Comparative Performance Analysis PID and Fuzzy Logic Controllers for AC Motor Speed Control under Load

Sumantri Kurniawan Risandriya[1*], Muhamad Revi Ilhami[1], and Muhamad Ikhsan Azri[1]

[1]Electronics Engineering Technology Study Program, Electrical Engineering Department, Politeknik Negeri Batam, Batam, Indonesia

[*]Email: sumantri@polibatam.ac.id

Received on 10-03-2026 | Revised on 22-06-2026 | Accepted on 26-06-2026

***Abstract*—The cutting quality of materials such as acrylic and PCB is strongly influenced by spindle speed stability. Excessive speed in acrylic cutting can cause melting and poor surface finish, while low speed increases cutting force and produces burrs. In FR-4 PCB materials, unstable rotation may lead to defects such as delamination and reduced hole quality. Therefore, precise and stable AC motor speed control is essential. This study analyzes and compares the performance of PID and Fuzzy Logic Controllers (FLC) in controlling AC motor speed using a VFD and Arduino in a closed-loop system. Performance is evaluated based on overshoot, rise time, and steady-state ripple error under different setpoints and loads, including 3 mm acrylic and single-layer PCB. Results show that higher setpoints improve stability. The PID controller consistently provides lower overshoot, faster response, and smaller SSRE, especially at low speeds. Overall, PID is more effective, while FLC gives smoother response.**

Keywords: AC Motor, Fuzzy Logic Controller, Nonlinearity, PID Controller, Speed Control

I. INTRODUCTION

In AC motor speed control implemented in material cutting machines, the maximum rotational speed of the motor occurs when the machine operates without any load. The motor speed decreases significantly when a load is applied. However, good cutting performance requires a motor speed that is stable and does not exhibit high fluctuations. Adjusting the voltage supplied to the motor can change the motor speed in accordance with the voltage applied to the AC motor. Therefore, to regulate the AC motor speed while maintaining stable rotation in a cutting machine, an AC motor control system is required. Controllers commonly used PID controller and the Fuzzy Logic Controller [1].

In the acrylic cutting process, the combination of machining parameters consisting of spindle speed and feed rate determines the surface quality of the cut. Excessively high rotational speed causes an increase in temperature, resulting in melting of the acrylic and a rough surface, while excessively low speed increases cutting force and causes burrs on the cut edges. This indicates that an optimum rotational speed setting is required to achieve good cutting quality [2].

On FR-4-based PCB materials, the drilling and back-drilling processes show that increasing the spindle speed can improve chip removal, reduce cutting force, and reduce processing defects such as delamination and hole plugging. However, at low speeds, the cutting force increases and the mechanical interaction between the cutter and the material becomes rougher, resulting in a decrease in surface quality. These findings indicate that spindle speed instability has the potential to reduce PCB cutting quality [3].

Electrical equipment driven by AC motors is generally equipped with speed regulation that is limited to several discrete speed levels. The selection of AC motor drive speed strongly depends on the type of work and the material being processed. However, certain machining operations that demand optimal performance require the use of an appropriate drive speed. Consequently, such applications require an AC motor drive capable of providing a wide range of speed variations [4].

AC motor speed control can be implemented using various methods, such as frequency control, changing the number of pole pairs, external resistance control, armature input voltage regulation, vector control, voltage conversion, grid frequency conversion, and the use of electronic components. Induction motors do not always operate under full-load conditions, which can limit system performance. Speed control techniques that adjust the input voltage magnitude while synchronizing it with the load can influence the motor power factor, allowing these limitations to be utilized effectively [5].

II. METHOD

A. Block Diagram

The AC power source connects to a connector as the main supply. This AC voltage has two uses: it powers the Variable Frequency Drive (VFD) and supplies voltage to the control circuit through a 12 V adapter. The adapter's output voltage is stepped down with a step-down circuit to meet the operating needs of the Arduino Uno and other supporting modules. Arduino Uno 1 (sensor) acts as the data acquisition unit. It receives signals from the LM393 IR sensor to detect the rotational condition of the AC motor, including speed and number of rotations. The sensor data is then sent to Arduino Uno 2 (control) as the main input for the control system.

Arduino Uno 2 is the central controller for the system that receives motor speed data, and sets this up to be processed using Arduino UNO which reads Venti by sending a car speed setpoint via Serial Monitor (i.e., input values). Using this processed information, the Arduino creates a control signal in analog voltage output via the 0–10 V module to be input as speed reference to VFD. The output varies in amplitude, allowing a controlled operating frequency of the VFD, and hence controlling the speed of 220V AC motor as per figure postulated value [6].

The Arduino sends 0-10 V control signal and AC power supply to the VFD for precise speed control of an AC motor. The AC motor then spins in accordance with this control. A 20×4 LCD shows system status, measurement values, and operating parameters, allowing users to monitor the system performance in real-time.

