Performance Comparison between Gain Scheduling PID and Fuzzy Mamdani Controllers for an Air Heating Module

Rizky Maulana Hendradi[1], Arief Rahman Hakim[2], Septiantar Tebe Nursaputro[3], Wahyu Sulistiyo [4]

[1,2,3,4] *Jurusan Teknik Elektro, Politeknik Negeri Semarang, Jawa Tengah, 50275, Indonesia*

Abstract—This study compares the performance of Gain Scheduling Proportional–Integral–Derivative (GS-PID) control and Fuzzy Mamdani control applied to an air heating module. The system was implemented using an STM32 microcontroller and evaluated experimentally under temperature setpoints of 50 °C, 60 °C, 70 °C, and 80 °C. Performance was assessed based on dynamic response parameters, including rise time, overshoot, steady-state error (SSE), Integral of Absolute Error (IAE), Integral of Squared Error (ISE), and Integral of Time-weighted Absolute Error (ITAE).

The experimental results show distinct characteristics between the Gain Scheduling PID (GS-PID) and Fuzzy Mamdani controllers under varying setpoints. The GS-PID controller exhibits a rise time of approximately 4.0–4.9 **s** with relatively small overshoot in the range of 1.18–2.87%, while maintaining a very low steady-state error of about 0.001–0.24%. In contrast, the Fuzzy Mamdani controller achieves a faster rise time of around 2.0–2.6 **s** at low to medium setpoints, which increases to approximately **5 s** at higher temperatures. The overshoot produced by the Fuzzy Mamdani method ranges from 0–4.78%, decreasing as the setpoint increases, with zero overshoot observed at high setpoints due to incomplete setpoint tracking. Moreover, the Fuzzy Mamdani controller yields higher performance index values (IAE, ISE, and ITAE), indicating larger accumulated error over time. Overall, GS-PID control is more effective in ensuring steady-state accuracy and stability, whereas Fuzzy Mamdani control provides a faster transient response at lower operating temperatures.

Key— Gain Scheduling PID, Fuzzy Mamdani, temperature control, air heating module, performance comparison

1. Introduction

Temperature control is a fundamental topic in control engineering education, particularly in vocational higher education institutions such as Politeknik Negeri Semarang. Small-scale air heater modules are commonly used as instructional media and experimental platforms to study and evaluate automatic temperature control methods due to their fast dynamic response and strong nonlinear characteristics. These properties make air heater systems suitable for testing advanced control strategies under varying operating conditions.

One of the most widely applied temperature control methods is the Proportional–Integral–Derivative (PID) controller. PID control is popular because of its reliability and applicability to a wide range of systems, especially in cases where an accurate mathematical model is unavailable, making analytical design methods difficult to apply (Katsuhiko Ogata, 2010) .The PID controller operates based on three main parameters proportional (Kp), integral (Ki), and derivative (Kd) which mathematically represent the system response to control error (Enriko et al., 2021). To achieve optimal performance, these parameters must be properly tuned. Several conventional tuning methods have been developed, including trial-and-error, Ziegler–Nichols, and step response methods (Allu & Toding, 2020). However, a major limitation of these methods is that they are designed for a single operating point or setpoint, resulting in performance degradation when the setpoint changes (Rifky Adji Fadlani, 2024). Consequently, retuning becomes necessary to maintain system stability and accuracy under varying operating conditions (Ghazali et al., 2024).

To overcome this limitation, adaptive PID control approaches have been introduced, one of which is Gain Scheduling PID (GS-PID). Gain scheduling is an adaptive control technique in which PID parameters are scheduled according to the system's operating conditions or setpoints (Karyatanti et al., 2024). Previous studies have demonstrated that GS-PID is capable of maintaining stable performance in nonlinear temperature control systems subjected to setpoint variations (Cui et al., 2022; Simanjuntak et al., 2022).

In addition to adaptive PID control, intelligent control methods such as Fuzzy Mamdani have been widely applied to temperature regulation problems. Fuzzy Mamdani control, introduced by Ebrahim Mamdani in 1975, is designed to handle uncertainty and nonlinearity through linguistic rules and membership functions (Qurrata Ayun et al., 2025; Sukandy et al., 2008), In this study, the Fuzzy Mamdani controller is implemented as a non-adaptive control method, where the fuzzy rules and membership functions remain constant during operation. The control output is generated through a series of processes including fuzzification, fuzzy inference, rule evaluation, and defuzzification (Salim & Rahman, 2022). Several studies have reported that Fuzzy Mamdani control can achieve satisfactory system responses that follow the desired input behavior (Amalia et al., n.d.; Sunanto et al., 2021).

Despite the advantages of both GS-PID and Fuzzy Mamdani control methods, direct performance comparisons between these two approaches on the same air heater system remain

limited, particularly in vocational laboratory-scale applications. Therefore, this study aims to experimentally compare the performance of GS-PID and Fuzzy Mamdani controllers applied to a small-scale air heater module operating within a temperature range of 40–80 °C. The air heater module consists of a resistive nichrome heating coil and a DC blower fan, designed to represent the nonlinear characteristics and fast dynamics commonly encountered in laboratory-scale thermal systems. The performance of both control methods is evaluated based on rise time, settling time, overshoot, and steady-state error.

The results of this study are expected to contribute to the development of control system learning in vocational education and provide a foundation for the application of adaptive and intelligent temperature control methods in practical thermal systems such as food warmers (S.Mathivanan et al., 2017), small-scale food dryers (Gonçalves et al., 2023), and environmental chambers for building material testing (Palani & Karatas, 2023).

