

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

170,000

International authors and editors

185M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



## Chapter

# Methodology for the Implementation of a Fuzzy Controller on Arduino, MATLAB™ and Nexys 4™ Platforms

*Jesus de la Cruz-Alejo, Hugo Beatriz-Cuellar,  
Agustin Mora-Ortega and Maria Belem Arce-Vazquez*

## Abstract

This chapter presents a methodology to implement a fuzzy controller in different hardware platforms, which can be used to control a system or process. The methodology proposes a programming algorithm to implement a fuzzy controller on the Arduino UNO, Arduino DUE, Nexys 4™, and MATLAB™ platforms. The programming algorithm uses two control statements (IF-THEN and FOR) and the basic mathematical operations. The fuzzy controller was designed for two input variables, one output variable, five fuzzy sets for each variable, and a Mamdani type structure. An analysis of convergence time, amount of memory, and control surface is performed to ensure that the fuzzy controller on all platforms is satisfactory. MATLAB™ is used to compare these platforms through numerical simulations, which demonstrates the effectiveness of the proposed methodology. The experimental results of the fuzzy controller are a processing time of 117 milliseconds and 40% of the memory of the Arduino UNO, a processing time of 21.275 milliseconds and 5% of the memory of the Arduino DUE, and a processing time of 17.871 milliseconds and 40% of the memory on the Nexys 4™. Finally, a Mean Square Error of 0.0326, 0.0643, and 0.1125 was obtained for MATLAB™, Arduino, and Nexys 4™, respectively.

**Keywords:** fuzzy controller, methodology, MATLAB™, Arduino, Nexys 4™

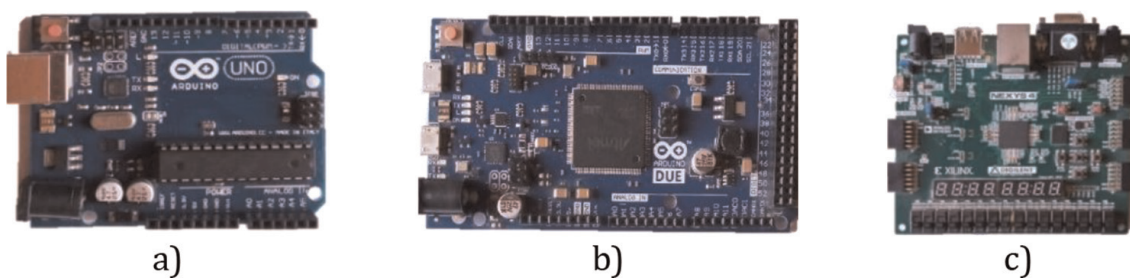
## 1. Introduction

Currently, most of the processes use a control system, which provides the necessary conditions and guarantees the correct operation of the process to obtain the final product. Additionally, there is a great variety of control systems, for example, neural networks, PID controllers, robust control, sliding modes, PLC (Programmable Logic Controllers), fuzzy controller, among others. On the other hand, the characteristics of the process must be analyzed to select a control system, for example, the cost of the control system, the software or hardware used for the implementation of the control

system, the mathematical requirements used to analyze the process, variables, types of sensors and actuators necessary to control the process, desired precision in the process, advantages and/or disadvantages of the control system, among other things. Therefore, this work shows a methodology for the implementation of a fuzzy controller in different software or hardware platforms, since a fuzzy controller does not need the mathematical model of the system, uses the experience or knowledge of a person, does not use complex mathematical equations for its implementation, and uses linguistic explanations (low, high, hot, cold, good, bad, etc.) to define process conditions and control action. Therefore, a fuzzy controller is one of the best options for controlling a process. Currently, fuzzy controllers are used in a wide variety of processes or applications; for example, modeling and simulation of the Maximum Power Point Tracking (MPPT) in photovoltaic solar energy systems [1–5], increase the accuracy in determining the degree of diabetes in a person [6, 7], identification of hot spots and analysis of the intensity of flames in pipes to prevent fires [8], improve the performance of a grid-connected wind generator system [9–11], control of the output voltage of a Boost converter [12], generate a suitable microclimate for an agricultural greenhouse [13], among other applications. On the other hand, the proposed methodology uses two control statements (IF-THEN and FOR), and the basic mathematical operations (addition, subtraction, multiplication, and division) for the design and implementation of the fuzzy controller stages. Therefore, the proposed methodology uses basic programming elements, which allows the fuzzy controller to be implemented in different software or hardware platforms. In this work, MATLAB™ and the Arduino UNO, Arduino DUE, and Nexys 4™ boards are used to show the correct operation of the proposed methodology. Also, Fuzzy Logic Toolbox™ is used to simulate and analyze the operation of the fuzzy controller. Finally, the structure of the chapter is as follows, Section 2 shows the procedure to implement the fuzzy controller in the different platforms, Section 3 shows the simulation and experimental results of the fuzzy controller, which was implemented in the different platforms, and Section 4 presents the conclusions.

## **2. Description of the methodology**

The methodology proposes a programming algorithm, which allows implementing a fuzzy controller on different hardware and software platforms, which have different technical characteristics. In this case, MATLAB™ and the Arduino UNO, Arduino DUE, and Nexys 4™ boards are used to show the correct operation of the proposed methodology. The Arduino UNO board uses the ATmega328P microcontroller, which has a 32 KB Flash memory, a 2 KB SRAM memory, and a 1 KB EEPROM memory. In addition, this board has a 16 MHz clock speed, 14 digital input or output pins, 6 analog inputs with a 10-bit resolution, and 8 PWM outputs, and its programming language is based on the C/C++ language. The Arduino DUE board uses the SAM3X8E microcontroller, which has a 512 KB Flash memory and a 96 KB SRAM memory. In addition, this board has a clock speed of 84 MHz, 54 digital input or output pins, 12 analog inputs with a 12-bit resolution, 2 digital-analog converters, and 12 PWM outputs, and its language of programming is based on the C/C++ language [14, 15]. The Nexys 4™ board model XC7A100T-1CSG324C contains two external memories, a 128Mbit cellular RAM and a 128Mbit non-volatile serial Flash device. In addition, this board has a clock speed of 100 MHz, USB ports, an Ethernet port, a micro-SD port, a micro-USB port, a VGA port, accelerometer, temperature sensor, digital microphone, speaker

