Control of the humidity percentage of a bioreactor using a fuzzy controller to grow bonsai

Jesus de la Cruz-Alejo, Hugo Beatriz Cuellar, Irving Cardiel Alcocer Guillermo
Division de ingenieria Mecanica, Mecatronica e Industrial, Tecnologico de Estudios Superiores de Ecatepec, Ecatepec de Morelos, México

Article Info

Article history:

Received May 6, 2021
Revised Sep 29, 2021
Accepted Nov 5, 2021

Keywords:

Arduino Uno
Bioreactor
Design
Fuzzy controller
Implementation
MATLAB™

ABSTRACT

Different controllers have been designed and used to cultivate bonsai, which need specific conditions to grow and survive in a different place or climate, for this case, humidity. In this work, theoretical, simulation and experimental level are compared and presented in terms of performance characteristics such as complexity, accuracy and convergence of an algorithm proposed to design and implement a fuzzy controller used in a bioreactor to control the humidity percentage to grow bonsai. The MATLAB™ script and fuzzy logic Toolbox™ were used for the analysis and simulation. The controller implementation was done on an Arduino Uno board, and 25850 bytes or 80% of the memory were used to implement it. A sensor to monitor the humidity percentage, a stepper motor connected to a water tap, and a DC motor connected to a propeller were used to adjust the humidity percentage of the bioreactor. The controller results show a maximum error of ±1% for all entire range, and a processing time of 5 milliseconds for one iteration. The results of the tests carried out in the bioreactor are in accordance with the predictions and theoretical simulations, which presents a maximum error of 3%, and a convergence time of 50 seconds for the worst case.

Corresponding Author:

Jesus de la Cruz Alejo
Division de ingenieria Mecanica, Mecatronica e Industrial, Tecnologico de Estudios Superiores de Ecatepec
Av. Tecnológico S/N C.P. 55210 Col. Valle de Anáhuac, Ecatepec de Morelos, Estado de México
Email: jdelacruz@tese.edu.mx

1. INTRODUCTION

Bioreactors are vessels used to provide a controlled environment, and allow efficient growth and formation of a product, that is, bioreactors produce the conditions necessary for a biological system. Generally, the main variables controlled in a bioreactor are pH, temperature, humidity, oxygen distribution, concentration of salts, among others. Also, bioreactors are used to produce enzymes, antibodies, treatment of polluted air, wastewater treatment, and cultivation of some type of plant [1]. For this reason, a control system is needed, which provides the necessary conditions to ensure the growth and development of the product within the bioreactor. Currently, there is a great variety of control systems, which can be used to control a process, for example, artificial neural networks, proportional integral derivative (PID) control, robust control, sliding modes, programmable logic controllers (PLC), fuzzy logic controller (FLC), among others. The necessary requirements for the implementation of a control system in a process must be analyzed (for example, mathematical model of the system or control system cost). Also, the difficulty of the implementation of a control system is directly related to the difficulty of the process. Fuzzy controllers have been used for many types of processes, for example, reducing the degree of difficulty and increasing the precision in determining the degree of diabetes [2], trajectory tracking and vibration control of a flexible joint manipulator [3], improvement of the performance of a wind generator system connected to the grid [4], [5], modeling and simulation of the monitoring of the maximum power point used in photovoltaic systems [6], [7], operation of a photovoltaic panel at the maximum power point to increase efficiency to provide the proper current and voltage to charge a battery [8], prediction of vehicle crash rate based on traffic level, speed limit and surface friction [9], among others. In this paper, a FLC is used to control the humidity percentage of a bioreactor, which is used to grow bonsai, since some types of bonsai need special conditions for their growth, since they are cultivated in a place or climate different from their place of origin. A FLC does not need the mathematical model of the system since the knowledge of people is used to control the process. It uses linguistic expressions to define the conditions of the process and control action [10]–[13]. Therefore, the design and implementation of this type of controller does not have a high degree of difficulty. The structure of the paper is as follows, section 2 presents the procedure for the design and implementation of the FLC, section 3 presents the experimental and simulation results of the FLC, section 4 presents the characteristics of the implementation of the FLC, the sensor and actuators used in the bioreactor, the design of the bioreactor, and the electrical diagram of the bioreactor, and section 5 presents the conclusions.

2. CONTROLLER DESCRIPTION

In this proposal, a mamdani type FLC is used, which uses the fuzzification, inference, aggregation and defuzzification stages. Figure 1 shows a block diagram of the controller structure [14]. Figure 2 shows a block diagram of the FLC used to adjust the percentage of humidity in the bioreactor. A sensor is used to measure the percentage of humidity in the bonsai pot, and two actuators are used to adjust the percentage of humidity, the first actuator is a stepper motor connected to a water tap, which is used to increase humidity, and the second actuator is a DC motor connected to a propeller, which is used to reduce humidity. This manner, if there is an error regarding the desired humidity percentage, the FLC will use one of the actuators to adjust the humidity percentage. The FLC uses two input variables to perform the humidity control, the first is a signal defined as "*error*", which is obtained from the difference between the desired value and the real value of the humidity percentage at that moment. The second is a signal defined as "*derivative of the error*", which is obtained from the difference between the value of the current error and the value of the previous error. Also, this variable is used to smooth the response of the FLC and not have the system stressed. The value of the variable "*error*" and the variable "*derivative of the error*" are determined using (1) and (2), respectively. The FLC uses the signal defined as "*actuator*" as the output variable. The MATLAB™ script and fuzzy logic toolbox™ were used for the design of the FLC, and the Arduino Uno board was used for its implementation. Finally, 50% (desired humidity percentage) humidity is needed in the bonsai pot for the bonsai to grow properly.

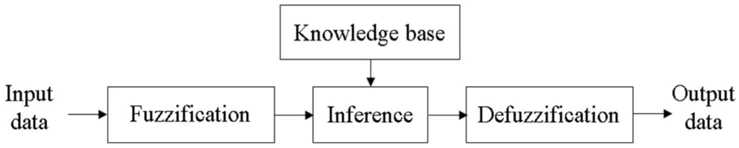


Figure 1. FLC block diagram

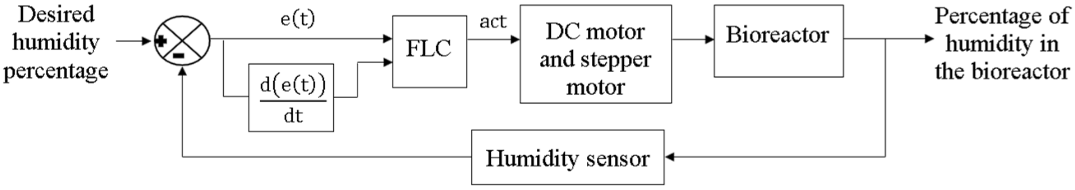

Desired humidity percentage
+
-
e(t)
$\frac{d(e(t))}{dt}$
FLC
act
DC motor and stepper motor
Bioreactor
Percentage of humidity in the bioreactor
Humidity sensor