2. Literature Review

2.1. Proportional-Integral-Derivative (PID)

Proportional–Integral–Derivative (PID) control consists of three components: proportional, integral, and derivative, represented by the gains K_p, K_i, and K_d. These gains depend on the error between the system input and output (Joseph et al., 2022). Proper tuning improves dynamic response, reduces overshoot, eliminates steady-state error, and enhances system stability. The PID control equation and block diagram are presented in Equation (1) and Figure 1.

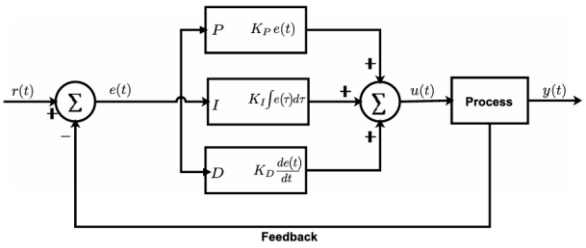


Figure 1. Structure of the PID control system

The variables in the PID control system are defined as follows: $e(t)$represents the difference between the setpoint and the system output, $u(t)$is the control signal generated from the summation of the proportional (P), integral (I), and derivative (D) actions, $y(t)$denotes the system output, and $r(t)$represents the setpoint or reference value.

$$\text{Output} = Kp \times e(t) + Ki \times \int_0^t e(t)dt + Kd \times \frac{de}{dt} \quad (1)$$

where K_p, K_i, and $K_d$represent the proportional (P), integral (I), and derivative (D) gains, respectively.

2.2. Gain Scheduling PID

Gain scheduling Proportional–Integral–Derivative (PID) control is an adaptive control method designed to overcome the limitations of conventional PID control. In nonlinear systems or systems with time-varying dynamics, fixed PID parameters, namely the proportional (K_p), integral (K_i), and derivative (K_d) gains often fail to provide optimal performance. This condition may result in slow system response, overshoot, or even instability when operating conditions change.

The basic principle of gain scheduling is the dynamic adjustment of the PID gains based on variations in a reference variable, known as the scheduling variable. This variable may include the setpoint, speed, temperature, or the magnitude of the system error. By adapting the controller parameters to the operating conditions, the system can maintain the desired response characteristics over a wide operating range. The general mathematical form of the gain scheduling PID controller is expressed in Equation (2).

$$\text{Output} = Kp(\mathbf{x})e(t) + Ki(x)\int_0^t e(t)dt + Kd(x)\frac{de(t)}{dt} \quad (2)$$

where $e(t)$represents the error between the setpoint and the system output, while $x$denotes the scheduling variable that influences the values of $K_p(x)$, $K_i(x)$, and $K_d(x)$. These parameters can be adjusted using predefined tables, mathematical functions, or specific algorithms to match the system operating conditions.

With this approach, gain scheduling PID control is able to improve the system's dynamic response, reduce settling time, minimize overshoot, and maintain system stability even under load variations or external disturbances.

2.3. Fuzzy Mamdani

Mamdani fuzzy logic is extensively applied in control systems because of its capability to handle uncertainty, nonlinearity, and processes that are difficult to represent using precise mathematical models. The method was first introduced by Ebrahim Mamdani in 1975 for the control of a steam generator. The Mamdani fuzzy approach employs linguistic reasoning based on IF–THEN rules, which emulate human decision-making derived from expert knowledge and experience. Generally, a Mamdani fuzzy control system comprises four main stages: fuzzification, rule base formulation, fuzzy inference, and defuzzification.

2.3.1. Fuzzification

Fuzzification is the process of transforming crisp input values into fuzzy membership values using predefined membership functions, such as triangular, trapezoidal, or Gaussian functions.

2.3.2. Rule Base Formulation

The rule base consists of a set of IF–THEN rules that define the relationship between input and output variables based on linguistic logic. In the Mamdani fuzzy method, the rule base employs the Min operator. The general form is expressed as follows:

$$\text{If } a \text{ is } A_i \text{and } b \text{ is } B_i, \text{ then } c \text{ is } C_i$$

where A_i, B_i, and $C_i$are fuzzy predicates representing the linguistic values of their respective variables. The number of rules is determined by the number of linguistic values for each input variable.

2.3.3. Fuzzy Inference

Fuzzy inference is the decision-making process that combines the active rules using the fuzzy Max operator. In this stage, the fuzzy set solution is obtained by taking the maximum value from the rule base, which is then used to modify the fuzzy domain and applied to the output using the OR operator (union). Once all rules have been evaluated, the output consists of a fuzzy set that reflects the contribution of each rule. In general, this can be expressed in Equation (3).

$$\mu(X_i) = \max(\mu_{sf}(X_i), \mu_{kf}(X_i)) \quad (3)$$

The variables are defined as follows: $\mu(X_i)$represents the membership degree, $\mu_{sf}(X_i)$denotes the fuzzy membership value of the solution up to the i-th rule, and $\mu_{kf}(X_i)$corresponds to the fuzzy membership value of the consequent of the i-th rule.

2.3.4. Defuzzification

The input to the defuzzification process is a fuzzy set obtained from the composition of the fuzzy rules, while the output is a crisp real number. Therefore, given a fuzzy set within a certain range, a specific crisp value must be derived as the output. In this study, the centroid method with a continuous variable is employed. The general equation for the continuous-variable centroid method is presented in Equation (4).

$$Z^* = \frac{\int_a^b Z\,\mu(z)\,dz}{\int_a^b \mu(z)\,dz} \quad (4)$$

where $Z^*$is the defuzzified value, $\mu(z)$represents the membership degree at a given point, and $z$de notes the domain value.