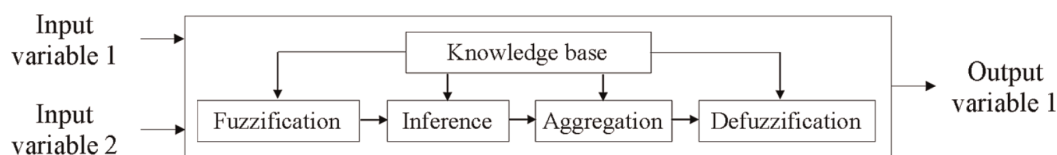


**Figure 1.** Board of (a) Arduino UNO, (b) Arduino DUE, and (c) Nexys 4™.

amplifier, 16 user switches, 16 user LEDs, two 4-digit 7-segment displays, two three-color LEDs and its programming language is VHDL [16, 17]. Additionally, the ISE Project Navigator 14.7 software was used to program the Nexys 4™ board, and the Arduino IDE (Integrated Development Environment) was used to program the Arduino boards. **Figure 1** shows the Arduino UNO, Arduino DUE, and Nexys 4™ boards. On the other hand, the fuzzy controller implementation is done in the MATLAB™ Script, which is a program file that allows building a programming algorithm and provides tools for displaying graphics in two and three dimensions. Additionally, the MATLAB™ Script is used for the analysis, design, and simulation of the different stages of the controller. Also, the fuzzy controller is implemented in Fuzzy Logic Toolbox™, which is a MATLAB™ tool used to design, analyze, and simulate a fuzzy controller. Therefore, the Fuzzy Logic Toolbox™ is used to analyze the results obtained from the implementation of the controller on the different platforms [18, 19]. Finally, this boards can be used to control a process, and the selection of the board will depend on the characteristics of the process, for example, the number of variables, sensors or actuators, the cost of the board, the cost of the system, among other things.

## 2.1 Fuzzy controller design

The design of a fuzzy controller of the Mamdani type is carried out, which is made up of the fuzzification, inference, aggregation, and defuzzification stages. On the other hand, a fuzzy controller can use more than one input variable and can determine more than one output variable; however, a fuzzy controller needs at least two input variables and one output variable to function properly. If the number of input and output variables of the fuzzy controller is increased, then the complexity of the fuzzy controller implementation will increase. **Figure 2** shows the structure of a fuzzy controller of the Mamdani type. The design and implementation of a fuzzy controller begins with the definition of the controlled variable of the process, for example,



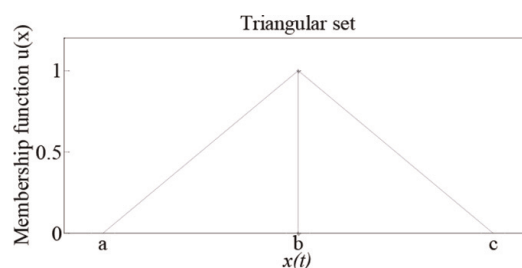
**Figure 2.** Structure of a fuzzy controller of the Mamdani type.

temperature, humidity, pressure, pH, among others. Subsequently, the input and output variables of the controller must be defined. For this, it must be considered that the input variables of the fuzzy controller are used to measure the state or condition of the process, and the output variable of the fuzzy controller is the control action, which will be used to adjust the controlled variable. Also, a universe of discourse must be specified for each of the controller variable, which can be defined as the range of values, where a specific value of the input or output variables can be found or located. In other words, a universe of discourse is made up of the values that are between the minimum and maximum values of a variable. On the other hand, a series of linguistic values (low, high, good, bad, etc.) must be defined, which describe the state or condition of the input and output variables of the controller. Subsequently, within the universes of discourse of the controller variables, fuzzy sets must be defined, which must be labeled with the name of the linguistic values. Also, the type of fuzzy set that will be used to implement the controller must be defined. For this, the computational load and the necessary programming elements must be considered. Finally, these elements should be considered as the initial parameters of the fuzzy controller [20].

The fuzzy controller should be used for a specific situation since this is the correct way to show how to implement the fuzzy controller. Therefore, the fuzzy controller is used to determine the tip of a food establishment, since this application is the simplest to understand the operation of the fuzzy controller. This application will allow to characterize the controller; that is, this application allows defining the input and output variables of the controller, the length of the universes of discourse, type of fuzzy sets, dimensions of fuzzy sets, among other things. In this case, the value of the tip depends on the quality of the food and the service of the food establishment. Therefore, a signal defined as “food” and a signal defined as “service” are used as controller input variables, and a signal defined as “tip” is used as the controller output variable. A universe of discourse from 0 to 100 was used for the input variables since food and service can be evaluated with a score of 0 to 100. A universe of discussion from 0 to 120 was used for the output variable, which represents the amount of money from 0 to \$120. However, the minimum value of the tip will be \$20, and the maximum value of the tip will be \$100.

## 2.2 Fuzzy sets of controller variables

There is a great variety of types of fuzzy sets, which can be used for the implementation of a fuzzy controller. **Figure 3** shows a triangular type of fuzzy set, which is used to define the fuzzy sets of the input and output variables of the controller. This type of fuzzy set uses the subtraction and division operations for its implementation in software and hardware.



**Figure 3.**  
Triangular fuzzy set.

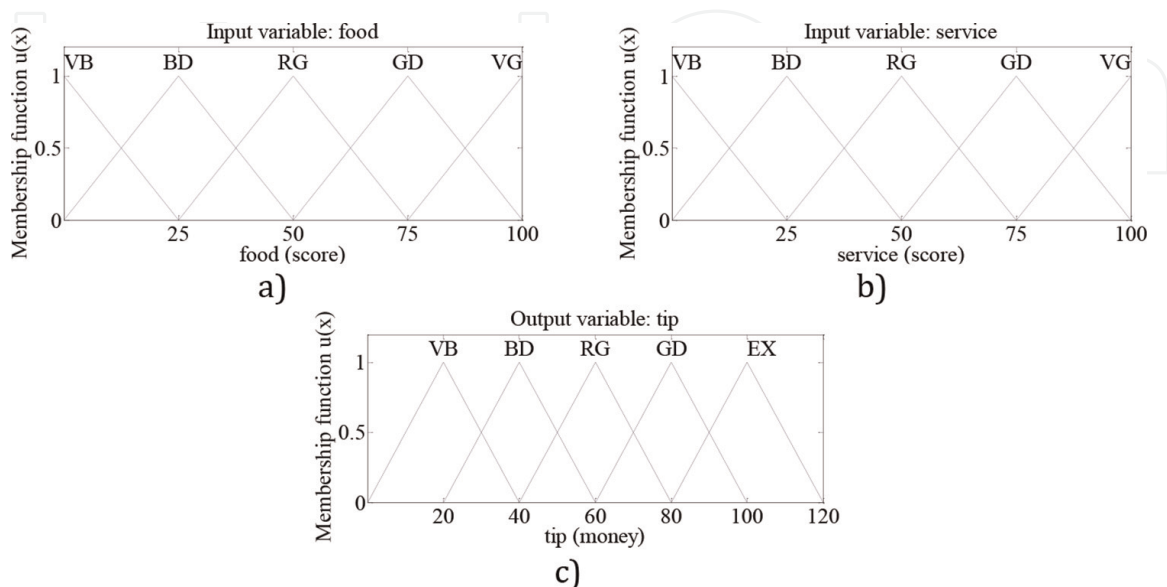


$$\mu(x) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{x - a}{b - a} & \text{si } a < x \leq b \\ \frac{c - x}{c - b} & \text{si } b < x \leq c \\ 0 & \text{si } x \geq c \end{cases} \quad (1)$$