Figure 2. Bioreactor control system

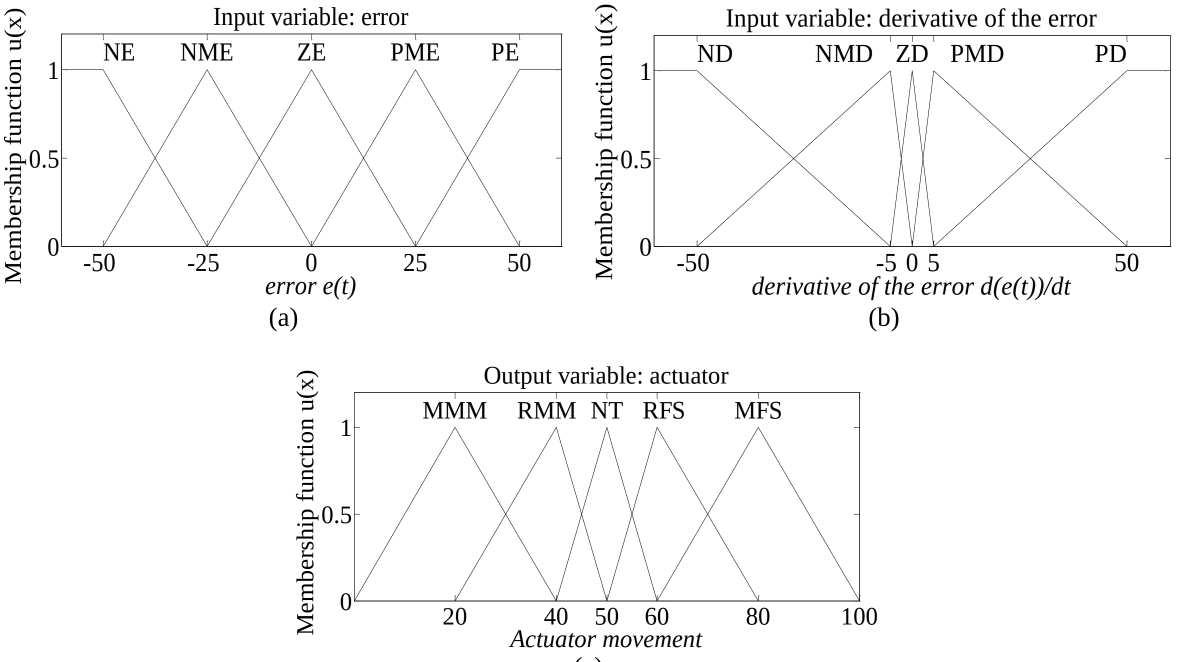
$$e(t) = Hrf - Hbi \tag{1}$$

$$\mathrm{d(e(t))} = \mathrm{e(t)} - \mathrm{e(t-1)} \tag{2}$$

where: *d(e(t))* is the derivative of the error, *e(t)* and *e(t-1)* are the current error and the previous error of the process, respectively. *Hrf* and *Hbi* are the current humidity and the desired humidity, respectively.

2.1. Fuzzy sets for the input and output variables of the FLC

The universes of discourse used for the input and output variables of the FLC are as follows: the variable "*error*" uses a range of values from -50 to 50, the variable "*derivative of the error*" uses a range of values from -50 to 50. These ranges are because the maximum change of the humidity percentage in the bioreactor is 50%. The "*actuator*" variable uses a range of values from 0 to 100. This universe of discourse is used to define the working ranges of the actuators. Additionally, triangular fuzzy sets were used to define the fuzzy sets in the universes of discussion for inputs and output variables. This type of fuzzy set allows to determine the degree of membership in a simple way. To find the number of fuzzy sets for each input variable, an analysis of the variation of the percentage of humidity was carried out, having as a result that the percentage of humidity increases rapidly and decreases slowly. Therefore, five fuzzy sets were proposed for the variable "*error*" labeled as negative error (NE), negative mean error (NME), zero error (ZE), positive mean error (PME) and positive error (PE). Also, five fuzzy sets were proposed for the variable "*derivative of the error*" labeled as negative derivative (ND), negative mean derivative (NMD), zero derivative (ZD), positive mean derivative (PMD) and positive derivative (PD). Finally, five fuzzy sets were proposed for the output variable "*actuator*", labeled as maximum fan speed (MFS), regular fan speed (RFS), nothing (NT), regular movement of the stepper motor (RMM) and maximum movement of the stepper motor (MMM). Additionally, the RMM fuzzy set indicates that the stepper motor will move to the middle of its working range. The MMM fuzzy set indicates that the stepper motor will move to the end of its working range. Figure 3 and Tables 1, 2 and 3 show the dimensions and graphs of the fuzzy sets of the FLC variables. Figure 3(a) shows the fuzzy sets of the variable "*error*", Figure 3(b) shows the fuzzy sets of the variable "*derivative of the error*", and Figure 3(c) shows the fuzzy sets of the variable "*actuator*". The characteristics and the action control of the process determined the shape and dimensions of the fuzzy sets. This manner, the FLC is activated with small changes in the humidity percentage. The MMM and MSF fuzzy sets of the output variable are larger, since, under extreme conditions (0% or 100% humidity), the actuators must operate at their maximum capacity.



(a)

(b)

(c)

Figure 3. Fuzzy sets of the FLC variables (a) error, (b) derivative of the error, and (c) actuator

Table 1. Dimensions of the fuzzy sets of the variable "*error*"

Set: NE			Set: NME			Set: ZE			Set: PME			Set: PE		
a1=-50	b1=-50	c1=-25	a2=-50	b2=-25	c2=0	a3=-25	b3=0	c3=25	a4=0	b4=25	c4=50	a5=25	b5=50	c5=50

Table 2. Dimensions of the sets of the variable "*derivative of the error*"

Set: ND			Set: NMD			Set: ZD			Set: PMD			Set: PD		
a6=-50	b6=-50	c6=-5	a7=-50	b7=-5	c7=0	a8=-5	b8=0	c8=5	a9=0	b9=5	c9=50	a10=5	b10=50	c10=50

Table 3. Dimensions of the sets of the variable "*actuator*"

Set: MMM			Set: RMM			Set: NT			Set: RFS			Set: MFS		
a11=0	b11=20	c11=40	a12=20	b12=40	c12=50	a13=40	b13=50	c13=60	a14=50	b14=60	c14=80	a15=60	b15=80	c15=100

2.2. Fuzzification stage

The first stage of the FLC is the fuzzification, which assigns the membership degree of an input variable to one or more fuzzy sets contained in the universe of discussion for that input variable [15]. To determine the membership function $\mu(x)$, (3) and the data in Figure 4 must be used. The Figure 5 proposed to perform the fuzzification stage in the MATLAB™ script and on the Arduino Uno board.

$$\mu(x) = 0\ si\ x \leq a, (x-a)/(b-a)\ si\ a \leq x \leq b, (c-x)/(c-b)\ si\ b \leq x \leq c, 0\ si\ x \geq c \quad (3)$$

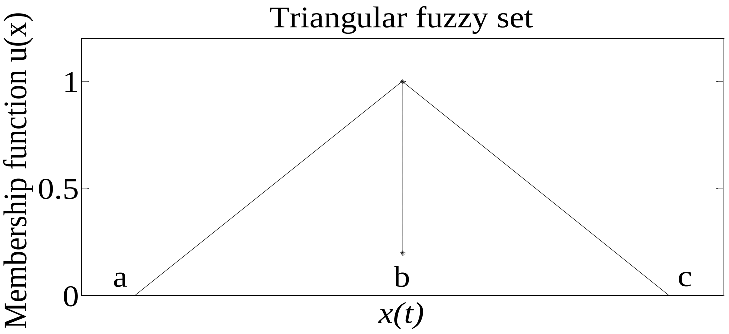
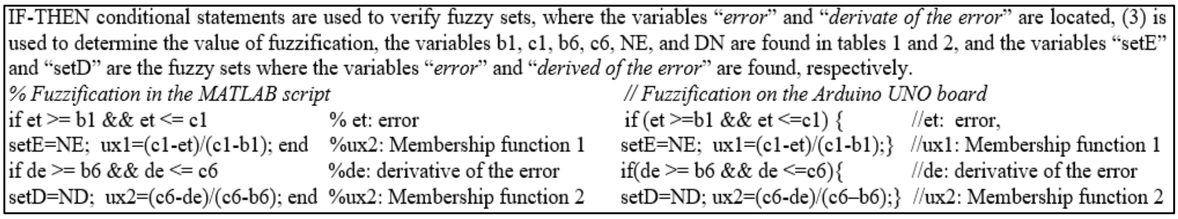