3. Design System

3.1. System Block Diagram

The design phase begins with the development of the system design and the physical structure of the device. The system design includes a flowchart, block diagram, and closed-loop control diagram. The block diagram consists of three main parts: input, processing, and output. The input section comprises a TMP36 temperature sensor and five push buttons as the user interface. The processing unit is an STM32F401RCT6 microcontroller. The output section includes a PWM control module, an ST7789 TFT LCD, and a heating element. The block diagram of the air heating system is shown in Figure 2.

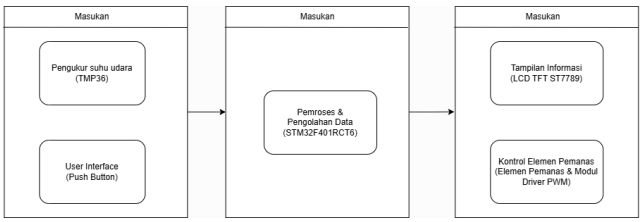


Figure 2. System Block Diagram

The functions of each block in the system shown in Fig. 3.2 are described as follows. The input unit consists of a TMP36 temperature sensor and push buttons. The TMP36 sensor measures the air temperature flowing through the heating element driven by a DC fan and serves as feedback for the air heating module. The push buttons function as a user interface for manual control, including selecting the control method, setting the temperature setpoint, starting and stopping the system, and resetting the system. The processing unit is an STM32F401RCT6 microcontroller, which processes input data from the temperature sensor and user interface, executes the control algorithm, and generates control signals. The output unit comprises a PWM driver module, a heating element, and an ST7789 TFT LCD. The PWM driver receives PWM signals from the microcontroller to regulate the power supplied to the heating element by adjusting the duty cycle, while the LCD displays system information to the user.

Following the system block diagram of the air heating module, a closed-loop control block diagram is presented as the main control mechanism of the system. The closed-loop control system employs two control methods, namely gain scheduling PID and Mamdani fuzzy logic, which are designed to maintain the air temperature at the desired setpoint by automatically adjusting the output power of the heating element. The closed-loop block diagrams of the gain scheduling PID and Mamdani fuzzy controllers are shown in Figure 3 and Figure 4, respectively.

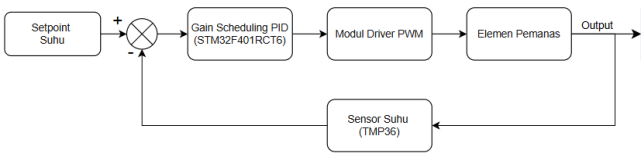


Figure 3. Closed-Loop Control Block Diagram of Gain Scheduling PID

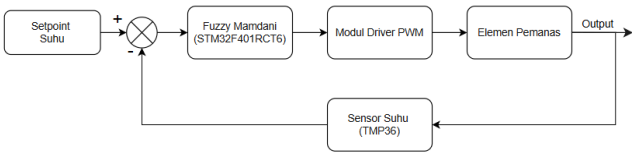


Figure 4. Closed-Loop Control Block Diagram of Mamdani Fuzzy Logic Controller

The closed-loop control block diagram in this study is divided into two parts, namely the gain scheduling PID controller and the Mamdani fuzzy logic controller. Figure 3 represents the working principle of the gain scheduling PID control. The temperature setpoint serves as the controlled input variable. After the setpoint is entered, it is processed by the STM32F401RCT6 microcontroller, which has been programmed with the gain scheduling PID control algorithm. The microcontroller then generates a PWM signal that is sent to the PWM driver module to regulate the heating power of the heating element. The resulting system output temperature is measured using a TMP36 temperature sensor. The measured air temperature is subsequently compared with the temperature setpoint to produce an error value between the setpoint and the actual system temperature. This error data is then processed again by the STM32 microcontroller. This process occurs continuously until the system produces an air temperature that matches the desired setpoint.

Meanwhile, Figure 4 illustrates the control system based on Mamdani fuzzy logic. The overall operational structure is generally similar to that of the gain scheduling PID system, involving the temperature setpoint as the input, measurement of the actual temperature using the TMP36 sensor, and PWM signal processing by the STM32 microcontroller to control the heating element. The main difference lies in the control logic employed, where the STM32 microcontroller is programmed using the Mamdani fuzzy logic approach. The system performs an inference process based on predefined fuzzy rules to determine the PWM output signal that is sent to the PWM driver module. This process is carried out continuously until the actual temperature approaches or reaches the specified temperature setpoint.

3.2. System Design

The hardware and software development stage is a crucial part of the air heating module implementation. At this stage, hardware development includes the mechanical and electronic components of the system, while software development involves programming tasks to operate the device based on the designed control logic. The development of both hardware and software can be divided into several stages as follows.

3.2.1. Hardware Design

The hardware of the air heating module consists of mechanical and electronic components assembled to generate and control air temperature according to the required specifications. The physical appearance and arrangement of the hardware components are shown in Figure 5, Figure 6, Figure 7, and Figure 8.

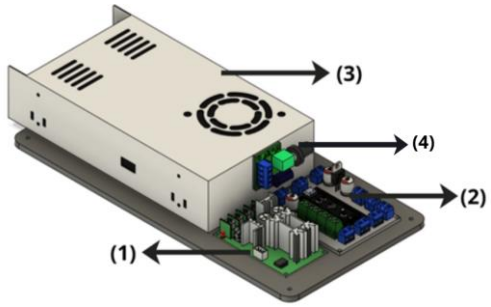


Figure 5. Internal Structure of the Air Heating Module

Description:
(1) PWM driver module
(2) STM32F401RCT6 microcontroller
(3) 12 V 20 A power supply
(4) PWM generation module