On the other hand, the accuracy of the process state measurement depends on the number of fuzzy sets of the input variables. Similarly, the accuracy of the control action to adjust the process depends on the number of fuzzy sets of the output variable. Therefore, a process analysis must be performed to determine the number of fuzzy sets for the input and output variables of the controller. In this case according to the approximation error through simulations using five and seven, the approximation error between five and seven was minimal. So, five fuzzy sets were defined for the variable “food” labeled Very Bad (VB), Bad (BD), Regular (RG), Good (GD), and Very Good (VG). Also, five fuzzy sets were defined for the variable “service” labeled Very Bad (VB), Bad (BD), Regular (RG), Good (GD), and Very Good (VG). Also, five fuzzy sets were defined for the output variable “tip” labeled Very Bad (VB), Bad (BD), Regular (RG), Good (GD), and Excellent (EX). The dimensions of the fuzzy sets depend on the importance of the control action; therefore, the length of the fuzzy sets can be different. Finally, **Figure 4** shows the dimensions of the fuzzy sets of the variables of the fuzzy controller.

### 2.3 Fuzzification stage

The first stage of the fuzzy controller is fuzzification, which is used to transform a real variable (“food” or “service”) into a fuzzy variable through the membership functions. Fuzzification determines the fuzzy sets that indicate the state or condition of an input variable. For this, the membership function  $\mu(x)$  must be determined using Eq. (1) [21–23]. The algorithm used to implement the fuzzification in the



**Figure 4.** Fuzzy sets of the variables (a) food, (b) service, and (c) tip.

MATLAB™ Script and on the Arduino UNO, Arduino DUE, and Nexys 4™ boards is described below.

<p>Eq. (1) is used to determine the <math>\mu(x)</math> of the input variables (food or service), and IF-THEN conditional statements are used to determine the fuzzy sets that define the condition of the input variables. FD and SV are the value of food and service, respectively. Setfd and Setsv store the fuzzy set that indicates the state of the input variables, Mf1 and Mf2 are the membership functions of the input variables, and a1, a2, b1, and b2 are dimensions of a fuzzy set.</p>		
<pre>% MATLAB Script if FD &gt;= a1 &amp;&amp; FD &lt;= b1 Setfd = VB; Mf1=(FD - a1) / (b1 - a1); end if SV &gt;= a2 &amp;&amp; SV &lt;= b2 Setsv = VB; Mf2=(SV - a2) / (b2 - a2); end</pre>	<pre>// Arduino UNO/DUE if (FD &gt;= a1 &amp;&amp; FD &lt;= b1) { Setfd =VB; Mf1=(FD - a1) / (b1 - a1); } if (SV &gt;= a2 &amp;&amp; SV &lt;= b2) { Setsv =VB; Mf2=(SV - a2) / (b2 - a2); }</pre>	<pre>- Nexys 4™ if (FD &gt;= a1 and FD &lt;= b1) then Setfd := VB; Mf1:=((FD-a1)*(100))/(b1-a1); end if; if (SV&gt;=a2 and SV&lt;=b2) then Setsv := VB; Mf2:=((SV-a2)*(100))/(b2-a2); end if;</pre>

## 2.4 Inference stage

The inference stage uses fuzzy rules, which represent the knowledge base of a fuzzy controller and determine the controllability of the process. Fuzzy rules relate the membership functions of the fuzzy sets that the input variables have. The result of a fuzzy rule is a fuzzy set contained in the output variable obtained using the Mamdani implication. The membership functions of the input variable fuzzy sets indicate the state or condition of the process, and the fuzzy set of the output variable indicates the control action for the process. Generally, fuzzy rules of the Mamdani type have the structure shown in the Eq. (2), which are the most used for the simulation and implementation of a fuzzy controller. The knowledge of a person can be used, or the simulation of the process can be carried out to determine the fuzzy rules. The inference stage determines the fuzzy sets that will be used in the defuzzification stage. Finally, **Table 1** shows the fuzzy rules that relate the membership values of the fuzzification stage used to determine the inference matrix and with it, the value of the tip [24, 25].

$$\text{IF } x \text{ is } A \text{ AND } y \text{ is } B \text{ THEN } z \text{ is } C \quad (2)$$

Food	Service				
	VB	BD	RG	GD	VG
VB	VB	VB	BD	BD	RG
BD	VB	BD	BD	RG	RG
RG	BD	BD	RG	RG	GD
GD	BD	RG	RG	GD	GD
VG	RG	RG	GD	GD	EX

**Table 1.** Knowledge base or inference matrix of the fuzzy rules to determine the tip.

Where:  $x$  and  $y$  are the input variables,  $z$  is the output variable,  $A$  and  $B$  are fuzzy sets of the input variables, and  $C$  is a fuzzy set of the output variable.

The algorithm used to implement fuzzy rules in MATLAB™ Script and Arduino UNO, Arduino DUE, and Nexys 4™ boards is described below.

---

Eq. (2) is used to declare fuzzy rules. Setfd and Setsv were defined in the fuzzification stage, and Settp contains the result of the fuzzy rule (output fuzzy set).

---

% MATLAB Script	// Arduino UNO / DUE	– Nexys 4
if Setfd==VB && Setsv==VB Settp=VB; end	if (Setfd==VB && Setsv==VB) { Settp = VB; }	if (Setfd=VB and Setsv=VB) then Settp := VG; end if;
if Setfd==VG && Setsv==VG Settp =EX; end	if (Setfd==VG && Setsv==VG) { Settp = EX; }	if (Setfd=VG and Setsv=VG) then Settp := EX; end if;

---

Also, the inference stage must define a membership function  $\mu(x)$  for the output fuzzy set or result of a fuzzy rule. Generally, the inference method of the Mamdani type (min-max) is used for the implementation of a fuzzy controller, which uses Eq. (3), which selects the membership function of the input variable with the minimum value. In this case, the IF-THEN control statement is used to determine the membership function with the minimum value, which reduces the computational load and allows the inference stage to be implemented on different platforms that cannot use the min (a, b) operation. Finally, there are fuzzy rules that have the same result (output fuzzy set), which implies that multiple membership functions can be associated with an output fuzzy set.

$$\mu_C(x) = \min\{\mu_A(x), \mu_B(x), \dots, \mu_M(x)\} \quad (3)$$

where:  $\mu_A(x)$ ,  $\mu_B(x)$ , ...,  $\mu_M(x)$  are the membership functions of the input variables, and  $\mu_C(x)$  is the membership function for a fuzzy set of the output variable.

The algorithm used to implement the inference method in MATLAB™ Script and Arduino UNO, Arduino DUE, and Nexys 4™ boards is described below.

