DESİGN AND SİMULATİON OF FUZZY LOGİC-BASED AUTOMATİC ENVİRONMENTAL CONTROLLER FOR OYSTER MUSHROOM CULTİVATİON

Dakhole Dipali[1*], Thiruselvan Subramanian[1], G Senthil Kumaran[2], Koppala Guravaiah[3]

Article History: **Received:** 01.02.2023 **Revised:**07.03.2023 **Accepted:** 10.04.2023

Abstract.
Traditional agriculture is transforming into smart agriculture due to the use of the Internet of Things. Smart agricultural greenhouses provide a well-controlled environment for crop cultivation to improve the quality and quantity of the yield. However, such a system requires accurate prediction of environmental factors to ensure ideal crop growth and management, which is achieved using an automatic environmental control system.

Mushroom cultivation is an eco-friendly, profitable agribusiness because of its nutritious and medicinal value but it is labor-intensive, and it grows in sensitive environment. It needs to optimize and provide the required growing environmental conditions through an automatic environmental control system. Hence, a smart mushroom growing and monitoring has been proposed using fuzzy logic controller (FLC) to optimize temperature and humidity of mushroom house and thereby to increase mushroom production. This research work proposes an automatic environmental control system using a FLC, which is simulated using MATLAB/Simulink and evaluated for response time of actuators such as sprinkler to optimize temperature and humidifier to optimize humidity. The results are compared with a simulated a proportional, integral, and derivative (PID) controller with evaluation metrics as transient time, settling time, overshoot and peak time. The result shows that response time of sprinkler and humidifier operated using FLC is far lesser than that of PID controller, also it performed better in terms of settling time, peak time and overshoot. This FLC is not only going to be cost effective, but also is going to help to improve mushroom quality and productivity.

Keywords: Oyster mushroom, fuzzy logic controller, PID Controller.

[1*] Computer Science and Engineering, Presidency University, Bengaluru, 560064, India, {prof.dipali,thirulic} @gmail.com, with orcid id https://orcid.org/0000-0003-2722-420X

[2] Agricultural Engineering, ICAR-Indian Institute of Horticulture Research, Bengaluru, 560089, India gsenthil64@gmail.com

[3] Computer Science and Engineering, Indian Institute of Information Technology, Bengaluru, 686518, India kguravaiah@iiitkottayam.ac.in

***Corresponding Author :** Dakhole Dipali
*Computer Science and Engineering, Presidency University, Bengaluru, 560064, India,{prof.dipali, thirulic} @gmail.com, with orcid id https://orcid.org/0000-0003-2722-420X

DOI: xyz

1 Introduction

The oyster mushroom is known for its high nutritional value and medicinal use [1]. Hence, its demand is increasing globally. In 2013-14, the production of mushroom in India was only 17.1 metric ton, whereas in 2020-21 it is increased to 243 metric ton (www.indiastat.com). Generally, manual management of mushroom house environmental conditions such as temperature, humidity, CO_2 level, light intensity, and ammonia concentration, is labor-intensive and time consuming. However, the oyster mushroom cultivation is eco-friendly and profitable business; where it can be grown in a limited space. As per its demand, it needs to optimize in oyster mushroom cultivations by maintaining the most crucial parameters such as humidity and temperature. In manual methods, farmers do not focus on maintaining these parameters, which results in less yield. In addition, poorly ventilated mushroom house affects the growth of the mushroom abnormally with a long stem and small caps. It will result in diseases like rose-comb [1]. Hence, techniques can be implemented by keeping these parameters set to an optimum level in the oyster mushroom house and should be monitored in real-time [2].

In the traditional method of mushroom cultivation, humidity is maintained by hanging gunny sheets or coir mats along the walls and kept wet by watering it frequently during the cropping phase. In addition, during summer, the loss of water due to evaporation is more, and it becomes very difficult to maintain the required humidity and moisture in the substrate. In such a period, sand is spread on the floor and drenched with water. The average yield gained is 60% of the substrate, as there is no fixed controlling mechanism for maintaining mushroom house temperature and humidity, but it can be achieved 100% under proper environmental conditions and crop management [3].

The FLC is a soft computing technique used in classic control to solve non-linearity problems [4] and decision support systems [5]. The purpose of the work leads to an automatic environmental control system using a FLC for oyster mushroom cultivation. In initial designing phase of FLC, it is required to integrate data that carries information on parameters and variables to be controlled as per system requirements [6]. The parameter ranges are taken from domain experts to get the correct operating ranges for devices]

1.1 Conditions of Growing Oyster Mushroom

The necessary conditions for growing oyster mushrooms are as follows:

- The temperature should be 25℃ to 27 ℃ in the Spawn-run phase for 17-18 days and should be 24 ℃ to 26 ℃ for 24-28 days in the cropping phase of mushroom cultivation [2].
- Relative humidity should be above 65% in the Spawn-run phase, whereas it should be 80-85% in the cropping phase for good yield [2].

In oyster mushroom cultivation, the specific environmental parameters such as temperature and humidity, affect the crop reasonably; it needs a complex, nonlinear, time series, and strongly coupled environmental control system, which will keep optimum temperature and humidity in mushroom house [7]. There are two kinds of the controller, a) an open-loop controller with no feedback from the controlled parameter and b) a closed-loop controller, which depends on the current parameter value and feedback parameter [8]. There are various controller systems and mechanisms used to control greenhouses; R. Santhana Krishnan et al. has developed an intelligent irrigation system to control water pump operated with ON/OFF logic using FLC [9]. Songwei Zeng et al. has simulated a hybrid control strategy, combining Radial Basis Function (RBF) network with conventional proportional, integral, and derivative (PID) controllers for the greenhouse climate control [10]. Xue-Bo Jin et al. has proposed an IoT-based environment-sensing system for greenhouse using a bidirectional self-attentive encoder-decoder framework (BEDA) that predicts temperature, humidity, and CO_2 level from noisy IoT-based sensors [7]. P. Javadi Kia et al. has simulated an automatic environmental closed-loop control system for greenhouse using the FLC in MATLAB with soil moisture and calculated period for irrigation to open/close hysteric valve [8]. Rim Ben Alia et al. has developed and simulated dynamic modeling based on a FLC to provide a controlled environment for tomato crop [4]. Romeo Urbieta Parrazales et al. has implemented an automatic irrigation control system using a FLC for rose cultivation by considering controllable variables such as temperature and humidity, and implementation is done using a field-programmable gate array (FPGA) in a domestic greenhouse [11]. Apart from this, an automatic control system is designed using machine vision and FLC to control the performance of rice whitening machine and compared the performance with human operator [5]. Similarly, there are very few automatic controllers developed to control various types of mushroom houses using various computing techniques. Murali Padmanabha et al. has designed and implemented an h/w and s/w based framework for mushroom automatic environmental control system [12]. S. Vellingiri et al. has used machine learning with a predictive analysis approach for predicting the disease in mushroom from history using Internet of Things devices and Kalman filter and then classified using decision tree algorithm [13]. Saiful Azimi has developed an automatic environmental control system based on simple ON/OFF logic to control parameters temperature, humidity, and CO_2 of mushroom house and implemented it using Node MCU ESP8266 V2.0 [14]. Ibrahim Mat et al. has used a feedback control system to optimize temperature, humidity, and CO_2 values for Shiitake mushroom cultivation [15]. G P Cikarge et al. has implemented an automatic humidity controller using FLC to optimize the humidity of oyster mushroom house [2]. Jennifer Dela Cruz-del Amen et al. has implemented an automatic environmental control system using FLC integrated with sound treatement [16].