Figure 4. Triangular fuzzy set and its equations to determine the membership function *μ(x)*



IF-THEN conditional statements are used to verify fuzzy sets, where the variables "*error*" and "*derivate of the error*" are located, (3) is used to determine the value of fuzzification, the variables b1, c1, b6, c6, NE, and DN are found in tables 1 and 2, and the variables "setE" and "setD" are the fuzzy sets where the variables "*error*" and "*derived of the error*" are found, respectively.

```
% Fuzzification in the MATLAB script                              // Fuzzification on the Arduino UNO board
if et >= b1 && et <= c1                % et: error                 if (et >=b1 && et <=c1) {              //et:  error,
setE=NE;  ux1=(c1-et)/(c1-b1); end     %ux2: Membership function 1 setE=NE;  ux1=(c1-et)/(c1-b1);}    //ux1: Membership function 1
if de >= b6 && de <= c6                %de: derivative of the error if(de >= b6 && de <=c6){              //de: derivative of the error
setD=ND;  ux2=(c6-de)/(c6-b6); end     %ux2: Membership function 2 setD=ND; ux2=(c6-de)/(c6–b6);}     //ux2: Membership function 2
```

Figure 5. Implement the fuzzification stage

2.3. Inference stage

The inference stage uses a set of fuzzy rules, which relate the fuzzy sets of the input variables (the fuzzy sets of the input variables indicate the state of the process). The result of a fuzzy rule is a fuzzy set of the output variable (the fuzzy set of the output variable indicates the control action for the process). In this work, the fuzzy rules of the Mamdani type are used, which have the structure shown in (4) [16], [17]. The accuracy of the control action depends on the number of fuzzy sets used for the input and output variables, and the number of fuzzy rules depends on the number of fuzzy sets of the input variables. Table 4 shows the fuzzy rules used to control the percentage of humidity. The gray boxes indicate the absence of a fuzzy rule, since a process analysis was performed, and these rules are not needed to control the percentage of humidity in the bioreactor. Additionally, a membership function must be assigned to the output fuzzy set (result of the fuzzy rule). Mamdani inference is used, which is shown in (5) and uses the membership functions from the input fuzzy sets and selects the membership function with the minimum value. Finally, the inference stage determines the fuzzy sets used in the next stages of the FLC.

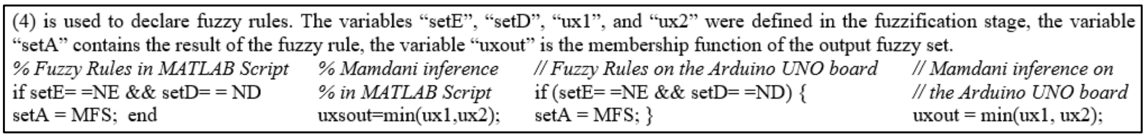
$$IF\ x\ is\ A\ AND\ y\ is\ B\ THEN\ z\ is\ C \quad (4)$$

$$\mu C(x) = \min(\mu A(x), \mu B(x), \dots, \mu M(x)) \quad (5)$$

Where x and y are the input variables, z is the output variable, A and B are the fuzzy sets of the input variables, C is a fuzzy set of the output variable, *μC(x)* is the membership function of the output fuzzy set C, and *μA(x), μM(x)* are the membership functions of input fuzzy sets. The Figure 6 proposed to perform the fuzzy rules and Mamdani type inference in the MATLAB™ Script and on the Arduino Uno board are shown as.

Table 4. Knowledge base or fuzzy rules used to control the bioreactor

		derivative of the error				
error		ND	NMD	ZD	PMD	PD
	NE	MFS	MFS	MFS	MFS	RFS
	NME		RFS	RFS	RFS	
	ZE	NT	NT	NT	NT	NT
	PME		RMM	RMM	RMM	
	PE	RMM	MMM	MMM	MMM	MMM



(4) is used to declare fuzzy rules. The variables “setE”, “setD”, “ux1”, and “ux2” were defined in the fuzzification stage, the variable “setA” contains the result of the fuzzy rule, the variable “uxout” is the membership function of the output fuzzy set.

```
% Fuzzy Rules in MATLAB Script   % Mamdani inference    // Fuzzy Rules on the Arduino UNO board   // Mamdani inference on
if setE= =NE && setD= = ND       % in MATLAB Script     if (setE= =NE && setD= =ND) {             // the Arduino UNO board
setA = MFS;  end                 uxsout=min(ux1,ux2);   setA = MFS; }                             uxout = min(ux1, ux2);
```

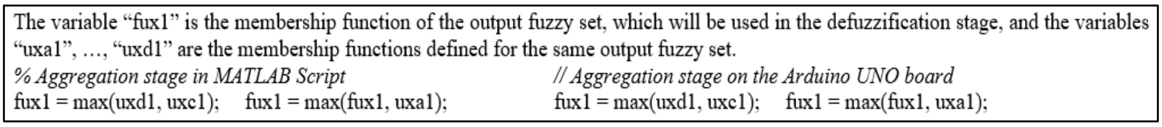
Figure 6. Implement the inference stage

2.4. Aggregation stage

The inference stage can associate multiple membership functions *μ(x)* for the same output fuzzy set. For this reason, the aggregation stage is used, which uses (6) to select the membership function *μ(x)* with the maximum value, which will be the membership function of the output fuzzy set [18], [19].

$$\mu C(x) = max\{\mu C1(x), \mu C2(x), \dots, \mu CM(x)\} \quad (6)$$

Where: *μC(x)* is the maximum membership function of output fuzzy set *C*, which will be used for defuzzification, and *μC1(x), ..., μCM(x)* are the membership functions defined for the output fuzzy set *C*. The Figure 7 proposed to perform the aggregation stage in the MATLAB™ Script and on the Arduino Uno board are shown as.



The variable “fux1” is the membership function of the output fuzzy set, which will be used in the defuzzification stage, and the variables “uxa1”, …, “uxd1” are the membership functions defined for the same output fuzzy set.

```
% Aggregation stage in MATLAB Script                      // Aggregation stage on the Arduino UNO board
fux1 = max(uxd1, uxc1);    fux1 = max(fux1, uxa1);        fux1 = max(uxd1, uxc1);    fux1 = max(fux1, uxa1);
```

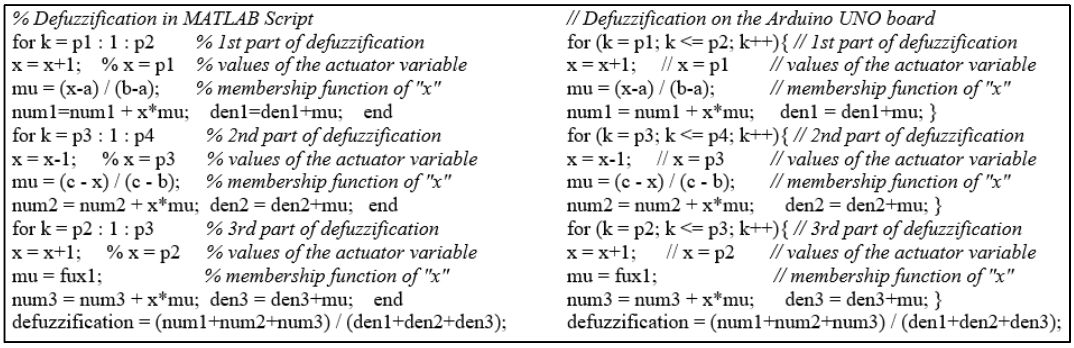
Figure 7. Implement the aggregation stage