---

MfOut is the membership function of the output fuzzy set (result of a fuzzy rule), and the variables “Mf1” and “Mf2” were defined in the fuzzification stage.

---

% MATLAB Script	// Arduino UNO / DUE	– Nexys 4
if Mf1 > Mf2 MfOut = Mf2; % Mf2 < Mf1 else MfOut = Mf1; % Mf1 < Mf2 end	if (Mf1 > Mf2) { MfOut = Mf2; // mf2 < mf1 else MfOut = Mf1; // mf1 < mf2 }	if (Mf1 > Mf2) then MfOut := Mf2; – mf2 < mf1 else MfOut := Mf1; – mf1 < mf2 end if;

---

## 2.5 Aggregation stage

The aggregation stage is used to determine and obtain the membership functions  $\mu(x)$  of the output fuzzy sets, which will be used in the defuzzification stage. As mentioned above, the inference stage can associate multiple membership functions to an output fuzzy set. Therefore, the aggregation stage uses Eq. (4) to select the highest value of the membership function of the aggregation stage. In this case, IF-THEN conditional statements were used to compare the multiple



membership functions and obtain the highest value of the membership function. This option reduces the computational load and allows the aggregation stage to be implemented on different platforms, which cannot use the *max* (a, b) operation [26, 27].

$$\mu_C(x) = \max(\mu_{C_1}(x), \mu_{C_2}(x), \dots, \mu_{C_M}(x)) \quad (4)$$

where:  $\mu_C(x)$  is the membership function of output fuzzy set, which will be used for defuzzification, and  $\mu_{C_1}(x), \mu_{C_2}(x), \dots, \mu_{C_M}(x)$  are the membership functions defined for the output fuzzy set.

The algorithm used to implement the aggregation stage in the MATLAB™ Script and on the Arduino UNO, Arduino DUE, and Nexys 4™ boards is described below.

---

Mftip is the membership function of the output fuzzy set, which will be used in the defuzzification, and MfOut1, ..., MfOut3 are membership functions, which were defined for an output fuzzy set.

---

<i>% MATLAB Script</i>	<i>// ArduinoUNO / DUE</i>	<i>- Nexys 4</i>
if MfOut1 >= MfOut2 Mftip = MfOut1; else Mftip = MfOut2; end	if (MfOut1 >= MfOut2) { Mftip = MfOut1; else Mftip = MfOut2; }	if (MfOut1 >= MfOut2) then Mftip := MfOut1; else Mftip := MfOut2; end if;
if Mftip >= MfOut3 Mftip = Mftip; else Mftip = MfOut3; End	if (Mftip >= MfOut3) { Mftip = Mftip; else Mftip = MfOut3; }	if (Mftip >= MfOut3) then Mftip = Mftip; else Mftip = MfOut3; end if;

---

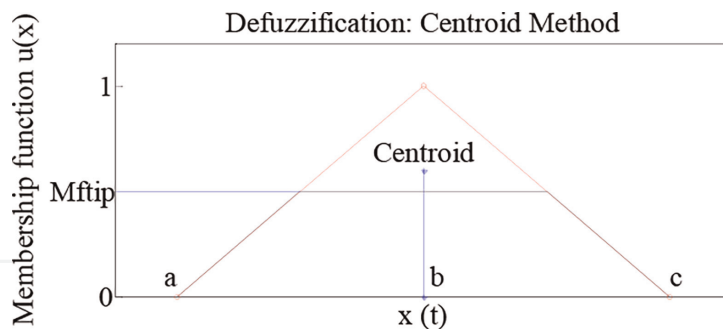
## 2.6 Defuzzification stage

The last stage of the fuzzy controller is defuzzification, which is used to determine a numerical value, which represents the output fuzzy sets. In this work, the defuzzification value is determined using Eq. (5), which represents the centroid method. The centroid method is used, since this method only requires the basic operations of addition, subtraction, multiplication, and division for its implementation in software or hardware. The defuzzification value represents the rigid value of the output controller, in this case, the value of the tip of the food establishment. Finally, the defuzzification value represents the control action that must be performed in a process [20, 28, 29].

$$\text{defuzzification} = (x_1 * \mu(x_1) + x_2 * \mu(x_2) + \dots + x_n * \mu(x_n)) / (\mu(x_1) + \mu(x_2) + \dots + \mu(x_n)) \quad (5)$$

where:  $x_1, x_2, \dots, x_n$  are values of the output variable (“tip”), which are found within the fuzzy sets obtained for defuzzification, and  $\mu(x_1), \mu(x_2), \dots, \mu(x_n)$  are the membership functions of  $x_1, x_2, \dots, x_n$ .

The algorithm used to implement the defuzzification stage in the MATLAB Script and on the Arduino UNO, Arduino DUE, and Nexys 4™ boards is described below.



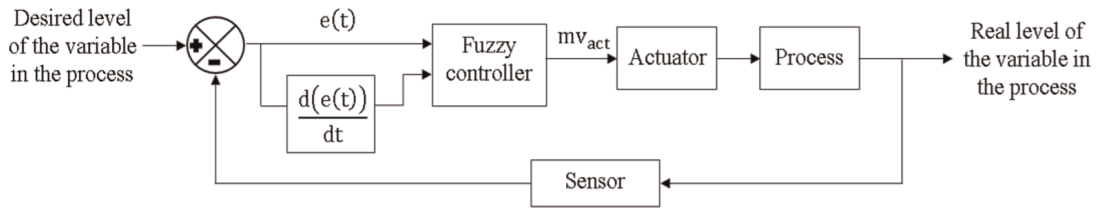
**Figure 5.**  
 Defuzzification using the centroid method.

**Figure 5** shows the fuzzy set used to show the defuzzification process. Eq. (5) is used to determine the defuzzification value.  $M_{ftip}$  is the membership function defined in the aggregation stage,  $m_{fx}$  is the membership function of  $x$ ,  $num1$  and  $den1$  are the numerator and denominator of Eq. (5), respectively.