An open-loop FLC is simulated in MATLAB/Simulink, and evaluated for response time of sprinkler and humidifier with parameters transient time, settling time, overshoot and peak time and the results are compared with a simulated PID controller [17, 18]. The aim is to take advantage of suggestions from the mushroom cultivation expert and design a simple, nonlinear, time series, and strongly coupled environmental control system with FLC, which is going to result in a knowledge-based and an intelligent controller. In this study, the session 2 explains FLC in detail; the session 3 explains the FLC system performance with respect to PID and describes results and discussion, and the session 4 concludes the article.

2 Materials and Methods

The cultivation of oyster mushrooms depends on environmental control factors such as temperature and humidity, which are rigid to monitor manually and meanwhile labor-intensive. The proposed model utilizes FLC to calculate the accurate response time for sprinkler and humidifier, based on current temperature and humidity values as inputs. The mushroom house is going to be maintained by switching ON/OFF sprinkler and humidifier whenever required. As per the oyster mushroom requirement, when the temperature of the house increases above 30°C, the sprinkler should starts to drops down the temperature in the range of 24-30°C. In addition, if the humidity of the mushroom house is less than 65%, the humidifier should start until the humidity increases in the range of 65-85%. The purpose of FLC is to calculate the sprinklers and humidifiers response time to maintain the mushroom house temperature and humidity as per requirement.

2.1 Fuzzy Logic Controller

A fuzzy logic controller is a soft computing technique used to monitor nonlinear systems that are exigent to manipulate mathematically [2]. In addition, it facilitates the implementation of any system into logical processes, like a human brain, which could improve new technologies according to system requirements. Hence, it is a suitable technique for automatically controlling the environmental parameters [11]. There are two approaches to implementing FLC; the Mamdani approach gives output in the form of crisp value only, and the Takagi-Sugeno-Kang approach gives output in mathematical analysis. In both methods, there are three steps to be followed fuzzification, an inference engine (rule base) and defuzzification. As mushroom cultivation requires a system, which is working in uncertain conditions, the Mamdani approach is being selected. This approach gives output in terms of fuzzy values in place of a mathematical equation and gives effective performance in uncertain conditions [11].

2.1.1 Fuzzification: Fuzzification is the method of changing crisp value into fuzzy (linguistic) value, which is presented in the form of fuzzy sets with membership function (MF). MF provides crisp value ranges for fuzzy value. There are various MFs available, such as triangular, trapezoidal, and Gaussian [19]. In proposed research, fuzzification processes two input variables, temperature, and humidity, and processes two output variables, response time of sprinkler and response time of humidifier. The trapezoidal MF has membership values, which slowly increase to particular values, and then remains steady for a period and again start decreasing slowly. It is selected for designing MF for inputs as here the value for temperature and humidity need to be steady in a particular range. In addition, the triangular MF is selected for output variables [9].

The temperature has three MFs Cold, Suitable, and Hot. The range for temperature is considered from 0 to 50 °C. The setpoint value is between 24 to 27°C as per mushroom house requirement. The MF used is trapezoidal with three membership values Cold, Suitable, and Hot. When the temperature of the system is below 24°C, it is treated like a Cold system by a FLC; hence, the sprinkler will be Off. Similarly, when the temperature is in the range of 24 to 27°C, the MF assigned by FLC is Suitable, where the sprinkler will stop sprinkling the water. The last MF is Hot, where the temperature is above 30°C, where the sprinkler starts sprinkling the water as per the respeonse time assigned by FLC for sprinkler. The scale for the MF of temperature is shown in Table 1. The MF for temperature is designed in MATLAB FLC, as shown in Fig 1. The Cold membership value is 1 when the temperature is below 15°C; its membership value starts decreasing when the temperature rises above 15°C, up to 24°C. The MF value for Suitable starts increasing slowly from 15°C and becomes 1 when it is in the range of 24 to 27°C. Iteratively, its value increases when the temperature rises above 27°C, up to 30°C. The Hot membership function value increases with temperature increase from 27°C and becomes one after 30°C. To control the tempearture of mushroom house, the actautor used is sprinkler. It helps to maintain mushroom house tempearture in between 24 to 27°C using FLC.

Table 1. Membership Functions and its ranges for Tempearture.

Membership Values	Membership Function	Domain	Trapezoidal Fuzzy Scale
Cold	1	x≤15	[0 0 15 24]
	24-x/24-15	15 ≤ x ≤ 24	
Suitable	x-15/24-15	15 ≤ x ≤ 24	[15 24 27 30]
	1	24≤ x ≤ 27	
	30-x / 30-27	27 ≤ x ≤ 30	
Hot	x-27 / 30-27	27 ≤ x ≤ 30	[27 30 50 50]
	1	x ≥ 30	

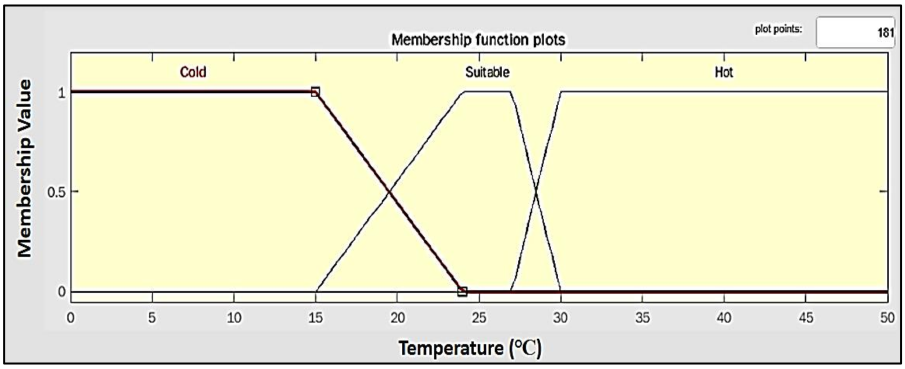


Fig. 1. Fuzzy Memebership Functions for input variable Temperature

The second input variable is humidity, which ranges from 0 to 100% and the set point value is between 65 to 85% as per mushroom house requirement. The MFs designed for humidity are Dry, Suitable, and Wet. The Dry membership value is 1 when humidity is below 65%, and its value decreases with an increase in humidity above 65%. The Suitable membership value increases with humidity increase from 65% and gets a value of 1 when it is in the range of 75 to 80%. In addition, when the humidity starts rising from 80%, the membership value of Suitable is decreasing. Then, Wet membership value increases with an increase in humidity value from 80%. The Wet membership value is 1 when humidity is above 85%. The MFs and their scale is shown in Table 2. The MF for humidity is designed in MATLAB FLC, as shown in Fig 2. To maintain the mushroom house humidity, the actuator used in FLC is humidifier. It helps to keep humidity in the range of 65 to 85%.

Table 2. Membership Functions and its ranges for Humidity.