2.5. Defuzzification stage

Defuzzification is the last stage of the FLC, this stage is used to associate the output fuzzy sets with a numerical value, which will represent the output of the FLC. This value is used to operate the actuators and adjust the percentage of humidity in the bioreactor. The defuzzification value is determined using (7), which represents the centroid method. Therefore, the variables *x1, ..., xn* represent the values of the output variable, which are found within the output fuzzy sets. The variables *μ(x1), ..., μ(xn)* are the membership functions of the values corresponding to *x1, ..., xn*. The centroid method was used for defuzzification, since the basic operations of addition, subtraction, multiplication, and division are used for its implementation in software or hardware [20].

$$Defuzzification = \frac{x1 * \mu(x1) + x2 * \mu(x2) + \dots + xn * \mu(xn)4}{\mu(x1) + \mu(x2) + \dots + \mu(xn)} \quad (7)$$

The Figure 8 proposed to perform the defuzzification stage in the MATLAB™ script and on the Arduino Uno board. Figure 9 shows the parts of the output fuzzy set.



```
% Defuzzification in MATLAB Script
for k = p1 : 1 : p2       % 1st part of defuzzification
x = x+1;   % x = p1       % values of the actuator variable
mu = (x-a) / (b-a);       % membership function of "x"
num1=num1 + x*mu;   den1=den1+mu;   end
for k = p3 : 1 : p4        % 2nd part of defuzzification
x = x-1;    % x = p3       % values of the actuator variable
mu = (c - x) / (c - b);    % membership function of "x"
num2 = num2 + x*mu;  den2 = den2+mu;   end
for k = p2 : 1 : p3        % 3rd part of defuzzification
x = x+1;    % x = p2       % values of the actuator variable
mu = fux1;                 % membership function of "x"
num3 = num3 + x*mu;  den3 = den3+mu;   end
defuzzification = (num1+num2+num3) / (den1+den2+den3);
```

```
// Defuzzification on the Arduino UNO board
for (k = p1; k <= p2; k++){ // 1st part of defuzzification
x = x+1;    // x = p1       // values of the actuator variable
mu = (x-a) / (b-a);         // membership function of "x"
num1 = num1 + x*mu;     den1 = den1+mu; }
for (k = p3; k <= p4; k++){ // 2nd part of defuzzification
x = x-1;    // x = p3       // values of the actuator variable
mu = (c - x) / (c - b);     // membership function of "x"
num2 = num2 + x*mu;     den2 = den2+mu; }
for (k = p2; k <= p3; k++){ // 3rd part of defuzzification
x = x+1;    // x = p2       // values of the actuator variable
mu = fux1;                  // membership function of "x"
num3 = num3 + x*mu;     den3 = den3+mu; }
defuzzification = (num1+num2+num3) / (den1+den2+den3);
```

Figure 8. Implement the defuzzification stage

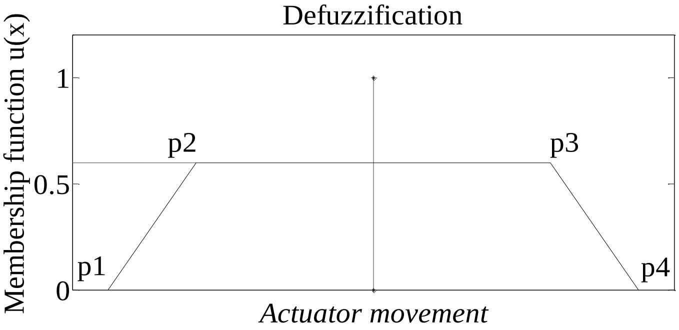
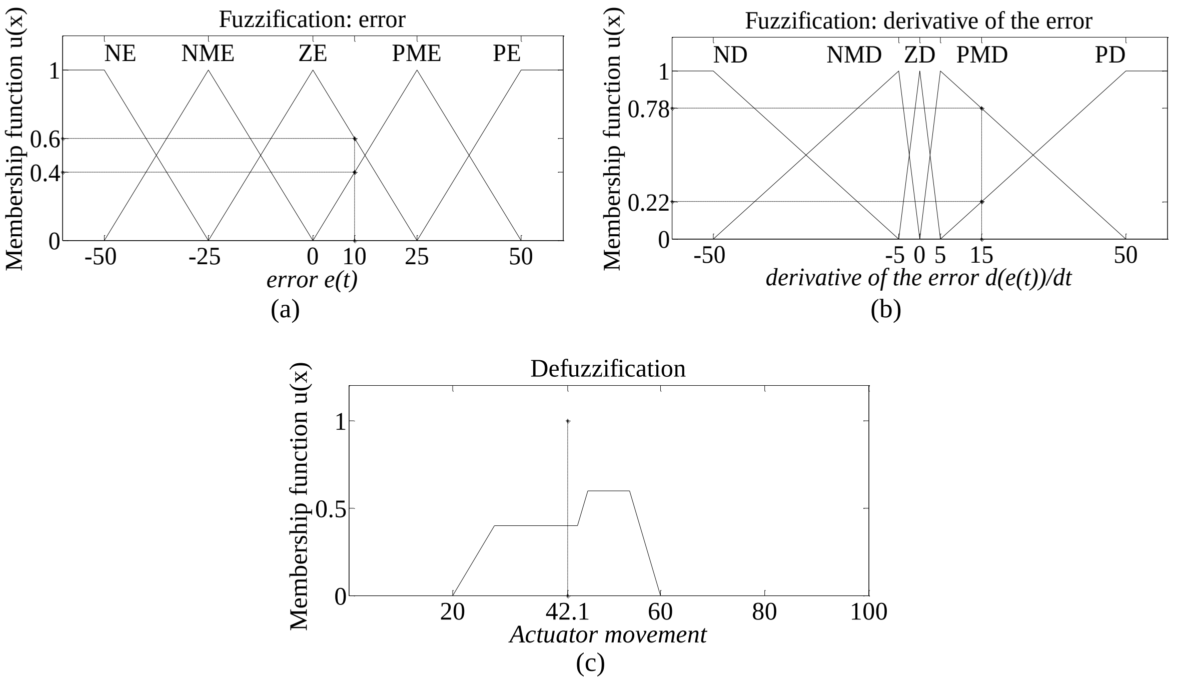


Figure 9. Example of defuzzification

3. EXPERIMENTAL AND SIMULATION RESULTS

The simulation results of the FLC stages were carried out in the MATLAB™ script, which is a program file that can be used to execute a series of commands sequentially, [21]. In addition, the MATLAB™ script and the Arduino Uno board were used, since both platforms use the C/C++ language. Figure 10 shows the results of the FLC simulation, Figure 10(a) shows the fuzzification of the input variable "*error*", Figure 10(b) shows the fuzzification of the input variable "*derivative of the error*", and Figure 10(c) shows the defuzzification. For this test, the parameters were: a value of 10 for the variable "*error*", a value of 15 for the variable "*derivative of the error*", and a defuzzification value of 42.1 was obtained. Table 5 shows the experimental results and the simulation results, which are obtained from the implementation of the FLC in the MATLAB™ script, the Arduino Uno board, and fuzzy logic control™ (MATLAB™ tool to simulate the operation of a FLC). To analyze the results of the FLC, which was implemented in the Arduino Uno board and the MATLAB™ Script, the results of fuzzy logic control™ were used. Therefore, the results of the FLC on the Arduino Uno board and the MATLAB™ script have an error percentage of ±1%, which represents a high degree of accuracy. This high degree of precision is obtained, since variables of type integer and floating point are used, and the control statements "*IF-THEN*" and "*FOR*" are used. Additionally, four digits were used after the decimal point in MATLAB™, and two digits after the decimal point in the Arduino Uno board, which allows obtaining similar results on both platforms. The controllability of the FLC is analyzed to determine if the control action will be carried out in a correct way in the bioreactor. Figure 11 (see in appendix) shows the control surface of the FLC for the different platforms. Figure 11(a) shows the control surface of the FLC implemented in fuzzy logic control™, Figure 11(b) shows the control surface of the FLC implemented in the MATLAB™ script, and Figure 11(c) shows the control surface of the FLC implemented on the Arduino Uno board. The control surface shows that the control in the bioreactor will be performed in a smooth manner and without sudden changes in increase or decrease the humidity percentage in the bioreactor. Finally, the control surface shows that the movement of the actuators will be smooth to avoid damage, errors, or failure of the system.