% MATLAB Script	// Arduino UNO / DUE	- Nexys 4™
$T_{paso} = (c - a) / 1000; x = a;$	$T_{paso} = (c - a) / 1000; x = a;$	$T_{paso} = (c - a) / 10; x = a;$
for $k=1:1:1000$	for ( $k=1; k \leq 1000; k++$ ) {	for $i$ in 0 to 10 loop
$x = x + T_{paso};$	$x = x + T_{paso};$	if ( $x \geq a$ and $x \leq b$ ) then
if $x >= a$ && $x <= b$	if ( $x >= a$ && $x <= b$ ) {	if ( $x \geq a$ and $x \leq b$ ) then
$m_{fx} = (x - a) / (b - a);$	$m_{fx} = (x - a) / (b - a);$	$m_{ux} := ((x-a)*(100))/(b-a);$
if $m_{fx} \geq M_{ftip}$	if ( $m_{fx} \geq M_{ftip}$ ) {	if ( $m_{fx} \geq M_{ftip}$ ) then
$m_{fx} = M_{ftip};$	$m_{fx} = M_{ftip};$	$m_{fx} := M_{ftip};$
end end	} }	end if; end if;
if $x > b$ && $x <= c$	if ( $x > b$ && $x <= c$ ) {	if ( $x > b$ and $x \leq c$ ) then
$m_{fx} = (c - x) / (c - b);$	$m_{fx} = (c - x) / (c - b);$	$m_{ux} := ((c-x)*(100))/(c-b);$
if $m_{fx} \geq M_{ftip}$	if ( $m_{fx} \geq M_{ftip}$ ) {	if ( $m_{fx} \geq M_{ftip}$ ) then
$m_{ux} = M_{ftip};$	$m_{ux} = M_{ftip};$	$m_{fx} := M_{ftip};$
end end	} }	end if; end if;
$num1 = num1 + (x * m_{fx});$	$num1 = num1 + (x * m_{fx});$	$num1 := num1 + (x * m_{fx});$
$den1 = den1 + m_{fx};$ end	$den1 = den1 + m_{fx};$ }	$den1 := den1 + m_{fx};$
% Defuzzification result	// Defuzzification result	- Defuzzification result
$dzz = num1 / den1;$	$dzz = num1 / den1;$	$dzz := num1 / den1;$

## 2.7 Procedure for the implementation of a fuzzy controller in the process

As mentioned above, a fuzzy controller is used to control some variables of a process; therefore, the controlled variable, the type of sensor to measure the controlled variable, and the type of actuator (DC motor, stepper motor, fan, heater, etc.) required to adjust the process must be defined. Additionally, the resolution of the sensor (8 bits, 10 bits, 12 bits, etc.), the sampling rate of the sensor, the sensor operating range, and the type of signal (the number of steps of a stepper motor, minimum and maximum speed of a fan, PWM signal, etc.) required to control the movement of the actuator must be determined to control a process using a fuzzy controller. In this case, the sensor operating range is used to define the characteristics of the input variables of the fuzzy controller and the working range of the actuator is used to define the characteristics of the output variable of the controller [30]. **Figure 6** shows a block diagram of a control system for a process, which uses a generic fuzzy controller. The controller uses the error  $e(t)$  and the derivative of the error  $d(e(t))/dt$  as input variables, the output variable is a control signal for the actuator ( $mv_{act}$ ). Finally, a fuzzy controller does not allow an overshoot to be generated in the system

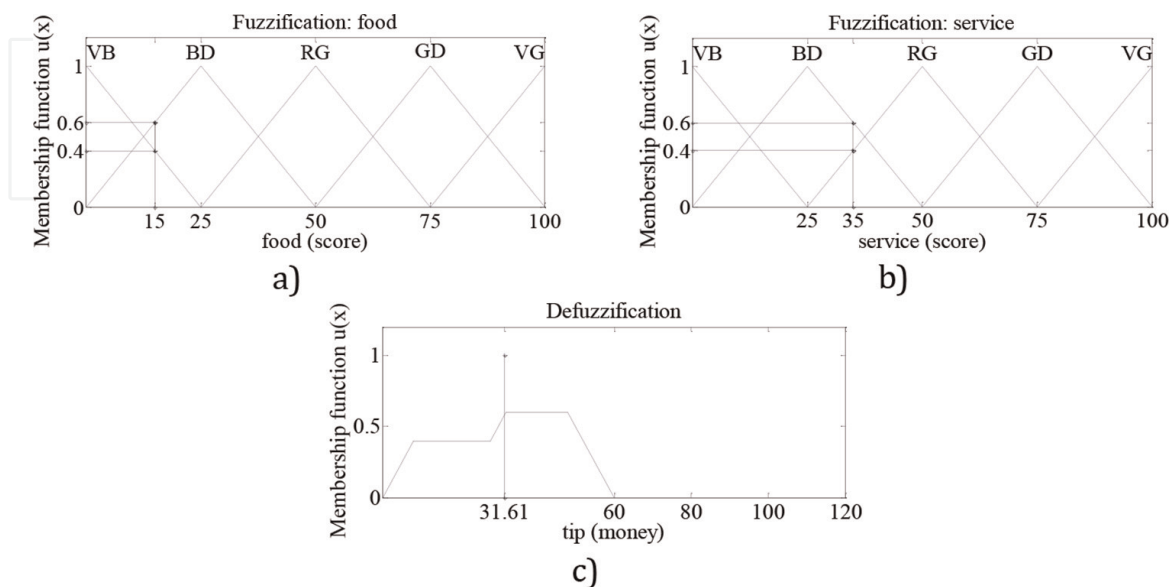


**Figure 6.**  
Block diagram of the control of a process using a fuzzy controller.

response like a classical controller. This is because a fuzzy controller relates the input variables to the output variable through fuzzy rules of the IF-THEN type. So the output response does not have oscillations in the output, and only the settling time and rise time of the system response are short.

### 3. Experimental and simulation results

The analysis, design, and simulation of the programming algorithm for the fuzzy controller were carried out in a MATLAB™ Script, which was used to perform an analysis of its operation. **Figure 7** shows the simulation of the fuzzy controller in the MATLAB™ Script, **Figure 7(a)** shows the fuzzification of the “food” variable with a score of 15, **Figure 7(b)** shows the fuzzification of the “service” variable with a score of 35, and **Figure 7(c)** shows a defuzzification value or tip value of \$31.61. Fuzzy Logic Toolbox™ was used to determine the accuracy of the fuzzy controller, which was implemented using the proposed methodology in the different platforms mentioned above. Therefore, the results of Fuzzy Logic Toolbox™ are considered as the ideal results or correct results. The results of the fuzzy controller in the MATLAB™ Script (MS), Fuzzy Logic Toolbox™ (FLT), the Arduino UNO board (AUNO), the Arduino DUE board (ADUE), and the Nexys 4™ board (NX4) are shown in **Table 2**. The Mean Square Error (MSE) is used to as an analysis of the accuracy of the proposed



**Figure 7.**  
Result of (a) fuzzification of the food variable, (b) fuzzification of the service variable, and (c) defuzzification value or tip value.