Membership Values	Membership Function	Domain	Trapezoidal Fuzzy Scale
Dry	1	x ≤ 60	[0 0 60 70]
	70-x / 70-60	60 ≤ x ≤ 70	
Suitable	x-60/ 65-60	60 ≤ x ≤ 65	[60 65 85 90]
	1	65 ≤ x ≤ 85	
	90-x/ 90-85	85 ≤ x ≤ 90	
Wet	x-85/90-85	85 ≤ x≤ 90	[85 90 100 100]
	1	x ≥ 85	

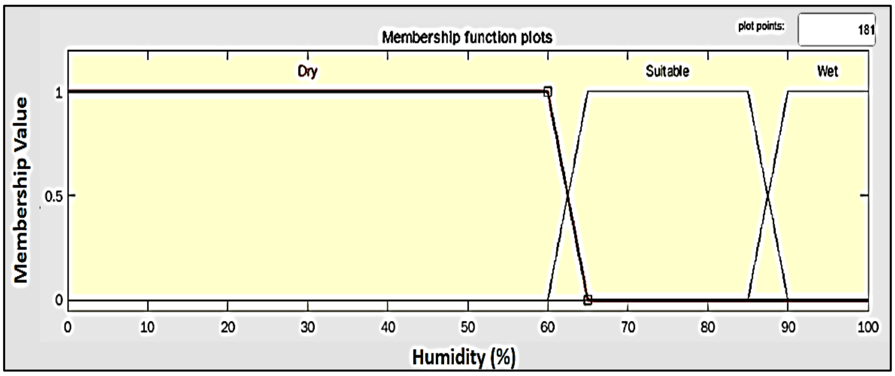


Fig. 2. Fuzzy Memebership Functions for input variable Humidity.

The sprinkler is the output device and FLC is used to calculate its response time to control temperature. The range for the response time of the sprinkler is considered from 0 to 60 minutes. There are four triangular MFs designed to operate the sprinkler are Off, Slow, Medium, and Fast, as shown in Table 3. The response time varies from 0 to 10 minutes for the Off MF. When the Off membership value starts decreasing, the Slow membership value increases to 10 minutes, as shown in Fig 3. Medium MF opeartes in th range of 15 to 40 minutes, whereas Fast MF operates in range of 30 to 60 minutes.

Table 3. Membership Functions and its ranges for Sprinkler Response Time.

Membership Values	Membership Function	Domain	Triangular Fuzzy Scale
Off	(10-x) / (10-0)	0≤ x ≤ 10	[0 0 10]
Slow	(x- 0) / (10-0)	0 ≤ x ≤ 10	[0 10 20]
	(20-x) / (20-10)	10 ≤ x ≤ 20	
Medium	(x-15)/ (27-15)	15 ≤ x ≤ 27	[15 27 40]
	(40-x) (40-27)	27 ≤ x ≤ 40	
Fast	(x-30) / (45-30)	30 ≤ x ≤ 45	[30 45 60]
	(60-x) / (60-45)	45 ≤ x ≤ 60	

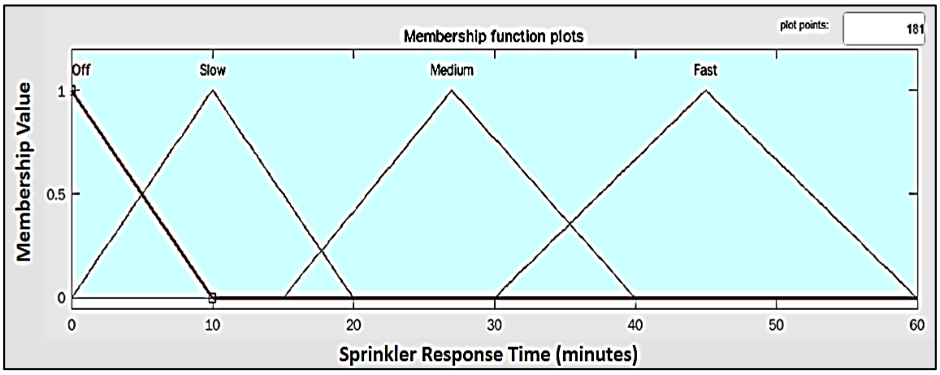


Fig. 3. Fuzzy Memebership Functions for output variable Sprinkler Response Time.

A humidifier is the output device and FLC is used to calculate its response time to control humidity. The range for the response time of the humidifier is considered from 0 to 60 minutes. There are Four MFs Off, Slow, Medium, and Fast, which are designed to operate humidifiers, and their scale is shown in Table 4. When the Off MF value decreases, the Slow membership value increases. When the Slow membership function values decrease, the Medium MF values increase. Similarly, when Medium membership values start decreasing, the Fast MF value increases. The MF for humidifier response time is designed in MATLAB FLC, as shown in Fig 4.

Fuzzification decides MFs and its ranges for all input and output variables of controller. The next step is to connecting these membership functions using rules, called as rule base explained in detail in next section.

Table 4. Membership Functions and its ranges for Sprinkler Response Time.

Membership Values	Membership Function	Domain	Triangular Fuzzy Scale
Off	(10-x) / (10-0)	0 ≤ x ≤ 10	[0 0 10]
Slow	(x- 0) / (10-0)	0 ≤ x ≤ 10	[0 10 20]
	(20-x) / (20-10)	10 ≤ x ≤ 20	
Medium	(x-15)/ (27-15)	15 ≤ x ≤ 27	[15 27 40]
	(40-x) (40-27)	27 ≤ x ≤ 40	
Fast	(x-30) / (45-30)	30 ≤ x ≤ 45	[30 45 60]
	(60-x) / (60-45)	45 ≤ x ≤ 60	

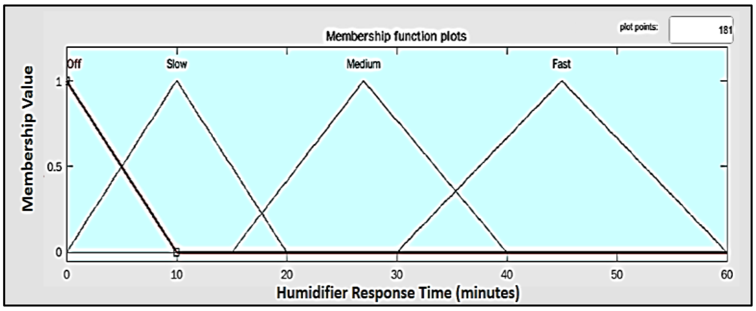
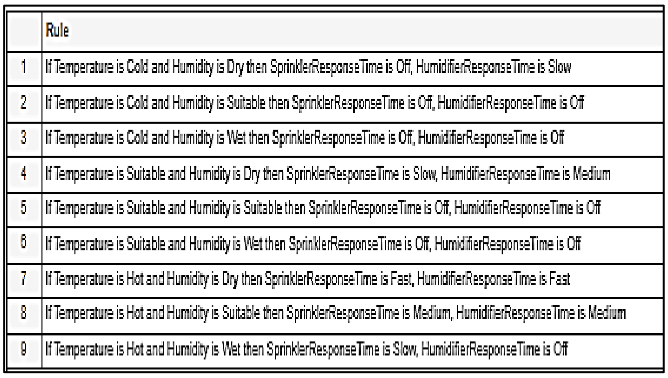


Fig. 4. Fuzzy Memebership Functions for output variable Humidifier Response Time.

2.1.2 Rule Base:

A fuzzy rule base is a collection of rules, created to determine the output. It consists of If-Then rules, as shown in Fig 5. In this, the rules are set so that the sprinkler is responsible for maintaining the temperature, and the humidifier is responsible for maintaining the humidity of the mushroom house. The rule base implements the logic of decision-making, whether sprinkler and humidifier should be ON/OFF. These rules are designed in such a manner that when the membership value for temperature is High, then only the sprinkler should start working, and when the humidity membership value is Dry, then only the humidifier should be ON. The rules in Fig 5 are set using domain expert suggestions and using the trial and error method [2].