(a)

(b)

(c)

Figure 10. FLC simulation, (a) fuzzification: "*error*", (b) fuzzification: "*derivative of the error*", and (c) defuzzification

Table 5. Fuzzy controller results in MATLAB™ script, Arduino Uno, and fuzzy logic control™

Input variables		Defuzzification			Percentage of error in the results	
error	derivative of the error	Fuzzy Logic Control™	Script de MATLAB™	Arduino UNO	Script de MATLAB™	Arduino UNO
50	10	20	19.9453	19.95	- 0.2735	- 0.2500
40	- 10	24.9	24.8519	24.85	- 0.1931	- 0.2008
30	5	31.2	31.2174	31.22	+ 0.0557	+ 0.0641
20	- 5	38.7	38.6829	38.68	- 0.0441	- 0.0516
15	20	40.3	40.3133	40.31	+ 0.0330	+ 0.0248
10	- 10	42.1	42.1026	42.10	+ 0.0061	0
5	- 10	44.7	44.7463	44.75	+ 0.1035	+ 0.1118
0	0	50	50	50	0	0
- 5	5	55.3	55.2537	55.25	- 0.0837	- 0.0904
- 10	15	57.9	57.8974	57.90	- 0.0044	0
- 15	15	59.7	59.6867	59.69	- 0.0222	- 0.0167
- 20	10	61.3	61.3171	61.32	+ 0.5495	+ 0.3425
-30	- 10	68.8	68.7826	68.78	- 0.0252	- 0.0290
- 40	- 20	75.1	75.1481	75.15	+ 0.0640	+ 0.0665
- 50	5	80	80	80	0	0

4. IMPLEMENTATION OF THE FUZZY CONTROLLER IN THE BIOREACTOR

The FLC implementation was carried out on the Arduino Uno board. Its main technical data are: Atmega328P microcontroller, 32 KB Flash memory, 16 MHz clock speed, 14 digital inputs/outputs, 6 analog inputs with a resolution of 10 bits, 8 pulse width modulation (PWM) outputs, and its programming language is C/C++ [22]. So, 25850 bytes or 80% of the memory of the Arduino Uno board was used to implement the FLC. Additionally, 614 bytes of dynamic memory were used to store global variables, 1434 bytes were used to store local variables. The time processing was 5 mseg for one iteration. To have greater accuracy, many variables and control statements were used in the FLC processing. The Arduino Uno board was selected due to its low cost, and its programming language is easy. Also, this board is compatible for a wide variety of sensors and actuators. However, the main limitations of this board are the amount of memory and the number of pins, since limit the type and number of devices that can use to control a process. However, the FLC presented in this work can be implemented in other types of platforms such as FPGAs, PLCs, other microcontrollers, or application specific integrated circuit (ASIC). Finally, Figure 12 shows the elements used to control the bioreactor. Figure 12(a) shows the Arduino Uno board, Figure 12(b) shows the direct current (DC) motor and propellers, and Figure 12(c) shows the stepper motor and water tap.

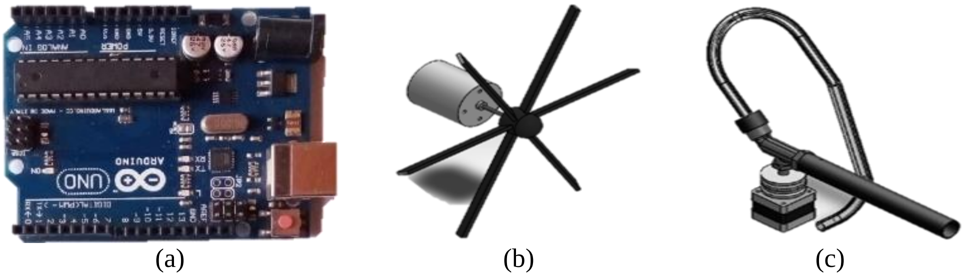
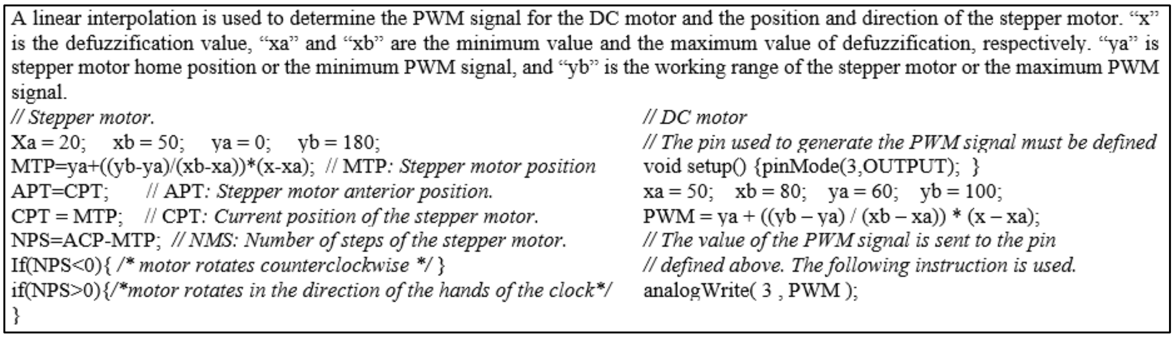


Figure 12. Bioreactor elements, (a) Arduino Uno, (b) DC motor and propellers, and (c) stepper motor and water tap

4.1. Control of the movement of the actuators

A sequence of voltage pulses is used to control the speed and direction of rotation of the stepper motor, which is connected to a water tap to increase the percentage of humidity, and a PWM signal is used to control the angular speed of the DC motor, which is connected to propellers to reduce the percentage of humidity [23]. A linear interpolation is used to correlate the output value of the FLC and the movement of the actuators. The stepper motor advances 1.8° for each step, however, the stepper motor was set to advance 0.9° for each step, which increases the accuracy in the process. Therefore, the stepper motor has a working range of 180º or 200 steps. With this, the water tap connected to the stepper motor will move 180°. On the other hand, a PWM signal with a duty cycle of 60% to 100% is used, which represents the working range of the DC motor. This working range is used for the DC motor, since a PWM signal with a duty cycle of less than 60% cannot modify the percentage of humidity in the bioreactor. Finally, the PWM signal and the voltage pulse sequence are designed and implemented in the Arduino Uno board. The Figure 13 used to control the movement of the actuators.