Input variables			Defuzzification (tip)			Mean Square Error (MSE)		
Food	Service	FLT	MS	AUNO/ ADUE	NX4	MS	AUNO/ADUE	NX4
0	0	20.0	20.0000	20.00	20	0.0000	0.0000	0.00
10	20	28.4	28.3873	28.38	28	0.0000	0.0002	0.08
20	5	24.8	24.8278	24.83	25	0.0003	0.0004	0.02
30	40	45.4	45.3849	45.38	45	0.0001	0.0002	0.08
40	80	65.4	65.3849	65.38	65	0.0001	0.0002	0.08
50	25	40.0	40.0000	40.00	40	0.0000	0.0000	0.00
60	5	45.4	45.3849	45.38	45	0.0001	0.0002	0.08
70	95	75.2	75.1726	75.17	75	0.0003	0.0004	0.02
80	10	48.4	48.3873	48.39	48	0.0000	0.0000	0.08
90	45	71.6	71.6131	71.61	72	0.0000	0.0000	0.08
100	100	100.0	100.0000	100.00	100	0.0000	0.0000	0.00

**Table 2.**  
 Experimental results and simulation results of fuzzy controller.

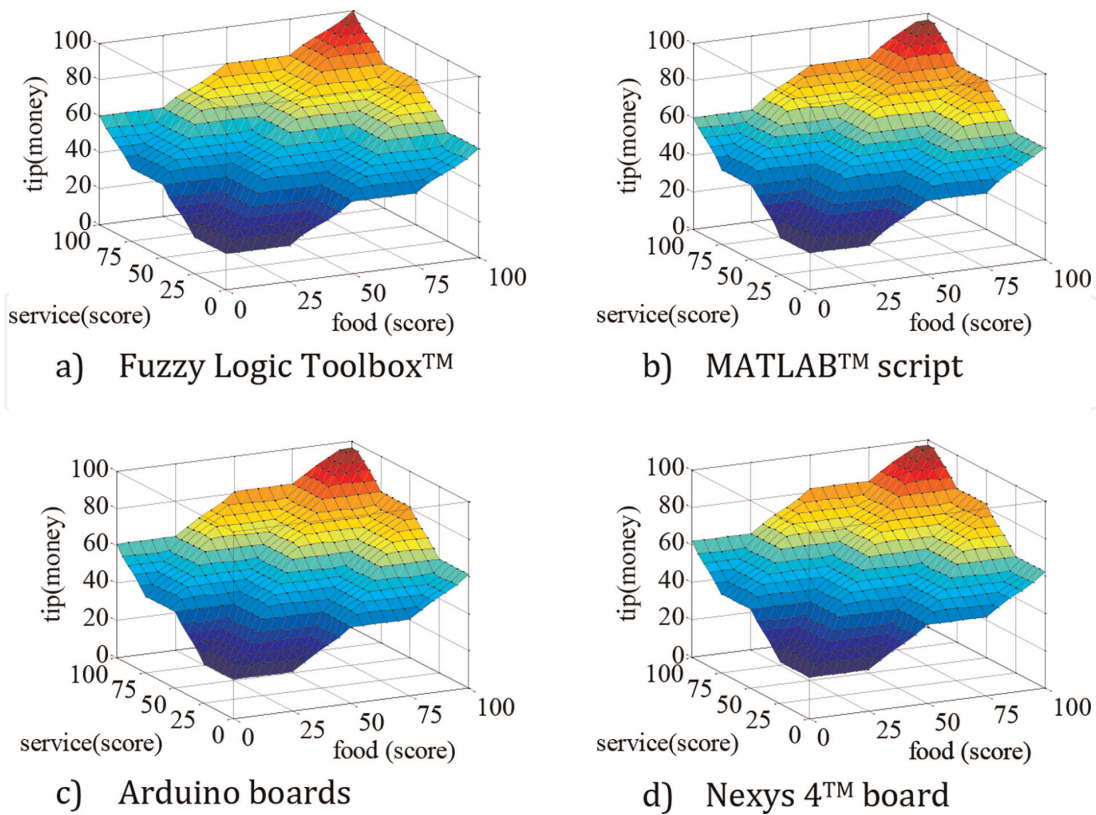
methodology, which is summarized in **Table 2**. It can be seen that the fuzzy controller implemented in the Arduino boards, Nexys 4™, and the MATLAB™ Script generate almost the same MSE in all the examples. Also, the results show that the fuzzy controller can estimate the optimal parameters and compensate the uncertainties and nonlinearity of the system. As we can see, the error is minimal in the different platforms. This is very important for the research presented here, which focuses on the retention of experience and its subsequent instead of the mathematical model and nonlinearities found in the system.

The MSE, which measures the error between two datasets, was used to determine the accuracy of the proposed fuzzy controller. Eq. (6) was used to determine the MSE, and the values of 0.0326, 0.0643, and 0.1125 were obtained for the Arduino UNO, Arduino DUE, and Nexys 4™ boards, respectively. A high degree of accuracy is obtained on the MATLAB™ and Arduino platforms, since their programming language allows the use of a wide variety of variable types (integers, floating point, bit, byte, etc.) and control statements (IF-THEN, FOR, WHILE, etc.). Finally, the precision of the Nexys 4™ board is lower, since this board does not allow the use of floating-point numbers or real numbers.

$$MSE = \frac{1}{n} \sum_{i=1}^n (dn_i - y_i)^2 \quad (6)$$

where  $n$  is the data number,  $dn_n$  is the desired value, and  $y_n$  is the system result.

An analysis of the control surface or controllability of the process is carried out, which is obtained from the fuzzy controller results. The control surface shows the form, in which the process control will be performed. The control surface shows the mapping of the input and output variables of the controller. The control surfaces for the MATLAB™ Script, Fuzzy Logic Toolbox™, Arduino UNO board, Arduino DUE board, and Nexys 4™ board are shown in **Figure 8**. We can see when achieving a monotonic curve as **Figure 8** has, means that do not exist large changes and for this,



**Figure 8.** Control surface of (a) Fuzzy Logic Toolbox™, (b) MATLAB™ script, (c) Arduino boards, and (d) Nexys 4™ board.

control action is smooth that means that the movement of the actuators works without stress, since an unsmooth control action can damage the actuators in the process. The control surface shows all the possible results of the controller; that is, the control surface shows all the results of the variable “tip” for all possible combinations of the variables “food” and “service.”

### 3.1 Characteristics of the implementation of the fuzzy controller

The amount of memory and the processing time are the most important aspects when implementing a programming algorithm on a hardware board. Therefore, the amount of memory and the processing time of the fuzzy controller should be analyzed. The fuzzy controller, which was implemented on the Arduino UNO board, has a processing time of 117 ms and uses 40% of the board’s memory. The fuzzy controller, which was implemented on the Arduino DUE board, has a processing time of 21.275 ms and uses 5% of the board’s memory. The fuzzy controller, which was implemented on the Nexys 4™ board, has a processing time of 17.871 milliseconds and uses 40% of the board’s memory.

### 3.2 Comparison between the methodology fuzzy controller proposed with other works

As a comparison, the analysis of the accuracy of the fuzzy controller proposed with different research works was taken into account, in which a fuzzy controller was



used. In this case, Refs. [31, 32] was used to compare the efficiency. Arduino Mega 2560 board and the Arduino UNO board are compared. **Tables 3-5** show the results of the fuzzy controller and the MSE of the data. As can be seen, the fuzzy controller using the proposed methodology has a higher precision than the fuzzy controller taking in consideration.