An inference engine gives output value for sprinkler and humidifier in terms of fuzzy value. Iteratively, this fuzzy output value needs to be converted into crisp values. This process is called as defuzzification.



	Rule
1	If Temperature is Cold and Humidity is Dry then SprinklerResponseTime is Off, HumidifierResponseTime is Slow
2	If Temperature is Cold and Humidity is Suitable then SprinklerResponseTime is Off, HumidifierResponseTime is Off
3	If Temperature is Cold and Humidity is Wet then SprinklerResponseTime is Off, HumidifierResponseTime is Off
4	If Temperature is Suitable and Humidity is Dry then SprinklerResponseTime is Slow, HumidifierResponseTime is Medium
5	If Temperature is Suitable and Humidity is Suitable then SprinklerResponseTime is Off, HumidifierResponseTime is Off
6	If Temperature is Suitable and Humidity is Wet then SprinklerResponseTime is Off, HumidifierResponseTime is Off
7	If Temperature is Hot and Humidity is Dry then SprinklerResponseTime is Fast, HumidifierResponseTime is Fast
8	If Temperature is Hot and Humidity is Suitable then SprinklerResponseTime is Medium, HumidifierResponseTime is Medium
9	If Temperature is Hot and Humidity is Wet then SprinklerResponseTime is Slow, HumidifierResponseTime is Off

Fig. 5. Fuzzy Rule Base.

2.1.3 Defuzzification:

Defuzzification is done using most common center of gravity (COG) method [11]. COG considers the rules from an inference engine, and triggers below and at the maximum membership level. It consists of finding the centroid of the area bounded by the controller output MF and its abscissa is taken as the crisp controlling value [19]. Hence, while calculating the response time for the output device, instead of considering control actions on extreme ranges of output, it will calculate the centroid of the area. In Simulink, the defuzzification gives results in terms of surface plot. The surface plot for response time of the sprinkler is shown in Fig 6. It shows that when the temperature starts increasing from 30℃ onwards, the sprinkler starts actuating and stabilizes the system when the temperature drops in between 24 ℃ to 27 ℃. Similarly, the surface plot for response time of humidifier is shown in Fig 7. It shows that when the humidity of the mushroom house drops below 65%, the humidifier starts actuating until the humidity reaches 65% and above.

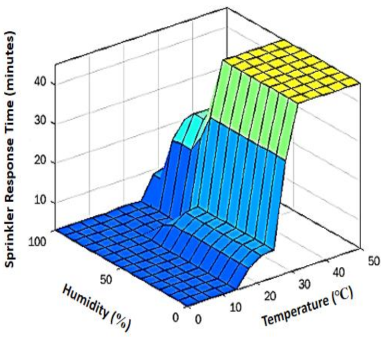


Fig.6 The surface plot for Sprinkler Time

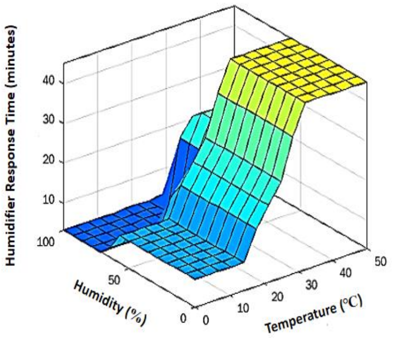


Fig.7 The surface plot for Humidifier Time

2.2 PID Controller

A proportional integrating derivative controller is a standard control loop feedback mechanism widely applicable in industrial control systems. In order to compare FLC with one classical controller we simulated the same system controlled by PID using equation 1 [17].

$$\mu(t) = K_p\, e(t) + K_i \int_0^t e(t)dt + K_d \frac{d\, e(t)}{dt} \qquad (1)$$

Here K_p is the proportional gain, K_i is the integral gain and K_d is the derivative gain along with time t. While designing PID controller the proportional gain and integral gain is used to maintain temperature set points in between 24℃ to 27℃. In addition, the humidity is maintained above 65% by self-tuning PID controller in Simulink. The self-tuning of PID is done using PID tuning app in Simulink. The derivative gain is error value in between set point value and current value, hence it is kept zero initially.

2.3 Experimental Dataset

All constructed datasets used in this simulation are taken from Intel lab data, collected from 54 sensors in Intel Berkeley Research lab between February 28th to April 5th, 2004 [20]. Mica2Dot sensors with weatherboards are used to collect timestamped topology information, along with relative humidity ranging from 0 to 100%, temperature in degrees Celsius, light in Lux, and voltage values in Volts after every 31 seconds. The dataset considered to design a FLC and PID, includes the timestamp, temperature, and relative humidity values. The sample dataset is shown in Table 5. This dataset is divided into two sets. The temp dataset has parameters, time and temperature values, and the humidity dataset has time and humidity variables. The datasets are cleaned and reframed in time series for simplifying the controller functioning.

Table 5. Sample Data Set

Date	Time	Temperature in ℃	Humidity in %
2/28/2004	59:16:0	19.9884	37.0933
2/28/2004	03:16:3	19.3024	38.4629
2/28/2004	06:16:0	19.1652	38.8039
2/28/2004	06:46:8	19.175	38.8379
2/28/2004	08:46:0	19.1456	38.9401
2/28/2004	09:22:3	19.1652	38.872
2/28/2004	09:46:1	19.1652	38.8039
2/28/2004	10:16:7	19.1456	38.8379
2/28/2004	10:46:3	19.1456	38.872
2/28/2004	11:4:9	19.1456	38.9401
2/28/2004	12:46:3	19.1358	38.9061
2/28/2004	14:16:6	19.1162	38.8039
2/28/2004	14:46:6	19.1162	38.872
2/28/2004	15:16:6	19.1064	39.0082

3 The Control System Performance

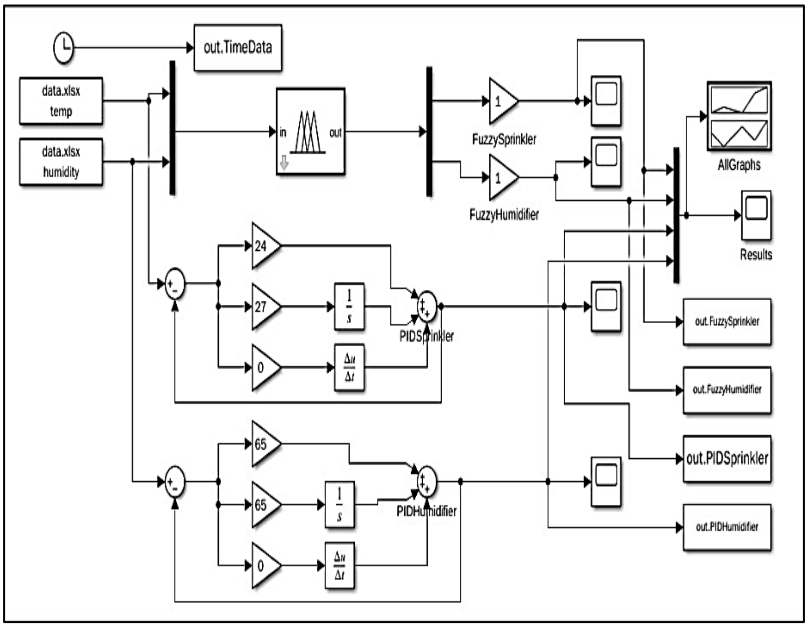


Fig.8 Implementation of FLC and PID controller in Simulink

The dynamic FLC is implemented in MATLAB/Simulink environment, as shown in Fig 8. For comparison PID controller is also implemented in Simulink. The starting two blocks are datasets; two different datasets are designed for temperature and humidity at the same timestamp. These datasets are given as inputs to both FLC and PID controller. The output of FLC and PID is sprinkler response time and humidifier response time, which are collected by the gain component and displayed graphically in scope.