A linear interpolation is used to determine the PWM signal for the DC motor and the position and direction of the stepper motor. “x” is the defuzzification value, “xa” and “xb” are the minimum value and the maximum value of defuzzification, respectively. “ya” is stepper motor home position or the minimum PWM signal, and “yb” is the working range of the stepper motor or the maximum PWM signal.

```
// Stepper motor.
Xa = 20;   xb = 50;   ya = 0;   yb = 180;
MTP=ya+((yb-ya)/(xb-xa))*(x-xa);  // MTP: Stepper motor position
APT=CPT;     // APT: Stepper motor anterior position.
CPT = MTP;   // CPT: Current position of the stepper motor.
NPS=ACP-MTP;  // NMS: Number of steps of the stepper motor.
If(NPS<0){ /* motor rotates counterclockwise */ }
if(NPS>0){/*motor rotates in the direction of the hands of the clock*/
}
```

```
// DC motor
// The pin used to generate the PWM signal must be defined
void setup() {pinMode(3,OUTPUT);  }
xa = 50;   xb = 80;   ya = 60;   yb = 100;
PWM = ya + ((yb – ya) / (xb – xa)) * (x – xa);
// The value of the PWM signal is sent to the pin
// defined above. The following instruction is used.
analogWrite( 3 , PWM );
```

Figure 13. Control the movement of the actuators

4.2. Bioreactor electrical system

The design and simulation of the electrical system of the bioreactor was carried out in Proteus software, which is a software for the design of electrical circuits. Figure 14 shows the electrical system of the bioreactor, which is made up of the Arduino Uno board, FC-28 sensor, L298N module, PM K031 P1V-17 stepper motor, DC motor, TIP41 transistors, and 220 Ω resistors. The operation of the electrical system is as follows, the analog inputs of the Arduino Uno board were used to connect the FC-28 sensor, which is used to read the percentage of humidity in the bioreactor. Then, the Arduino Uno board executes the FLC, which determines the control action. Finally, the actuators adjust the percentage of humidity in the bioreactor. The bioreactor structure was designed in SOLIDWORKS. Figure 15 shows the design for the construction of the bioreactor, and the bioreactor put into operation. Figures 15(a) and 15(b) show the bioreactor design, which was made in SOLIDWORKS. Figures 15(c), 15(d) and 15(e) show the operation of the bioreactor, that is, the FLC is used to control the percentage of humidity of the bonsai. Finally, the environmental conditions do not affect the operation of the Arduino Uno board, since this board can operate in a temperature range of -40 ºC to +125 ºC. Additionally, the electronic system could be placed away from the bioreactor in case the environmental conditions affect its operation.

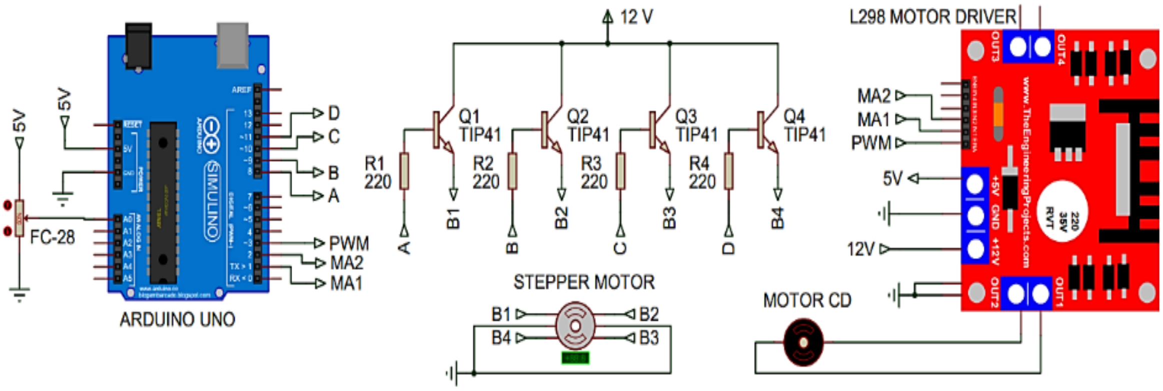


Figure 14. Electrical system of the bioreactor

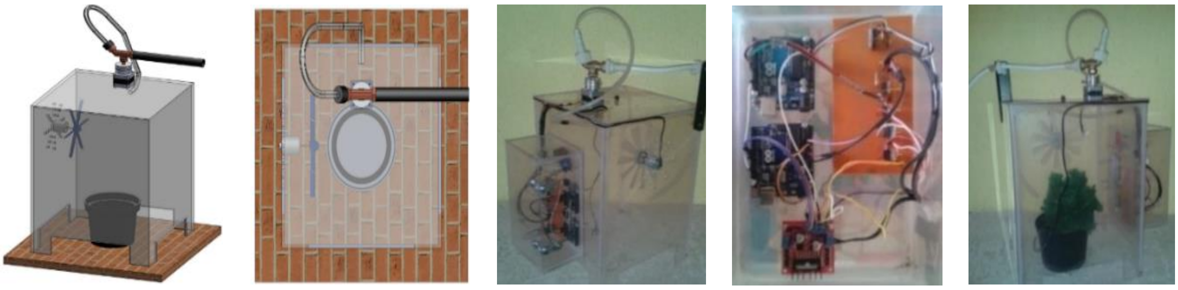

(a) (b) (c) (d) (e)

Figure 15. Design of the bioreactor (a) isometric view of Bioreactor in operation, (b) top view of Bioreactor in operation, (c) bioreactor structure, (d) electric system, and (e) control of the humidity percentage of the bonsai

5. RESULTS

On the other hand, the initial configuration of the bioreactor was 2% humidity percentage, and the desired humidity percentage is 50%. Figure 16 shows the results of the control of the percentage of humidity of the bioreactor. Figure 16(a) shows the variation of the humidity percentage during the execution of the FLC, and as time passes, the humidity percentage is adjusted until the desired humidity percentage is reached. Additionally, a maximum error of ±3% was obtained in the percentage of humidity of the bioreactor. Figure 16(b) shows the mean square error (MSE) of the FLC for the configuration mentioned above. Additionally, the control system takes 12 seconds to converge, that is, the percentage of humidity in the bioreactor has the minimum error with respect to the percentage of humidity desired. Figure 16(c) shows that the stabilization time is 12 seconds for an initial humidity percentage of 2%. Figure 16(d) shows that the stabilization time is 5 seconds for a percentage of initial humidity of 40%. On the other hand, to reach the desired humidity percentage of 50% with an initial humidity percentage of 2%, 35 iterations of the fuzzy controller are required as can be seen in Figure 16(e). Additionally, to reach the desired humidity percentage of 50% with an initial humidity percentage of 40%, 12 iterations of the fuzzy controller are required as can be seen in Figure 16(f). Finally, the stepper motor advances one step every 5 milliseconds, and in each iteration of the fuzzy controller, the number of steps of the stepper motor must be determined.

5.1. Comparison of the fuzzy controller proposed with other control systems

Different control systems such as PID control, fuzzy control, intelligent sensors and actuators, the Ziegler Nichols model, personal computers equipped with some software (for example, LabView), and artificial neuronal networks have been used for the control of some variable in a bioreactor. Also, it has been observed that control strategies based on artificial neural networks require extensive training data to learn the model of complex nonlinear systems, in addition, the accuracy of the process depends on the number of artificial neurons, which increases the complexity of the control system. Also, a PID control is a system prone to oscillation and the adjustment of its parameters is sometimes a complex task. Different investigations have been carried out, which makes the comparison of different control techniques for bioreactors. However, a fuzzy control is the most stable and adequate since it has a shorter settling time and less complexity, which have been showed in different experimental results that confirm the greater effectiveness of fuzzy control compared with other control systems [24]–[26].