On the other hand, there are research works that use a computer to implement a fuzzy controller and a hardware board as an interface to interconnect the environment (physical variables) with the computer [33, 34]. This action can increase the cost of the system and make it difficult to implement it in a process. **Table 6** shows the results of a fuzzy controller, which is implemented in MATLAB™ and used for the prediction of GSM tissue and wrinkle recovery angle of laser-engraved denim [35]. Additionally, the results of the fuzzy controller implemented using the proposed methodology and the MSE are shown. As can be seen, a high degree of accuracy can be obtained in a process using the methodology proposed in this work.

Light intensity	Brightness (LED)			Mean Square Error	
	Fuzzy Logic Toolbox™	Fuzzy controller of the Greenhouse (Arduino Mega)	Proposed fuzzy controller (Arduino uno)	Arduino MEGA	Arduino UNO
523	450	450	450.00	0.0000	0.0000
498	455	455.85	454.93	0.3612	0.0024
78	850	850	850.00	0.0000	0.0000
57	850	850	850.00	0.0000	0.0000
218	810	808.5	809.60	1.1250	0.0800
213	820	818.4	820.00	1.2800	0.0000
693	267	269.2	266.70	2.4200	0.0450
689	276	278.9	275.64	4.2050	0.0648
801	250	250	250.00	0.0000	0.0000
688	278	281.2	277.80	5.1200	0.0200

**Table 3.**  
 Test of the light intensity of an LED using a diffuse driver.

PH	Moisture (%)	Experimental results of the process	Fuzzy controller for fertilization (MATLAB)	Proposed fuzzy controller (Arduino uno)	Mean Square Error	
					MATLAB	Arduino UNO
5.3	43	0 ml	0 ml	0 ml	0.0000	0.0000
6.89	71	18 ml	18.35 ml	18.37	0.0612	0.0684
4.01	57	0 ml	0 ml	0 ml	0.0000	0.0000
7.15	59	23 ml	23.23 ml	23.31 ml	0.0264	0.0480

**Table 4.**  
 Results of the acid solution for the process using a fuzzy controller.

PH	Moisture (%)	Experimental results of the process	Fuzzy controller for fertilization (MATLAB)	Proposed fuzzy controller (Arduino uno)	Mean Square Error	
					MATLAB	Arduino UNO
5.3	43	55 ml	54.68 ml	54.53 ml	0.0512	0.1104
6.89	71	0 ml	0 ml	0 ml	0.0000	0.0000
4.01	57	25 ml	25.49 ml	25.64 ml	0.1200	0.2048
7.15	59	0 ml	0 ml	0 ml	0.0000	0.0000

**Table 5.** Results of the neutral solution for the process using a fuzzy controller.

Dots Per Inch	Pixel Time	Fabric weight (GSM)	Fuzzy controller to predict the GSM	Proposed fuzzy controller (Arduino uno)	Recovery Angle (Degree)	Fuzzy controller to predict the angle of recovery (Degree)	Proposed fuzzy controller (Arduino uno)
15	100	439	429	428.5105	77	75	75.00
15	150	411	409	408.8334	71	70.8	70.83
15	200	385	379	378.3335	66	64.5	64.54
20	100	400	409	408.8334	69	70.8	70.83
20	150	375	379	378.3335	65	64.5	64.54
20	200	354	348	348.1668	60	58.3	58.38
25	100	372	379	378.3335	62	64.5	64.54
25	150	345	348	348.1668	58	58.3	58.38
25	200	318	328	328.1575	52	54	54.23

**Table 6.** Fuzzy controller to predict the strength values of laser-engraved denim seams.

#### 4. Conclusions

In this work, a methodology was proposed to implement a fuzzy controller of the Mamdani type on different platforms (software or hardware). The methodology is based on the IF-THEN and FOR control statements, and mathematical operations such as addition, subtraction, multiplication, and division, which are used to establish each of the stages of a fuzzy controller (fuzzification, fuzzy rules, or defuzzification). The key ideas explored are the use of programming languages such as VHDL, C++, and MATLAB™. The operation of the fuzzy controller based on the proposed methodology is tested using different examples, which are compared with the Fuzzy Logic Toolbox™ (MATLAB™ tool). The results show that the fuzzy controller methodology can improve the convergence of the system and improve the error properties. In addition, the proposed methodology reduces the computational difficulty and the computational load. Also, the procedure to implement the fuzzy controller in a process is described. Compared to other works, the greatest contribution of this work is to describe the elements to program and implement each of the stages of a fuzzy controller in software or hardware. This will allow estimating the optimal parameters,

uncertainties, and nonlinearities of a dynamic system to control a system without the mathematical model. Finally, simulations were shown to verify that the proposed methodology, and the theoretical and experimental results are valid.

## **Acknowledgements**

This study was funded by Consejo Mexiquense de Ciencia y Tecnologia (COMECYT).


## **Author details**

Jesus de la Cruz-Alejo\*, Hugo Beatriz-Cuellar, Agustin Mora-Ortega  
and Maria Belem Arce-Vazquez  
Tecnológico de Estudios Superiores de Ecatepec, Estado de Mexico, Mexico

\*Address all correspondence to: [jdelacruz@tese.edu.mx](mailto:jdelacruz@tese.edu.mx)