The performance of both controllers are evaluated using transient time [21], which includes delay, overshoot, settling time, rise time and peak time. All results are graphically fetched from scope of Simulink.

3.1 Transient Time

The transient time is the output characteristic of a control system. After applying an input to the control system, the output takes a specific time to reach a steady state. Therefore, the response time of the control system required for the transient state to reach a steady state is called transient time. While calculating transient response, the parameters considered are delay, settling time, maximum overshoot, peak time, and rise time [18].

3.1.1 Delay (t_d): It is a time required for output response to reach its half the final value for the first time.

3.1.2 Maximum Overshoot (M_p): Overshoot is the occurrence of a signal or function exceeding its target. It is calculated using equation 2.

$$M_p = e^{-\zeta\pi/\sqrt{1-\zeta^2}} \tag{2}$$

Where, The damping ratio (ζ) $=\frac{1}{\sqrt{1+(\frac{2\pi}{\ln d})}}$ and $d = \frac{M_{p2}-y_\infty}{M_{p1}-y_\infty}$

3.1.3 Settling Time (T_s): In control theory, the settling time of an output device is the time elapsed from the application of an ideal instantaneous step input to the time at which the device output has entered and remained within a specified error band. The formula for T_s is given in equation 3.

$$T_s = \frac{1}{\zeta\omega_n} \tag{3}$$

Where, natural frequency $\omega_n = \frac{2\pi}{(T_p\sqrt{1-\zeta^2})}$ and $T_p = t_{p2} - t_{p1}$

3.1.4 Rise Time (t_r): It is a time required for output response to reach its final value. It is calculated using equation 4.

$$t_r = \frac{\pi-\Phi}{\omega_d} \tag{4}$$

Where, $\Phi = tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta}$ and $\omega_d = \omega_n\sqrt{1-\zeta^2}$

3.1.5 Peak Time (t_p): It is the time required by output response to reach first peak value of its overshoot. It is calculated using the equation 5.

$$t_p = \frac{\pi}{\omega_d} \tag{5}$$

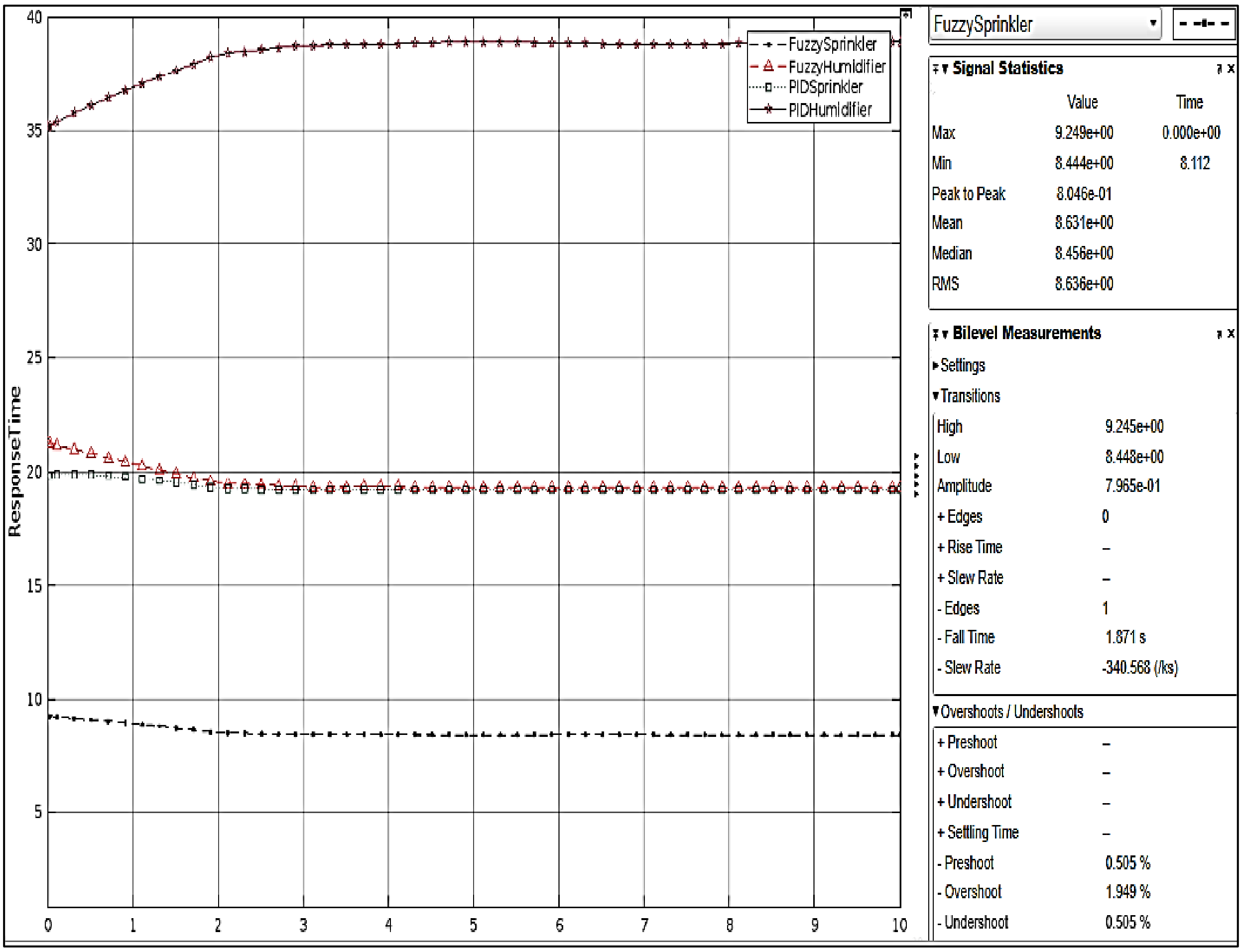
3.2 Comparison between FLC and PID

Fuzzy controller has the capability that can deal with nonlinear systems and use the human expertise knowledge. Here FLC is compared with a linear system PID controller with same system parameters. In PID controller there are only three parameters to adjust, whereas FLC has a lot of parameters to select like membership functions and its parameters, correct choice of rule base. In both controller the ouput is the response time for sprinkler and humidifier. Using response time plots, the values of transient time, settling time, peak time, delay and overshoot is calculated and compared.

3.2.1 Comparison between Response Time of Sprinkler using FLC and PID: The sprinkler is the output device and it's response time is evaluated in Simulink using FLC response plot and PID response plot with all parameters as shown in Fig 9 and Fig 10. It shows that in Table 6 the transient time for fuzzy based sprinkler is lesser than that PID base. It means fuzzy based sprinkler acheives better and fast stability. Also the overshoot is lesser in fuzzy sprinkler than that in PID, less overshoot means less difference in value from required value. Furthermore, it responds faster with smaller settling time and peak values than that of PID based sprinkler.