In general, the system works by acquiring sensor data, processing it in an Arduino, controlling a VFD through an analog signal and displaying the measurement values in LCD and Serial Monitor, as shown in the image below.

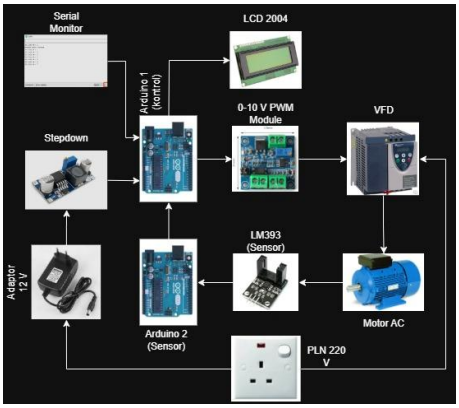

Figure 1. Block Diagram

B. Fuzzy Logic Controller

The concept of fuzzy theory was first introduced by L.A. Zadeh in 1965 in the form of Fuzzy Set theory. In classical sets, membership values are crisp, meaning that an element either belongs to the set (with a value of 1) or does not belong to it (with a value of 0). In contrast, for fuzzy sets, the membership of an element within a universe of discourse is ambiguous. The constituent elements possess different levels or values of membership rather than a binary classification [7].

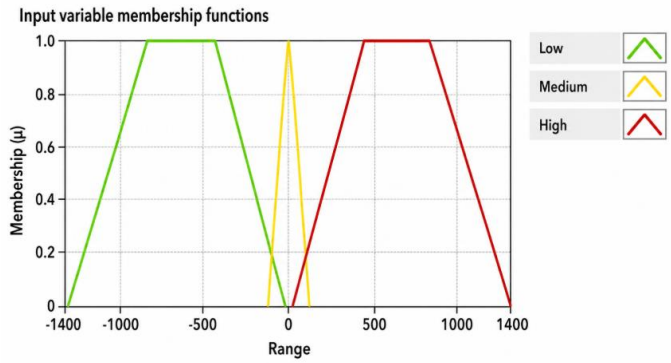

Figure 2. Error Membership

The figure above shows how errors are categorized into three levels (low, medium, and high), which the FLC uses to determine control actions. A negative reading implies that the current velocity falls short of the target value, whereas a value of zero signifies that the real-time speed aligns closely with the reference point. Conversely, a positive result indicates that the measured speed has surpassed the designated set point.

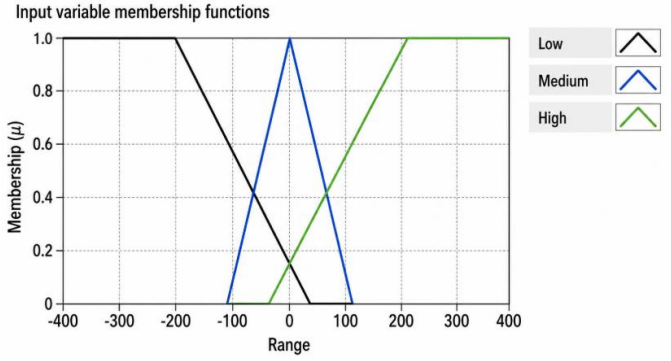

Figure 3. Delta Error Membership

The figure above shows the membership for delta error. The change in error, or delta error, is computed by subtracting the preceding error from the current one. Identical to the prior error classification, the membership function for delta error is divided into three tiers: low, medium, and high. In this setup, a negative deviation falls into the low tier, an error value nearing zero is assigned to the medium tier, and a positive discrepancy signifies the high tier.

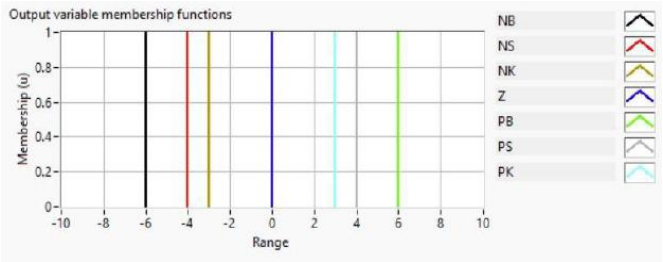

Figure 4. Single Tune

The figure above shows the membership functions of the output variables in a Fuzzy Logic Controller (FLC) system, which are used to determine the control action. There are several linguistic categories: NB (Negative Big), NS (Negative Small), NK (Negative Small), Z (Zero), PK

(Positive Small), PS (Positive Small), and PB (Positive Big). Each vertical line indicates a specific output value that will be selected based on the results of the fuzzy rules (rule base). Negative values (on the left) indicate a control action to reduce the system output, while positive values (on the right) indicate an action to increase the output. The midpoint (Z) indicates a condition where there is no change or the system is already stable.