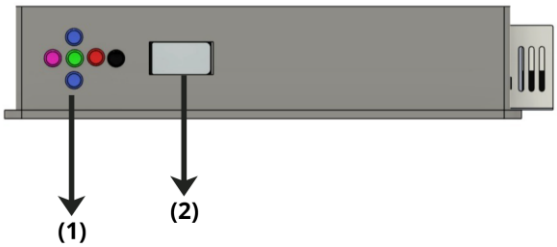


Figure 6. Rear View of the Air Heating Module

Description:
(1) User control push buttons
(2) ST7789 TFT LCD user display

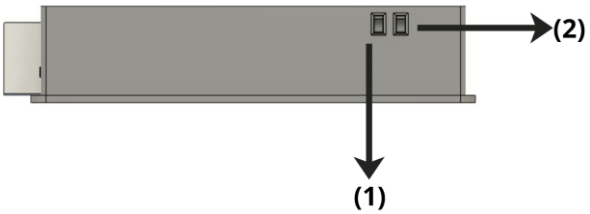


Figure 7. Front View of the Air Heating Module

Description:
(1) Switch for microcontroller supply voltage
(2) Switch for overall system supply voltage

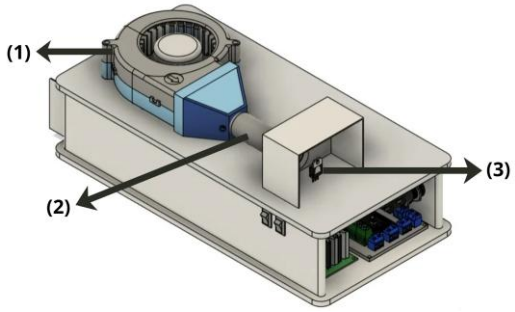


Figure 8. Top View of the Air Heating Module

Description:
(1) 12 V DC fan
(2) Heating element tube
(3) TMP36 temperature sensor

3.2.2. Software Design

The software serves as the core of the air heating system, functioning as the control unit that manages all system activities. Its primary tasks include executing given instructions, such as performing mathematical and logical operations, and managing the program flow in a structured manner. The processor used in this study is the STM32F401RCT6 microcontroller based on the ARM Cortex-M4 architecture. The microcontroller is responsible for reading and executing commands from all input and output components. The STM32 reads input signals from the TMP36 temperature sensor in the form of analog DC voltage and digital input signals (high or low) from the push buttons. In addition, the output devices connected to the microcontroller include an ST7789 TFT LCD as the user display, the heating element as the heat source of the air heating module, and a PWM driver module to control the pulse width applied to the heating element. The list of STM32 pin assignments used in this system is presented in Table 1.

Table 1. List of Pins Used for the Air Heating Module

Requirements	Description
PA0	Pin for TMP36 sensor reading
PA5	SCK pin for SPI communication
PA6	MISO pin for SPI communication
PA7	RESET pin for SPI communication
PA8	PWM output pin for controlling the heating element
PA9	TX pin for UART communication
PA10	RX pin for UART communication
PA11	Pin untuk push button STOP
PB0	RESET pin for SPI communication
PB1	DC pin for SPI communication
PB2	CS pin for SPI communication
PB12	Pin for MODE push button
PB13	Pin for UP push button
PB14	Pin for DOWN push button
PB15	Pin for OK push button

3.3. Gain Scheduling PID Design

The design of the gain scheduling PID controller was implemented using a program developed on the STM32CubeIDE platform. Gain scheduling was applied by varying the PID controller parameters (Kp, Ki, and Kd) based on the system operating conditions within a temperature range of 50 °C to 80 °C. This method aims to improve system response under different operating conditions, enabling the controller to adapt to setpoint changes and variations in thermal characteristics.

The gain scheduling PID controller is designed by assigning different PID parameter values (Kp, Ki, and Kd) for each temperature operating range. This approach allows the controller to adapt to variations in system dynamics and thermal characteristics. The gain scheduling PID parameters are presented in Table 2.

Table 2. Gain Scheduling PID Value

Setpoint	Kp	Ki	Kd
50 - <51	155	21	2
51 - <52	156	22	2,1
52 - <53	158	23	2,2
53 - <54	159	23	2,4
54 - <55	159,8	23,1	6
55 - <56	160	23,5	6
56 - <57	160,5	23,6	7,8
57 - <58	160,8	23,7	8,0
58 - <59	161	23,8	8,9
59 - <60	161,5	23,98	9
60 - <61	161,8	23,98	10
61 - <62	161,8	23,99	18,1
63 - <64	161,9	23,99	40,1
64 - <65	162	23,99	50,1
65 - <66	162,5	23,99	65,6
66 - <67	163	23,99	73
67 - <68	168	23,99	73,6
68 - <69	169	24	75,6
69 - <70	170	24	80,6
70 - <71	172	24.6	80,6
71 - <72	173	24,8	80,6
72 - <73	178	24,98	80,6
73 - <74	178	24,98	80,6
74 - <75	178	24,98	80,6
75 - <76	178	24,98	80,6
76 - <77	178	27	80,6
77 - <78	178	29	80,6
78 - <79	178	31	80,6
79 - <80	178	31	90,6
80	178	31	90,6

3.4. Fuzzy Mamdani Design

The Mamdani fuzzy control design for the air heating system is intended to produce a control signal that is more adaptive to temperature variations and thermal dynamics. The

Mamdani method is selected due to its intuitive rule structure and ease of adjustment based on system knowledge. The design process consists of four main stages as follows:

1. Fuzzification

At this stage, the input values in the form of error (the difference between the setpoint and the actual temperature) and delta error (the change in error between sampling times) are converted into fuzzy membership degrees. The input ranges are determined based on the characteristics of the system. The input error range is mapped within −12 °C to 50 °C using a combination of trapezoidal and triangular membership functions. The error fuzzification is illustrated in Figure 9.