Table 6. Evaluation Metrix Results for FLC and PID Sprinkler Time

Performance Parameters for Sprinkler	Parameters value using FLC	Parameters value using PID
Transient Time	4.1119	7.2451
Settling Time	8.8463	19.5114
Settling Min	8.4440	19.1456
Settling Max	9.2486	19.8772
Overshoot	1.9490	2.3900
Peak	9.2486	19.8772
Peak Time	0	0.3116



Signal Statistics

	Value	Time
Max	9.249e+00	0.000e+00
Min	8.444e+00	8.112
Peak to Peak	8.046e-01	
Mean	8.631e+00	
Median	8.456e+00	
RMS	8.636e+00	

Bilevel Measurements

Settings

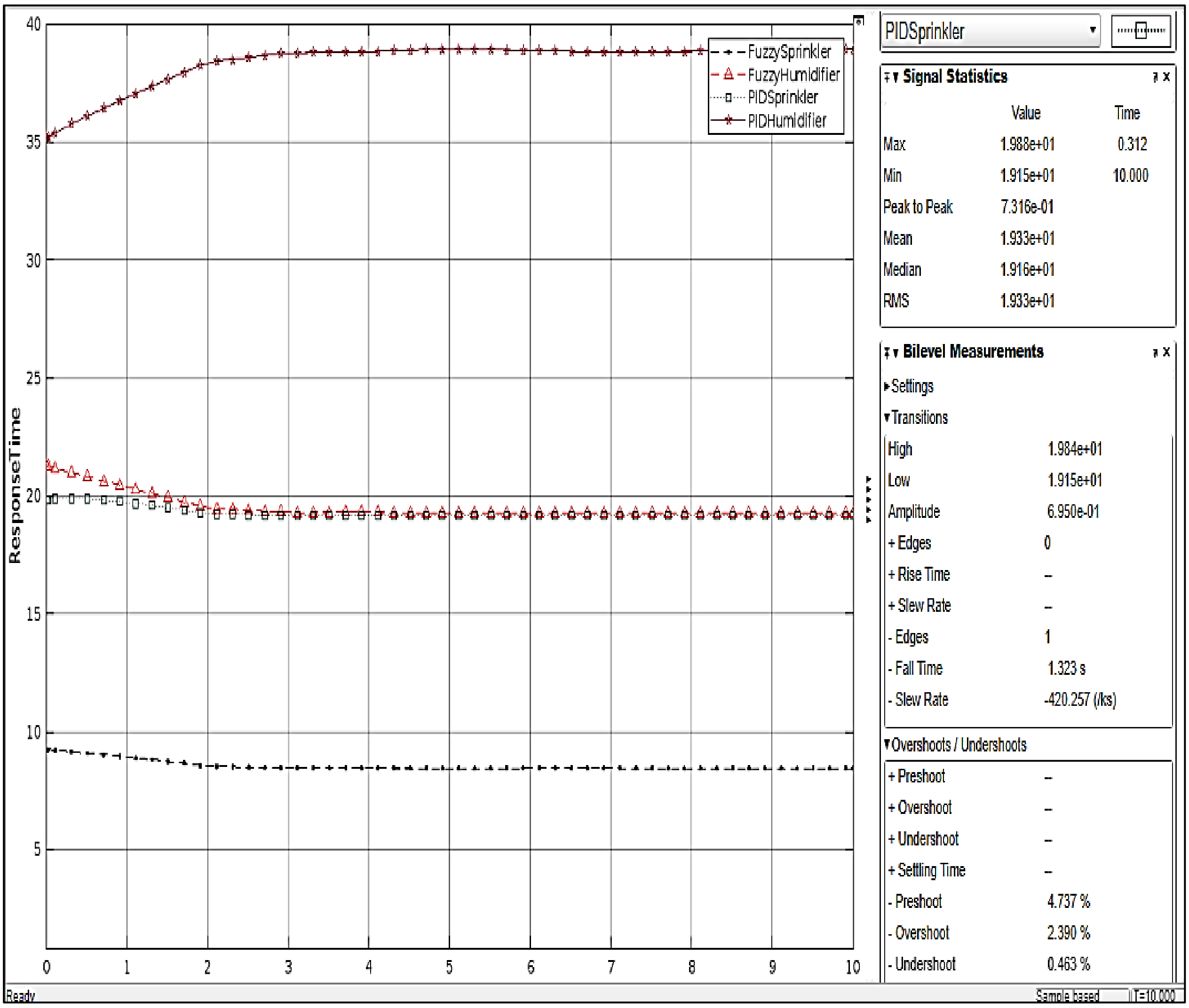
Transitions

High	9.245e+00
Low	8.448e+00
Amplitude	7.965e-01
+ Edges	0
+ Rise Time	–
+ Slew Rate	–
- Edges	1
- Fall Time	1.871 s
- Slew Rate	-340.568 (/ks)

Overshoots / Undershoots

+ Preshoot	–
+ Overshoot	–
+ Undershoot	–
+ Settling Time	–
- Preshoot	0.505 %
- Overshoot	1.949 %
- Undershoot	0.505 %

Fig. 9 Performance parameters for response time for sprinkler using FLC



Signal Statistics

	Value	Time
Max	1.988e+01	0.312
Min	1.915e+01	10.000
Peak to Peak	7.316e-01	
Mean	1.933e+01	
Median	1.916e+01	
RMS	1.933e+01	

Bilevel Measurements

Settings

Transitions

High	1.984e+01
Low	1.915e+01
Amplitude	6.950e-01
+ Edges	0
+ Rise Time	--
+ Slew Rate	--
- Edges	1
- Fall Time	1.323 s
- Slew Rate	-420.257 (/ks)

Overshoots / Undershoots

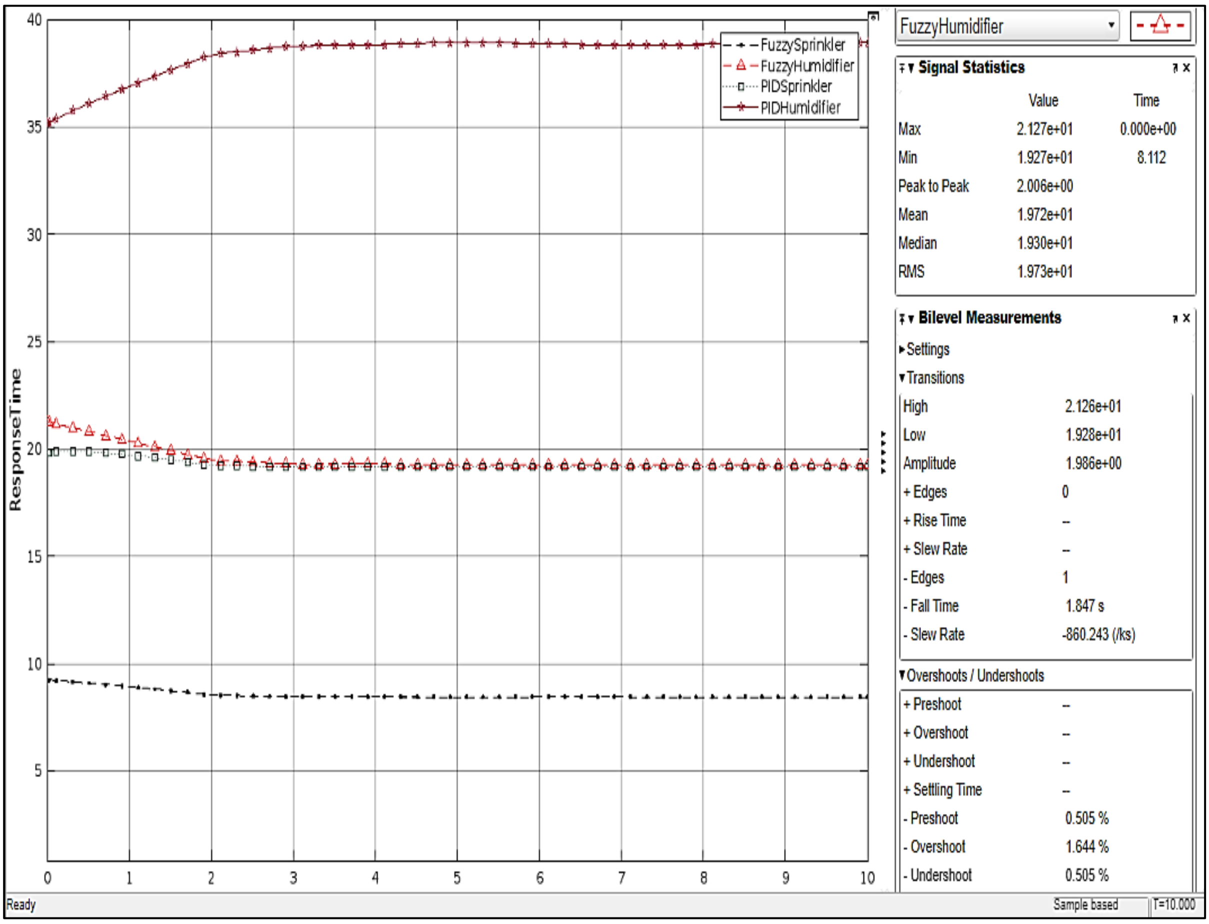
+ Preshoot	--
+ Overshoot	--
+ Undershoot	--
+ Settling Time	--
- Preshoot	4.737 %
- Overshoot	2.390 %
- Undershoot	0.463 %

