](#)]
40. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modelling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *15*, 116–132. [[CrossRef](#)]
41. Czogala, E.; Leski, J. Fuzzy and neuro-fuzzy intelligent systems. *Stud. Fuzziness Soft Comput.* **2000**, *47*, 1–195.
42. Gehrke, M.; Walker, C.; Walker, E. Some Comments on Interval Valued Fuzzy Sets. *Int. J. Intell. Syst.* **1996**, *11*, 751–759. [[CrossRef](#)]
43. Klementa, E.P.; Mesiar, R.; Pap, E. Triangular norms. Position paper I: Basic analytical and algebraic properties. *Fuzzy Sets Syst.* **2004**, *143*, 5–26. [[CrossRef](#)]
44. 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](#)]
45. Boulkaibet, I.; Belarbi, K.; Bououden, S.; Marwala, T.; Chadli, M. A new T-S fuzzy model predictive control for nonlinear processes. *Expert Syst. Appl.* **2017**, *88*, 132–151. [[CrossRef](#)]
46. Precup, R.E.; Tomescu, M.L.; Radac, M.B.; Petriu, E.M.; Preitl, S.; Dragos, C.A. Iterative performance improvement of fuzzy control systems for three tank systems. *Expert Syst. Appl.* **2012**, *39*, 8288–8299. [[CrossRef](#)]
47. Sadeghi, M.S.; Safarinejadian, B.; Farughian, A. Parallel distributed compensator design of tank level control based on fuzzy Takagi–Sugeno model. *Appl. Soft Comput.* **2014**, *21*, 280–285. [[CrossRef](#)]
48. Xiao, Q.; Zou, D.; Wei, P. Fuzzy Adaptive PID Control Tank Level. In Proceedings of the International Conference on Multimedia Communications, Hong Kong, China, 7–8 August 2010.
49. Doebelin, E. *Control System Principles and Design*, 4th ed.; McGraw Hill Education: New York, NY, USA, 2012.



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).