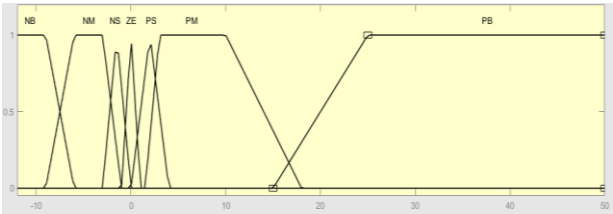


Figure 9. Design Membership Error

The delta error range is mapped within −6 °C to 6 °C using a combination of trapezoidal and triangular membership functions. The delta error fuzzification is illustrated in Figure 10.

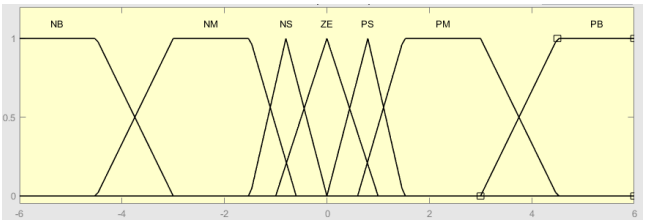


Figure 10. Design Membership Delta Error

2. Rule Base

The rule base consists of a set of "IF–THEN" rules that describe the relationship between the error and delta error conditions and the heating control signal. The rules are formulated based on knowledge of the air heating system behavior operating within the specified temperature range of 50 °C to 80 °C, enabling the controller to respond to real conditions such as slow temperature increase, rapid heating, or operation near the setpoint. The design of the Mamdani fuzzy rule base is presented in Table 3.

	Delta Error						
Error	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NB	NB	NB
NM	NB	NB	NB	NB	NB	NB	NB
NS	NB	NB	NS	NS	NS	NB	NB
ZE	PS	ZE	ZE	ZE	PS	ZE	ZE
PS	PB	PM	PM	PS	PM	PM	PB
PM	PM	PM	PB	PB	PS	PB	PB
PB	PB	PB	PB	PB	PB	PB	PB

Description:
NB = Negative Big
NM = Negative Medium
NS = Negative Small
ZE = Zero
PS = Positive Small
PM = Positive Medium
PB = Positive Big

3. Fuzzy Inference

At the fuzzy inference stage, the system performs a decision-making process based on all active rules in the fuzzy rule base table. Each rule is evaluated by measuring its activation level using the fuzzy logical AND operator, which is commonly represented by the minimum (MIN) operation, based on Eq. 3.

4. Defuzzification

The defuzzification process in this system employs the centroid (center of gravity) method to convert the fuzzy output into a numerical (crisp) PWM value. After the inference process produces activation degrees for each of the seven fuzzy output sets (NB, NM, NS, ZE, PS, PM, and PB), the system constructs a combined membership curve by scanning the entire PWM domain within the range of 0 to 4095.

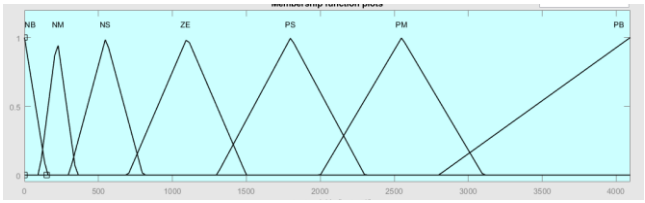


Figure 11. Design Membership PWM

4. Results and Discussion

4.1. Gain Scheduling PID Control Test Result

In the 50°C setpoint test, the graph in Figure 12 illustrates the temperature change over time until it reaches the setpoint value generated by the gain scheduling PID control.

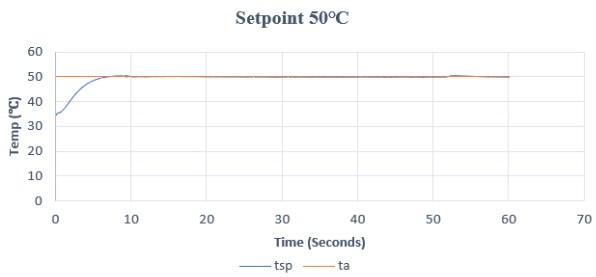


Figure 12. Gain scheduling PID Graph Setpoint 50°C

In the 65°C setpoint test, the graph in Figure 13 illustrates the temperature change over time until it reaches the setpoint value generated by the gain scheduling PID control.

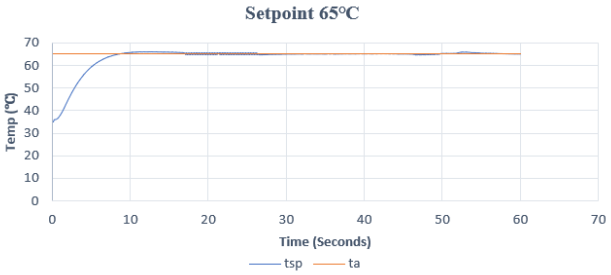


Figure 13. Gain scheduling PID Graph Setpoint 65°C

In the 80°C setpoint test, the graph in Figure 14 illustrates the temperature change over time until it reaches the setpoint value generated by the gain scheduling PID control.

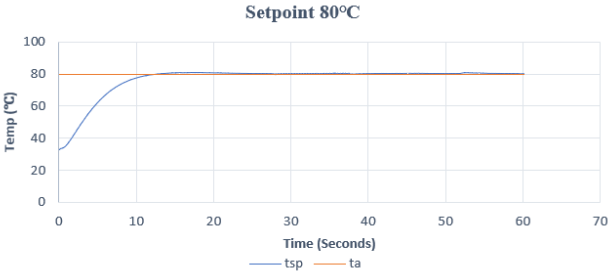


Figure 14. Gain scheduling PID Graph Setpoint 80°C

4.2. Fuzzy Mamdani Control Test Result

Figure 15 displays the temperature response over time for the 50°C setpoint test, showing the system's progression toward the target value under Fuzzy Mamdani control.