Fig. 10 Performance parameters for response time for sprinkler using PID

3.2.2 Comparison between Response Time of Humidifier using FLC and PID: The humidifier is the output device and it's response time is evaluated in Simulink uisng FLC response plot and PID response plot with its parameters as shown in Fig 11 and Fig 12. It shows that in Table 7 the transient time for fuzzy based humidifier is lesser than that PID based, which means it achieves better and faster stability than PID humdifier. Also the overshoot is lesser in fuzzy humidifier than that in PID, less overshoot means less difference in value from required value. Furthermore, it responds faster with smaller settling time and peak values than that of PID based humidifier.

Table 7. Evaluation Metrix Results for FLC and PID Humidifier Time

Performance Parameters for Humidifier	Parameters value using FLC	Parameters value using PID
Transient Time	2.9461	8.7821
Settling Time	20.2718	37.0608
Settling Min	19.2689	35.1824
Settling Max	21.2748	38.9393
Overshoot	1.6440	2.2360
Peak	21.2748	38.9393
Peak Time	0	10



Signal Statistics

	Value	Time
Max	2.127e+01	0.000e+00
Min	1.927e+01	8.112
Peak to Peak	2.006e+00	
Mean	1.972e+01	
Median	1.930e+01	
RMS	1.973e+01	

Bilevel Measurements

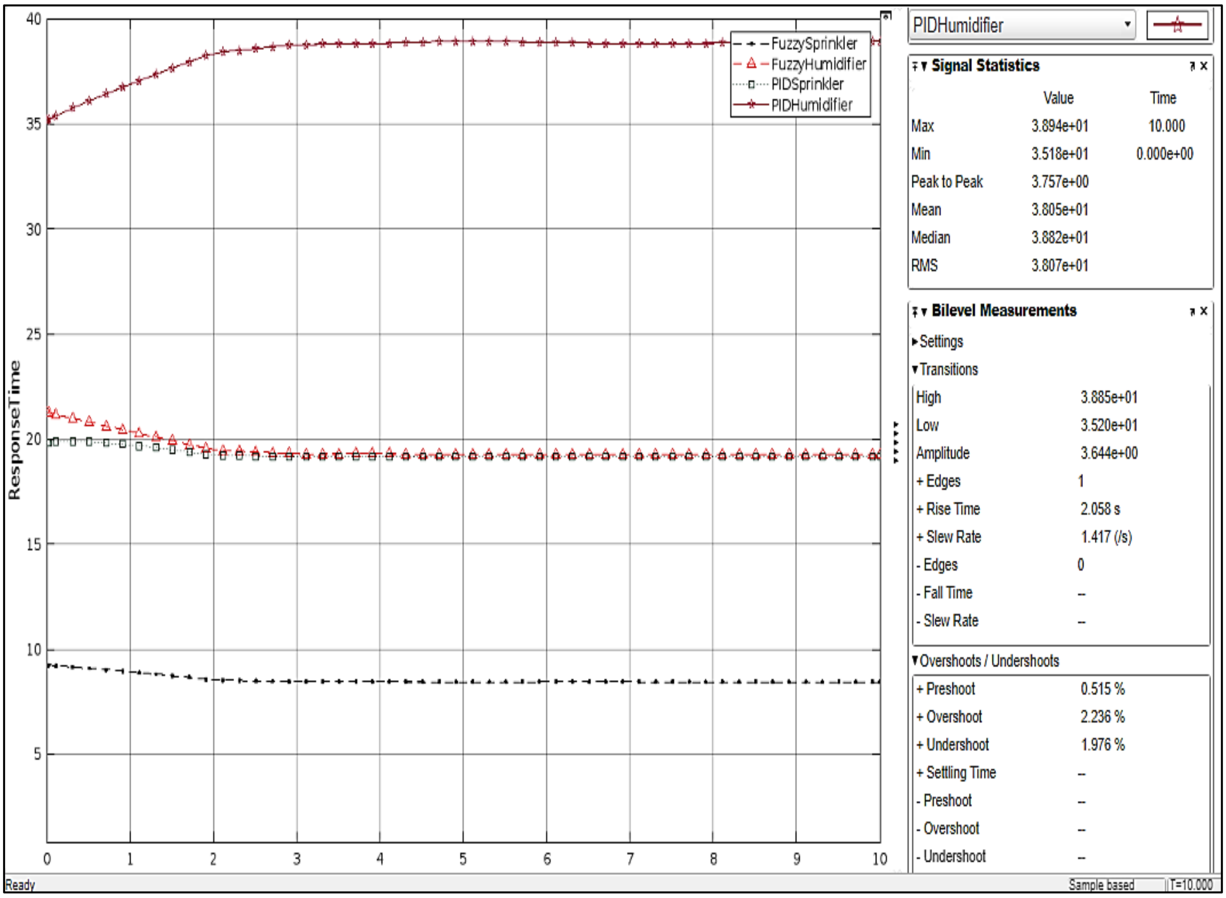
- Settings
- Transitions

High	2.126e+01
Low	1.928e+01
Amplitude	1.986e+00
+ Edges	0
+ Rise Time	--
+ Slew Rate	--
- Edges	1
- Fall Time	1.847 s
- Slew Rate	-860.243 (/ks)

- Overshoots / Undershoots

+ Preshoot	--
+ Overshoot	--
+ Undershoot	--
+ Settling Time	--
- Preshoot	0.505 %
- Overshoot	1.644 %
- Undershoot	0.505 %

Fig. 11 Performance parameters for response time for humidifier using FLC



Signal Statistics

	Value	Time
Max	3.894e+01	10.000
Min	3.518e+01	0.000e+00
Peak to Peak	3.757e+00	
Mean	3.805e+01	
Median	3.882e+01	
RMS	3.807e+01	

Bilevel Measurements

- Settings
- Transitions

High	3.885e+01
Low	3.520e+01
Amplitude	3.644e+00
+ Edges	1
+ Rise Time	2.058 s
+ Slew Rate	1.417 (/s)
- Edges	0
- Fall Time	--
- Slew Rate	--

- Overshoots / Undershoots

+ Preshoot	0.515 %
+ Overshoot	2.236 %
+ Undershoot	1.976 %
+ Settling Time	--
- Preshoot	--
- Overshoot	--
- Undershoot	--

Fig. 12 Performance parameters for response time for humidifier using PID

The Simulink graphs clearly shows that the fuzzy based sprinkler and humidifier is performing better than PID based, in terms of transient time, settling time, overshoot and peak time.