## **IntechOpen**

---

© 2023 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## References

- [1] Zou Y, Yan F, Wang X, Zhang J. An efficient fuzzy logic control algorithm for photovoltaic maximum power point tracking under partial shading condition. *Journal of the Franklin Institute*. 2020; **357**(6):3135-3149. DOI: 10.1016/j.jfranklin.2019.07.015
- [2] Subramanian V, Indragandhi V, Kuppusamy R, Teekaraman Y. Modeling and analysis of PV system with fuzzy logic MPPT technique for a DC microgrid under variable atmospheric conditions. *Electronics (Basel)*. 2021; **10**(20):2541. DOI: 10.3390/electronics10202541
- [3] Hassan T-U, Abbassi R, Jerbi H, Mehmood K, Tahir MF, Cheema KM, et al. A novel algorithm for MPPT of an isolated PV system using push pull converter with fuzzy logic controller. *Energies*. 2020; **13**(15):4007. DOI: 10.3390/en13154007
- [4] Farajdadian S, Hosseini SMH. Design of an optimal fuzzy controller to obtain maximum power in solar power generation system. *Solar Energy*. 2019; **182**:161-178. DOI: 10.1016/j.solener.2019.02.051
- [5] Loukil K, Abbes H, Abid H, Abid M, Toumi A. Design and implementation of reconfigurable MPPT fuzzy controller for photovoltaic systems. *Ain Shams Engineering Journal*. 2020; **11**(2): 319-328. DOI: 10.1016/j.asej.2019.10.002
- [6] Muñoz Villacorta R, Oscco Agüero C, Andrade-Arenas L. Implementation of an intelligent system for the diagnosis and treatment of venereal diseases. *International Journal of Online Engineering*. 2022; **18**(11):58-76. DOI: 10.3991/ijoe.v18i11.32329
- [7] Aamir KM, Sarfraz L, Ramzan M, Bilal M, Shafi J, Attique M. A fuzzy rule-based system for classification of diabetes. *Sensors (Basel)*. 2021; **21**(23): 8095. DOI: 10.3390/s21238095
- [8] Zulkarnain AF, Sari Y, Rakhmadani R. Monitoring system for early detection of fire in wetlands based internet of things (IoT) using fuzzy methods. *IOP Conference Series: Materials Science and Engineering*. 2021; **1115**(1):012007. DOI: 10.1088/1757-899x/1115/1/012007
- [9] Soliman MA, Hasanien HM, Azazi HZ, El-Kholy EE, Mahmoud SA. An adaptive fuzzy logic control strategy for performance enhancement of a grid-connected PMSG-based wind turbine. *IEEE Transactions on Industrial Informatics*. 2019; **15**(6):3163-3173. DOI: 10.1109/tii.2018.2875922
- [10] Ganthia BP, Barik SK. Fault analysis of PI and fuzzy-logic-controlled DFIG-based grid-connected wind energy conversion system. *Journal of The Institution of Engineers (India): Series B*. 2022; **103**(2):415-437. DOI: 10.1007/s40031-021-00664-9
- [11] Ngo Q-V, Yi C, Nguyen T-T. The maximum power point tracking based-control system for small-scale wind turbine using fuzzy logic. *International Journal of Electrical and Computer Engineering (IJECE)*. 2020; **10**(4):3927. DOI: 10.11591/ijece.v10i4.pp3927-3935
- [12] Murillo-Yarce D, Munoz J, Restrepo C. Mamdani type PI-fuzzy controller in a boost converter. *IEEE International Conference on Industrial Technology (ICIT)*. 2020:487-492. DOI: 10.1109/ICIT45562.2020.9067257
- [13] Riahi J, Vergura S, Mezghani D, Mami A. Intelligent control of the

- microclimate of an agricultural greenhouse powered by a supporting PV system. *Applied Sciences (Basel)*. 2020;**10**(4):1350. DOI: 10.3390/app10041350
- [14] Sarmiento G. *Arduino Curso Práctico: Manual Práctico*. North Charleston, SC, United States of America: Createspace Independent Publishing Platform; 2015
- [15] McRoberts M. *Beginning Arduino*. 2nd ed. New York, NY, United States of America: Apress; 2013
- [16] Maxinez DG, Alcalá Jara J. *VHDL. El arte de programar sistemas digitales*. 1st ed. CECSA; 2002. p. 368
- [17] Nexys 4™ FPGA Board Reference Manual [Internet]. Digilent.com. 2016. Available from: [https://digilent.com/reference/\\_media/nexys4:nexys4\\_rm.pdf](https://digilent.com/reference/_media/nexys4:nexys4_rm.pdf)
- [18] Kalechman M. *Practical MATLAB Applications for Engineers*. Londres, Inglaterra: CRC Press; 2018
- [19] Knight A. *Basics of MATLAB and Beyond*. 1st ed. Chapman and Hall/CRC; 2019. p. 216. DOI: 10.1201/9780429186882
- [20] Michels K, Klawonn F, Kruse R, Nurnberger A. *Fuzzy Control: Fundamentals, Stability and Design of Fuzzy Controllers*. Berlín, Germany: Springer; 2010
- [21] Blokdyk G. *Fuzzy Logic: Beyond the Basics*. North Charleston, SC, United States of America: Createspace Independent Publishing Platform; 2017
- [22] Santos W, editor. *Fuzzy Control Systems: Design, Analysis & Performance Evaluation*. Nova Science: Hauppauge, NY, United States of America; 2017
- [23] Ebrahimnejad A, Verdegay JL. *Fuzzy Set Theory*. En: *Fuzzy Sets-Based Methods and Techniques for Modern Analytics*. Cham: Springer International Publishing; 2018. pp. 1-27
- [24] *Fuzzy logic projects with matlab*. En: *Introduction to Fuzzy Logic using MATLAB*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007. pp. 369–408
- [25] Zadeh LA, Aliev RA. *Fuzzy Logic Theory and Applications: Part I and Part II: Part I and Part II*. Singapur, Singapur: World Scientific Publishing; 2019
- [26] Ross TJ. *Fuzzy Logic with Engineering Applications*. 4a ed. Nashville, TN, United States of America: John Wiley & Sons; 2016
- [27] Gerla G. *Fuzzy Logic: Mathematical Tools for Approximate Reasoning*. Dordrecht, Netherlands: Springer; 2010
- [28] Kovacic Z, Bogdan S. *Fuzzy Controller Design: Theory and Applications*. 1st ed. CRC Press; 2019. p. 416. DOI: 10.1201/9781420026504
- [29] Pourabdollah A. *Fuzzy Number Value or Defuzzified Value; Which One Does It Better?*. *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 2020:1-6. DOI: 10.1109/FUZZ48607.2020.9177533
- [30] Rodriguez Lopez ML. *Tendencias en Instrumentación y Control de Procesos*. Encuentro Sennova del Oriente Antioqueño. 2021. DOI: 10.23850/22565035.2953
- [31] Kurniawan D, Witanti A. *Prototype of Control and Monitor System with Fuzzy Logic Method for Smart Greenhouse*. *Indonesian Journal of Information Systems*. 2021;**3**(2):116-127. DOI: 10.24002/ijis.v3i2.4067



[32] Penzol N, Adnan R. Design of an Internet of Things (Iot) Based Smart Irrigation and Fertilization System Using Fuzzy Logic for Chili Plant. Shah Alam, Malaysia: IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS 2020); 2020. DOI: 10.1109/I2CACIS49202.2020.9140199

[33] Hassan T, Abbassi R, Jerbi H, Mehmood K, Tahir MF, Cheema KM. A Novel Algorithm for MPPT of an Isolated PV System Using Push Pull Converter with Fuzzy Logic Controller. *Energies*. 2020;**13**(15):4007. DOI: 10.3390/en13154007

[34] Dehghani M, Taghipour M, B. Gharehpetian G, Abedi M. Optimized Fuzzy Controller for MPPT of Grid-connected PV Systems in Rapidly Changing Atmospheric Conditions. *Journal of Modern Power Systems and Clean Energy*. 2021;**9**(2):376-383. DOI: 10.35833/MPCE.2019.000086

[35] Sarkar J, Mondal M, Khalil E. Predicting fabric GSM and crease recovery angle of laser engraved denim by fuzzy logic analysis. *Journal of Engineering and Applied Science*. 2020; **4**(1):52-64