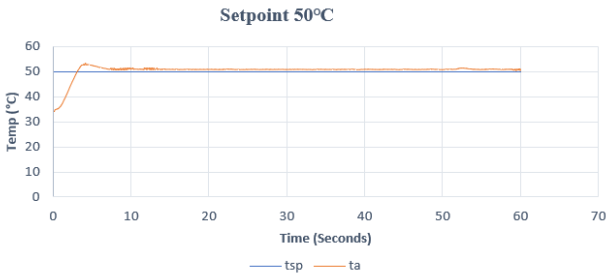


Figure 15. Fuzzy Mamdani Graph Setpoint 50°C

Figure 16 displays the temperature response over time for the 65°C setpoint test, showing the system's progression toward the target value under Fuzzy Mamdani control.

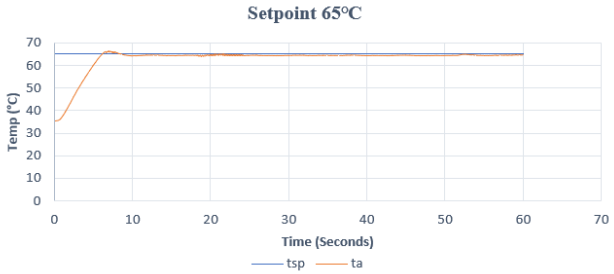


Figure 16. Fuzzy Mamdani Graph Setpoint 65°C

Figure 17 displays the temperature response over time for the 80°C setpoint test, showing the system's progression toward the target value under Fuzzy Mamdani control.

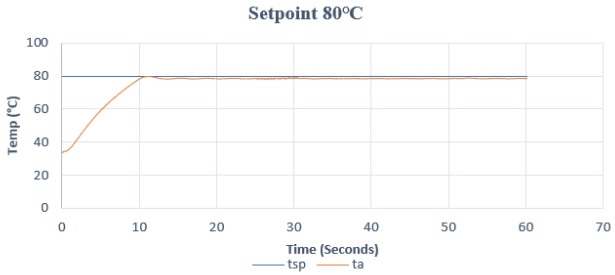
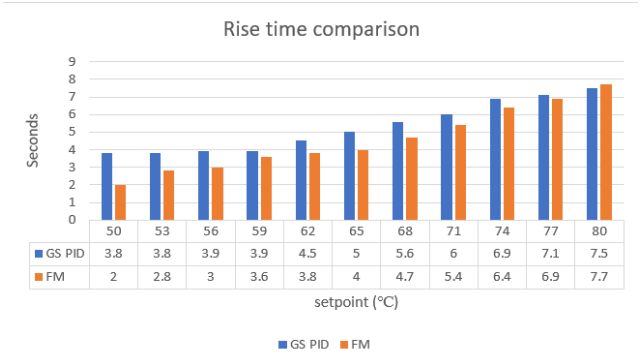


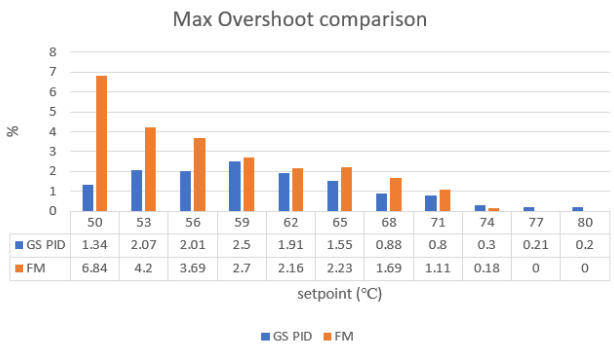
Figure 17. Fuzzy Mamdani Graph Setpoint 80°C

4.3. Gain Scheduling PID and Fuzzy Mamdani Control Comparison



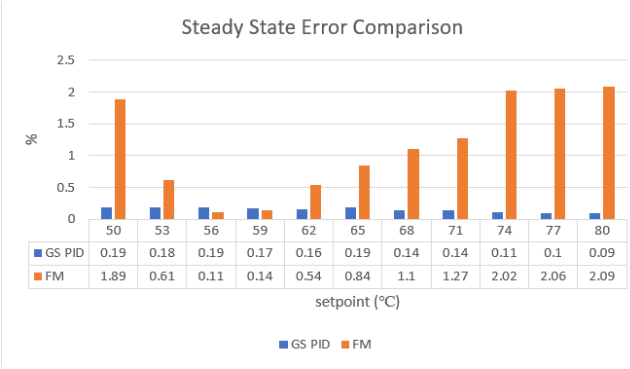
	50	53	56	59	62	65	68	71	74	77	80
GS PID	3.8	3.8	3.9	3.9	4.5	5	5.6	6	6.9	7.1	7.5
FM	2	2.8	3	3.6	3.8	4	4.7	5.4	6.4	6.9	7.7

Figure *18*. Rise time graph comparison



	50	53	56	59	62	65	68	71	74	77	80
GS PID	1.34	2.07	2.01	2.5	1.91	1.55	0.88	0.8	0.3	0.21	0.2
FM	6.84	4.2	3.69	2.7	2.16	2.23	1.69	1.11	0.18	0	0

Figure *19*. Max overshoot comparison



	50	53	56	59	62	65	68	71	74	77	80
GS PID	0.19	0.18	0.19	0.17	0.16	0.19	0.14	0.14	0.11	0.1	0.09
FM	1.89	0.61	0.11	0.14	0.54	0.84	1.1	1.27	2.02	2.06	2.09

Figure *20*. Steady state error comparison

Both control methods exhibit relatively consistent differences across the entire setpoint range. Regarding initial dynamics, Fuzzy Mamdani consistently demonstrates a faster rise time compared to Gain Scheduling (GS) PID. At a setpoint of 50°C, the Fuzzy controller achieves a rise time of 2 seconds, significantly shorter than the 3.8 seconds required by the GS PID. This pattern persists through to high setpoints, where at temperatures of 77–80°C, the Fuzzy rise time ranges between 6.9 and 7.7 seconds, while the GS PID is slightly slower, ranging from 7.1 to 7.5 seconds. This indicates that Fuzzy Mamdani provides a more responsive initial heating phase across all testing conditions.