4 Conclusion

The environmental conditions for Oyster mushroom cultivation are not always favorable. Hence, a dynamic model for an automatic environmental control system is designed and modeled in MATLAB/Simulink. The proposed model is based on FLC, which tries to keep oyster mushroom house temperature in the range of 24-27℃ and humidity in the range of 65-85%. FLC is a knowledge-based system without using traditional mathematical methods and helping to eliminate the uncertainty in the measurements using expert guidance. Hence, it increases the robustness of the model by oprtimizing its tranient time.

Furthermore, the FLC is compared with the PID controller for validation of results. FLC has given far better results than PID in terms of transient time, settling time, peak time and overshoot. In addition, the response time range for both sprinkler and humidifier is less as compared to that of the PID. Finally, FLC is going to improve the traditional mushroom cultivation process with less effort and great precision.

Appendix A. Supplementary Material

Supplementary data used in this study can be found, in online version,at http://db.csail.mit.edu/ labdata/labdata.html

References

1. Lambert, E. B. (1938). Principles and problems of mushroom culture. The Botanical Review, 4(7), 397-426. https://link.springer.com/article/10.1007/BF02 872538
2. Cikarge, G. P., & Arifin, F. (2018, December). Oyster Mushrooms Humidity Control Based On Fuzzy Logic By Using Arduino ATMega238 Microcontroller. In Journal of Physics: Conference Series (Vol. 1140, No. 1, p. 012002). IOP Publishing. 10.1088/1742-6596/1140/1/012002
3. Meera Pandey, G. Senthil Kumaran, Veena S. S., R.P. Tewari, Chandrashekara C. (2021) Production Technology of Edible and Medicinal Mushrooms. ICAR-Indian Institute of Horticultureal Research, Benagluru-India.
4. Ali, R. B., Bouadila, S., & Mami, A. (2018). Development of a Fuzzy Logic Controller applied to an agricultural greenhouse experimentally validated. Applied Thermal Engineering, 141, 798-810. https://doi.org/10.1016/j.applthermaleng.2018. 06.014
5. Prabakaran, G., Vaithiyanathan, D., & Ganesan, M. (2018). Fuzzy decision support system for improving the crop productivity and efficient use of fertilizers. Computers and electronics in agriculture, 150, 88-97. 10.1016/j.compag.2018.03.030
6. Urbieta Parrazales, R., Zagaceta Álvarez, M. T., Aguilar Cruz, K. A., Palma Orozco, R., & Fernández Muñoz, J. L. (2021). Implementation of a Fuzzy Logic Controller for the Irrigation of Rose Cultivation in Mexico. Agriculture, 11(7), 576. https://doi.org/10.3390/agriculture11070576
7. Jin, X. B., Zheng, W. Z., Kong, J. L., Wang, X. Y., Zuo, M., Zhang, Q. C., & Lin, S. (2021). Deep-Learning Temporal Predictor via Bidirectional Self-Attentive Encoder–Decoder Framework for IOT-Based Environmental Sensing in Intelligent Greenhouse. Agriculture, 11(8),802. https://doi.org/10.3390 / agriculture11080802
8. Kia, P. J., Far, A. T., Omid, M., Alimardani, R., & Naderloo, L. (2009). Intelligent control based fuzzy logic for automation of greenhouse irrigation system and evaluation in relation to conventional systems. World applied sciences journal, 6(1), 16-23. http://www.idosi.org/wasj/wasj6(1)/3.pdf
9. Krishnan, R. S., Julie, E. G., Robinson, Y. H., Raja, S., Kumar, R., & Thong, P. H. (2020). Fuzzy logic based smart irrigation system using internet of things. Journal of Cleaner Production, 252, 119902. https://agris.fao.org /agris-search/search.do?recordID=US202000131647
10. Zeng, S., Hu, H., Xu, L., & Li, G. (2012). Nonlinear adaptive PID control for greenhouse environment based on RBF network. Sensors, 12(5), 5328-5348. https://doi.org/10.3390/s120505328
11. Urbieta Parrazales, R., Zagaceta Álvarez, M. T., Aguilar Cruz, K. A., Palma Orozco, R., & Fernández Muñoz, J. L. (2021). Implementation of a fuzzy logic controller for the irrigation of rose cultivation in mexico. Agriculture, 11(7), 576. https://doi.org/10.3390/agriculture11070576
12. Padmanabha, M., & Streif, S. (2019). Design and validation of a low cost programmable controlled environment for study and production of plants, mushroom, and insect larvae. Applied Sciences, 9(23), 5166.

https://doi.org/10.3390/app9235166

13. Velliangiri, S., Sekar, R., & Anbhazhagan, P. (2020). Using MLPA for smart mushroom farm monitoring system based on IoT. International Journal of Networking and Virtual Organisations, 22(4), 334-346. https://doi.org/10.1504/ijnvo.2020.107559
14. Mahmud, M. A., Buyamin, S., Mokji, M. M., & Abidin, M. Z. (2018). Internet of things based smart environmental monitoring for mushroom cultivation. Indonesian Journal of Electrical Engineering and Computer Science, 10(3), 847-852. http://doi.org/10.11591/ijeecs.v10.i3.pp847-852
15. Mat, I., Kassim, M. R. M., Harun, A. N., & Yusoff, I. M. (2017, November). Environment control for smart mushroom house. In 2017 IEEE Conference on Open Systems (ICOS) (pp. 38-42). EEE.DOI:10.1109/ICOS.2017.8280271
16. Dela Cruz-del Amen, J., & Villaverde, J. F. Fuzzy Logic-based Controlled Environment for the Production of Oyster Mushroom. In 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM) (pp. 1-5). IEEE. 10.1109/HNICEM 48295.2019.9072902
17. Dantas, A. D. O. D. S., Dantas, A. F. O. D. A., Campos, J. T. L., de Almeida Neto, D. L., & Dórea, C. E. T. (2018). PID control for electric vehicles subject to control and speed signal constraints. Journal of Control Science and Engineering, 2018. https://doi.org/10.1155/2018/6259049
18. Altbawi, S. M. A., Mokhtar, A. S. B., Jumani, T. A., Khan, I., Hamadneh, N. N., & Khan, A. (2021). Optimal design of Fractional order PID controller based Automatic voltage regulator system using gradient-based optimization algorithm. Journal of King Saud University-Engineering Sciences. 10.1016/j.jksues.2021. 07.009
19. Saade, J. J., & Diab, H. B. (2004). Defuzzification methods and new techniques for fuzzy controller.
20. Xu, Y., Sun, G., Geng, T., & He, J. (2019). Low-energy data collection in wireless sensor networks based on matrix completion. Sensors, 19(4), 945. https://doi.org/10.3390/s19040945
21. Muliadi, J., & Kusumoputro, B. (2018). Neural network control system of UAV altitude dynamics and its comparison with the PID control system. Journal of Advanced Transportation, 2018. 10.1155/2018/ 3823201