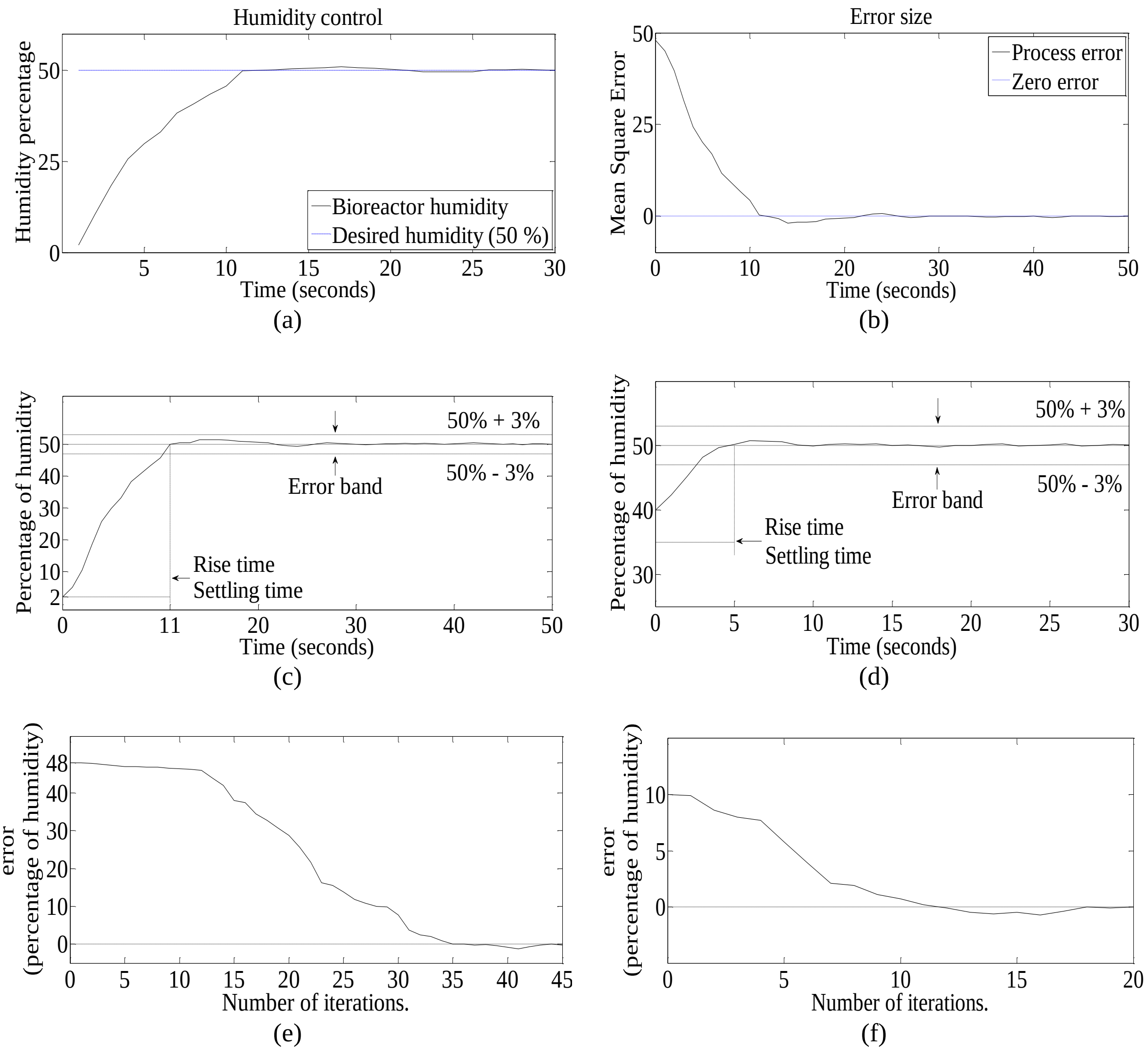
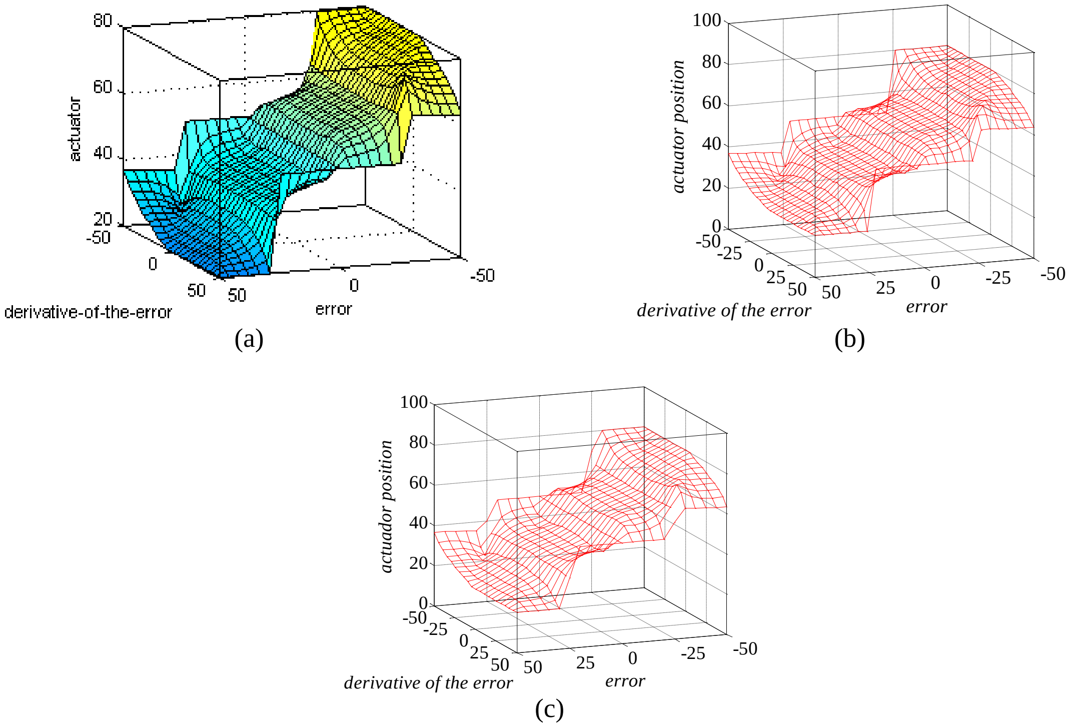


Figure 16. Bioreactor results, (a) humidity, (b) MSE, (c) settling time for a humidity percentage of 2%, (d) Settling time for a humidity percentage of 40%, (e) Iterations of the FLC for the bioreactor to converge with a humidity percentage of 2%, and (f) iterations of the FLC for the bioreactor to converge with a humidity percentage of 40%

6. CONCLUSION

In this work, the humidity percentage is controlled in a bioreactor, used to grow bonsai. The control of the humidity percentage was carried out with a FLC. For this, an algorithm was proposed for its implementation both in software and hardware. The analysis and simulation of the FLC was performed in the MATLAB™ script, using the four basic math operations (addition, subtraction, division, and multiplication) and control statements (IF-THEN and FOR). The fuzzy controller was implemented on the Arduino Uno board. The experimental results obtained on the Arduino Uno board and in the MATLAB™ script have an error rate of ±1%. To determine this error, the results were compared with the results of the Fuzzy Logic Control™ tool. The time processing of the FLC on Arduino Uno board is 5 milliseconds. A sensor is used to measure the percentage of humidity with a resolution of 10 bits which is the same for the input and output variables of the FLC, to increase accuracy and reduce the error in the process. Also, the design of the actuators was carried out with the greatest possible precision. Consequently, a 3% bioreactor humidity percentage error was obtained, and the convergence time for the worst case is 50 seconds.