However, the overshoot behavior of the two methods reveals distinct characteristics. GS PID produces a small and relatively stable overshoot, ranging from 0.3% to 1.55% across all setpoints, signifying a consistent response. Meanwhile, Fuzzy Mamdani shows a declining overshoot trend as the setpoint increases, starting at 6.84% at 50°C and dropping to 0% at 77°C and 80°C. Although this may appear superior at first glance, the zero overshoot at high setpoints actually indicates that the fuzzy system is unable to reach the target temperature, thus failing to surpass the setpoint altogether. Consequently, the absence of overshoot in the Fuzzy controller is not a result of high precision, but rather stems from poor tracking capability at higher temperatures.

4.4. Performance Comparison of Gain Scheduling PID and Fuzzy Mamdani Control Under Setpoint Changes

This section discusses the performance comparison between Gain Scheduling PID and Fuzzy Mamdani methods when the system undergoes setpoint changes. The analysis is conducted to evaluate how both controllers respond to heating dynamics, including characteristics such as rise time, overshoot, and steady-state error at each setpoint variation. This comparison aims to determine the more effective method for maintaining temperature stability and accuracy in the air heating module. The performance graphs for Gain Scheduling PID and Fuzzy Mamdani are presented in Figure 21 and Figure 22.

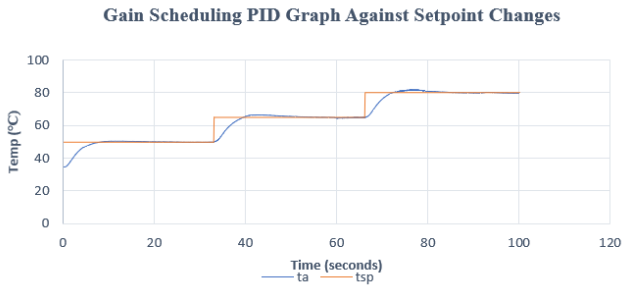


Figure *21*. Gain Scheduling PID graph against setpoint changes

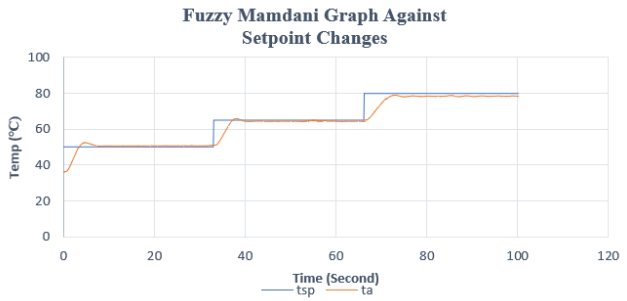


Figure *22*. Fuzzy Mamdani graph against setpoint changes

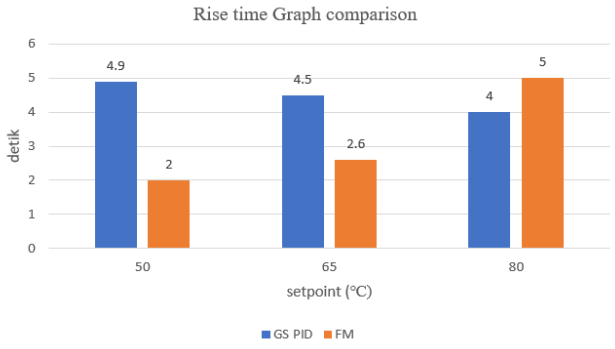


Figure *23*. Rise Time graph comparison

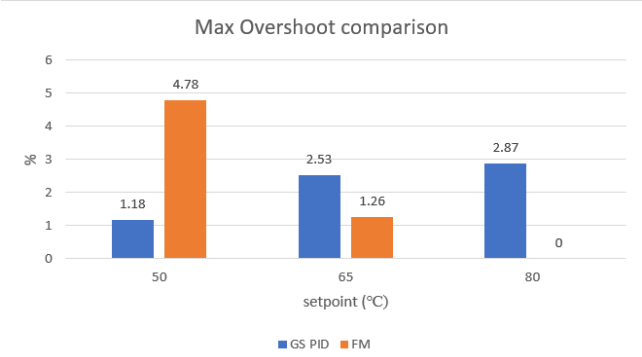


Figure *24*. Max Overshoot comparison

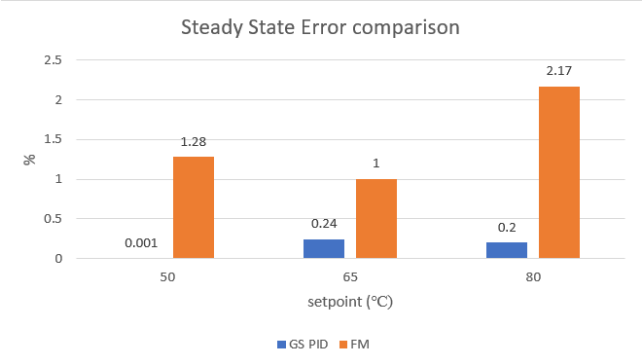


Figure *25*. Steady State Error comparison

5. Conclusion

Based on the research findings comparing controller performance on the air heating module, several key conclusions can be drawn as follows:

A. System Realization: Both control methods, Gain Scheduling PID and Fuzzy Mamdani, were successfully implemented on the system. Gain Scheduling PID operates by dividing the temperature range into several adaptive parameter zones, while Fuzzy Mamdani utilizes error and delta error inputs to produce smooth control without the need for an explicit mathematical model.