APPENDIX



(a)

(b)

(c)

Figure 11. Control surface of the FLC, (a) fuzzy logic control™, (b) MATLAB™ script, and (c) 2475 Arduino Uno board

ACKNOWLEDGEMENTS

I would like to thank the Tecnologico de Estudios Superioes de Ecatepec.

REFERENCES

[1] Q. Ye, J. Bao, and Z. Jian-Jiang, *Bioreactor engineering research and industrial applications I*, vol. 155. Springer Berlin Heidelberg, 2016.

[2] M. Benamina, B. Atmani, and S. Benbelkacem, “Diabetes diagnosis by case-based reasoning and fuzzy logic,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 3, 2018, doi: 10.9781/ijimai.2018.02.001.

[3] M. A. Ahmad, M. Z. M. Tumari, and A. N. K. Nasir, “Composite fuzzy logic control approach to a flexible joint manipulator,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, Jan. 2013, doi: 10.5772/52562.

[4] M. A. Soliman, H. M. Hasanien, H. Z. Azazi, E. E. El-Kholy, and S. A. Mahmoud, “An adaptive fuzzy logic control strategy for performance enhancement of a grid-connected PMSG-Based wind turbine,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3163–3173, Jun. 2019, doi: 10.1109/TII.2018.2875922.

[5] M. Medeiros Rocha, J. Patrocinio Da Silva, and F. Das Chagas Barbosa De Sena, “Simulation of a fuzzy control applied to a variable speed wind system connected to the electrical network,” *IEEE Latin America Transactions*, vol. 16, no. 2, pp. 521–526, Feb. 2018, doi: 10.1109/TLA.2018.8327408.

[6] A. Youssef, M. El Telbany, and A. Zekry, “Reconfigurable generic FPGA implementation of fuzzy logic controller for MPPT of PV systems,” *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 1313–1319, Feb. 2018, doi: 10.1016/j.rser.2017.09.093.

[7] A. S. Samosir, H. Gusmedi, S. Purwiyanti, and E. Komalasari, “Modeling and simulation of fuzzy logic based maximum power point tracking (MPPT) for PV application,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 3, pp. 1315–1323, Jun. 2018, doi: 10.11591/ijece.v8i3.pp1315-1323.

[8] U. Yilmaz, A. Kircay, and S. Borekci, “PV system fuzzy logic MPPT method and PI control as a charge controller,” *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 994–1001, Jan. 2018, doi: 10.1016/j.rser.2017.08.048.

[9] S. Najafi, G. W. Flintsch, and S. Khaleghian, “Fuzzy logic inference-based pavement friction management and real-time slippery warning systems: A proof of concept study,” *Accident Analysis and Prevention*, vol. 90, pp. 41–49, May 2016, doi: 10.1016/j.aap.2016.02.007.

[10] R. Belohlavek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics*. Oxford University Press, 2017.

[11] Q. Zhou, H. Li, C. Wu, L. Wang, and C. K. Ahn, “Adaptive fuzzy control of nonlinear systems with unmodeled dynamics and input saturation using small-gain approach,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 1979–1989, Aug. 2017, doi: 10.1109/TSMC.2016.2586108.

[12] N. A. Sofianos and Y. S. Boutalis, “Stable indirect adaptive switching control for fuzzy dynamical systems based on T-S multiple models,” *International Journal of Systems Science*, vol. 44, no. 8, pp. 1546–1565, Aug. 2013, doi: 10.1080/00207721.2012.659697.

[13] N. A. Sofianos and Y. S. Boutalis, “Robust adaptive multiple models based fuzzy control of nonlinear systems,” *Neurocomputing*, vol. 173, pp. 1733–1742, Jan. 2016, doi: 10.1016/j.neucom.2015.09.047.

[14] K. M. Passino, S. Yurkovich, and H.-J. Zimmermann, “Fuzzy Control,” in *Fuzzy Set Theory and Its Applications*, Springer Netherlands, 2001, pp. 223–264.

[15] L. A. Zadeh and R. A. Aliev, *Fuzzy Logic Theory and Applications*. World Scientific, 2018.

[16] K. M. Passino and S. Yurkovich, *Fuzzy control*. Menlo Park Calif.: Addison-Wesley, 1998.

[17] K. Michels, *Fuzzy control : fundamentals, stability and design of fuzzy controllers*. Springer, 2006.

[18] Z. Kovacic and S. Bogdan, *Fuzzy Controller Design: Theory and Applications*. CRC press, 2006.

[19] S. Chakraverty, D. M. Sahoo, and N. R. Mahato, “Defuzzification,” in *Concepts of Soft Computing*, Springer Singapore, 2019, pp. 117–127.

[20] A. Pourabdollah, “Fuzzy number value or defuzzified value; Which one does it better?,” in *IEEE International Conference on Fuzzy Systems*, Jul. 2020, vol. 2020-July, doi: 10.1109/FUZZ48607.2020.9177533.

[21] D. Results, *MATLAB primer*, 8th ed. Boca Raton FL: CRC Press, 2011.

[22] S. Chaouch *et al.*, “DC-motor control using arduino-uno board for wire-feed system,” *Proceedings of 2018 3rd International Conference on Electrical Sciences and Technologies in Maghreb, CISTEM 2018*, Oct. 2019, doi: 10.1109/CISTEM.2018.8613492.

[23] C. R. M. V. Herman I. Veriñaz Jadan, “Deployment of a competition sprinter robot over FPGA platform with feedback control systems for velocity and position,” *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2017.

[24] B. J. Pandian and M. M. Noel, “Control of a bioreactor using a new partially supervised reinforcement learning algorithm,” *Journal of Process Control*, vol. 69, pp. 16–29, Sep. 2018, doi: 10.1016/j.jprocont.2018.07.013.

[25] O. A. Daniel, F. E. Shaibu, and I. O. Megbowon, “Evaluation of three tuning techniques of PID control for a bioreactor process plant,” *2019 15th International Conference on Electronics, Computer and Computation, ICECCO 2019*, Dec. 2019, doi: 10.1109/ICECCO48375.2019.9043279.

[26] B. Wang, Z. Wang, T. Chen, and X. Zhao, “Development of novel bioreactor control systems based on smart sensors and actuators,” *Frontiers in Bioengineering and Biotechnology*, vol. 8, Feb. 2020, doi: 10.3389/fbioe.2020.00007.

BIOGRAPHIES OF AUTHORS

Jesus de la Cruz-Alejo PhD in Electrical Engineering, I did a Postdoctoral Stay at the University of Guadalajara with the topic: "VLSI Analog Design with CMOS Technology". He has carried out different projects related to the design and manufacture of CMOS and FGMOS in low-power, low-voltage VLSI analog mode. He has carried out research projects on intelligent control systems applied to Mechatronics and Alternative Energies systems. Currently, he is a full-time professor at the Tecnológico de Estudios Superiores de Ecatepec developing projects. He can be contacted at email: jdelacruz@tese.edu.mx.

Hugo Beatríz Cuellar was born in Mexico in 1995. He obtained the title of Mechatronic Engineer at Tecnológico de Estudios Superiores de Ecatepec (TESE) in 2019. Currently, he is studying for a master's degree in energy efficiency and renewable energies at the Tecnológico de Estudios Superiores de Ecatepec. He can be contacted at email: 201921499@tese.edu.mx.

Irving Cardiel Alcocer Guillermo holds a degree in mathematics teaching from the Faculty of Mathematics of the Autonomous University of Yucatan in the specialty of Calculus, a Master of Science from the Department of Educational Mathematics of the Center for Research. Currently serves as a teacher in the Technology of Higher Studies of Ecatepec. He is a member of the steering committee of the Mexican Society in Information Technologies, Mechatronics and Telematics A.C., He can be contacted at email: ialcocer@tese.edu.mx.