B. Response Characteristics: Gain Scheduling PID exhibits a stable and consistent response across all setpoints. Conversely, Fuzzy Mamdani has a faster initial response (**rise time**) than PID across the entire temperature range; however, it shows limitations in fully reaching the target temperature at medium to high setpoints.

C. Performance Parameters:

1. **Gain Scheduling PID:** Features a slower rise time but is superior in steady-state accuracy with small, stable **Steady-State Error (SSE)** and controlled overshoot.

2. **Fuzzy Mamdani:** Excels in rise time and exhibits a reduction in overshoot to 0% at high temperatures. However, this is caused by the system's inability to reach the setpoint, leading to a larger SSE.

3. **System Stability:** Both methods demonstrate good stability against repetitive setpoint changes. Gain Scheduling PID provides higher consistency in maintaining the temperature close to the setpoint compared to Fuzzy Mamdani.

4. **Implementation Complexity:** Gain Scheduling PID requires a more complex tuning process and extensive testing time since parameters must be defined for each temperature zone. In contrast, Fuzzy Mamdani is technically easier to implement but still requires significant expertise in designing **membership functions** and **rule bases** to achieve optimal results.

Reference

Allu, N., & Toding, A. (2020). Tuning with Ziegler Nichols Method for Design PID Controller At Rotate Speed DC Motor. *IOP Conference Series: Materials Science and Engineering*, *846*(1), 012046. https://doi.org/10.1088/1757-899X/846/1/012046

Amalia, S., Saputra, E., & Syukriansah, R. (n.d.). *Pemodelan Sistem Pengontrolan Suhu Ruangan Berbasis Logika Fuzzy Mamdani*. *10*(1), 2021. https://doi.org/10.21063/JTE.2021.31331006

Enriko, A., Putra, R. A., & Estananto. (2021). Automatic Temperature Control System on Smart Poultry Farm Using PID Method. *Green Intelligent Systems and Applications*, *1*(1), 37–43. https://doi.org/10.53623/gisa.v1i1.40

Ghazali, A. K., Aziz, N. A. A., & Ismail, W. Z. W. (2024). Hybrid Sliding Mode Control with Gain Scheduling Proportional Integral Derivative Controller for Water Tank System. *International Journal of Integrated Engineering*, *16*(3), 107–117. https://doi.org/10.30880/ijie.2024.16.03.011

Gonçalves, E. M., Pereira, N., Silva, M., Alvarenga, N., Ramos, A. C., Alegria, C., & Abreu, M. (2023). Influence of Air-Drying Conditions on Quality, Bioactive Composition and Sensorial Attributes of Sweet Potato Chips. *Foods*, *12*(6), 1198. https://doi.org/10.3390/foods12061198

Katsuhiko Ogata. (2010). *Modern Control Engineering - Fifth Edition*. Pearson Education.

Palani, H., & Karatas, A. (2023). Innovative Environmental Chamber Construction for Accurate Thermal Performance Evaluation of Building Envelopes in Varied Climates. *Buildings*, *13*(5), 1259. https://doi.org/10.3390/buildings13051259

Qurrata Ayun, A., Dwita, U., & Thoriq, M. (2025). *Penerapan Metode Fuzzy Mamdani Dalam Menentukan Suhu Optimal Pada Pendingin Ruangan (AC) Di Ruang Kelas* (Vol. 2, Issue 1). https://ojs.adzkia.ac.id/index.php/jtechρ46Journalhomepage:https://ojs.adzkia.ac.id/index.php/jtech

Rifky Adji Fadlani. (2024). Implementasi Metode PID dalam Sistem Kendali Temperatur Pada Ruang Produksi “Clean Room” Berbasis IOT. *Mars : Jurnal Teknik Mesin, Industri, Elektro Dan Ilmu Komputer*, *3*(1), 77–93. https://doi.org/10.61132/mars.v3i1.526

Salim, A. N., & Rahman, A. (2022). Implementasi Fuzzy-Mamdani untuk Pengendalian Suhu dan

Kekeruhan Air Aquascape Berbasis IoT. *Jurnal Algoritme*, *2*(2), 159–169.

S.Mathivanan, C. Kavirajan, M. Mahendraperumal, B.Meinathan, & B.Loganandh. (2017). Design and Analysis of Food Warmer . *International Journal of Engineering Research & Technology (IJERT)*, *5*(7), 1–3.

Sukandy, D.M., B. A. T., & Puspasari, S. (2008). *Penerapan Metode Fuzzy Mamdani Untuk Memprediksi Jumlah Produksi Minyak Sawit Berdasarkan Data Persediaan Dan Jumlah Permintaan (Studi Kasus PT Perkebunan Mitra Ogan Baturaja)*. 1–9.

Sunanto, S., Firdaus, R., & Makmur Setiawan Siregar. (2021). Implementasi Logika Fuzzy Mamdani Pada Kendali Suhu dan Kelembaban Ruang Server. *Jurnal CoSciTech (Computer Science and Information Technology)*, *2*(2), 128–136. https://doi.org/10.37859/coscitech.v2i2.3362