The defuzzification process is performed using the Weighted Average (WA) method. This method was chosen because it is computationally efficient and well-suited for implementation on an Arduino Uno. The defuzzification result is a PWM value that is then sent to the VFD to control the operating frequency of the AC motor. With this FLC design, the system is able to produce a more stable speed response, reduce overshoot, and maintain the motor speed close to the setpoint even when load changes occur during operation.

TABLE I
DELTA ERROR MEMBERSHIP

Error Delta	EN (Error Negative)	EZ (Error Zero)	EP (Error Positive)
DN (Delta Negative)	PK	PS	PB
DZ (Delta Zero)	NS	Z	PS
DP (Delta Positive)	NB	NK	PK

PB: Positive Big
PS: Positive Middle
PK: Positive Small
Z: Zero
NK: Negative Small
NS: Negative Middle
NB: Negative Big

The table above illustrates the rule base in a Fuzzy Logic Controller (FLC) system that links two input variables—Error and Delta Error (Δerror)—to generate a control output. This determines the magnitude of the control action (such as a change in motor speed or VFD frequency) based on the error condition and the rate of change of the error, enabling the system to reach the setpoint more smoothly and adaptively.

The rule base indicates that when the error is negative and the delta error is also negative, the system produces a Small Positive (SP) output. This is intended to reduce the tendency for excessive deceleration. When the error approaches zero and the delta error is zero, the output is Zero (Z) because the system is already near the setpoint. Conversely, when the error is positive and the delta error is also positive, the system produces a Small Positive (SP) output or an appropriate corrective action to maintain system stability.

C. Flowchart of the Fuzzy Logic Controller

The flow chart above depicts the overall operation of a fuzzy logic-based motor control system. The flow chart shows in figure 6 that the motor control system and the hardware components must be initialized first. Next, the LCD will display a message that says “Ready” indicating that the system has completed its initialization and is now ready to accept an input from the keypad. Following this, the keypad is used to input a desired setpoint for the motor speed to the motor controller. After the desired speed has been determined, the system will use an infrared (IR) sensor (LM393) to measure the current motor speed to calculate the system’s current motor speed. After measuring the motor speed, the system will calculate the error (i.e., the difference between the setpoint and current motor speed) as well as the delta error (i.e., the change in error from the previous error value). Following this, the fuzzy logic system will fuzzify both of the calculated errors into fuzzy sets. Lastly, the system will defuzzify both of the fuzzy sets into clear output signals[8].

The system reads the current motor speed using the IR sensor (LM393). After getting the speed measurement, it calculates the error as the difference between the setpoint and the current motor speed. It also calculates the delta error as the change in error from the previous value. The next step is to convert the error and delta error values into fuzzy sets. This is followed by defuzzification, which turns the fuzzy sets back into a precise value.

The output value of the PWM is determined by the PWM's last output value, plus the defuzzification value. The speed of the motor will be controlled by the PWM value being sent to the VFD to adjust it. Also, the speed of the motor and the error value are displayed in an LCD for monitoring. Then the system checks to see if the STOP button has been pressed. If the STOP button has not been pressed, the system will once again go back to read the speed from the IR sensor (LM393) and repeats the control cycle. If the STOP button has been pressed, the motor will be turned off and an LCD message “Motor Off” will be displayed to indicate the control will no longer continue.

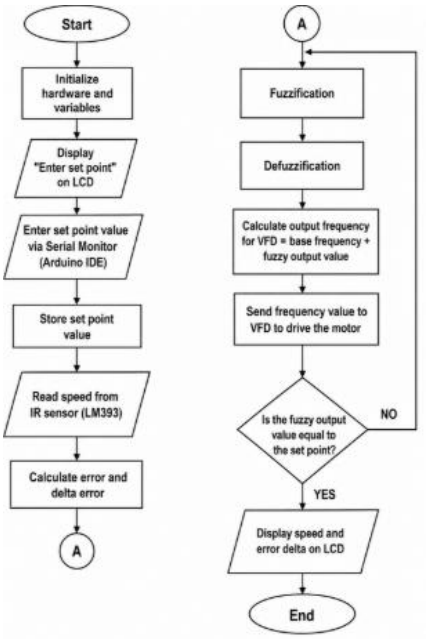

Figure 5. FLC Operation System

D. PID Kontroller

Within a fuzzy set, the constituent elements possess different levels or values of membership rather than a binary classification, so that the sistem final result is precise and moves closer to the predetermined target parameters. In a closed-loop control system, corrective action depend heavily on the system own output. The main characteristic of this system is the use of a feedback path to measure the difference between the desired target and the actual conditions in the field. This difference, or error value, is then processed by the control element as the basis for minimizing the error, so that system performance remains stable and the resulting output always matches the reference value [9].

E. PID Tuning

In designing a PID control system, it is necessary to adjust the P, I, and D parameters so that the system output response to a given input meets the desired performance. PID tuning can also be based on the Ziegler–Nichols tuning method. This method aims to achieve a maximum overshoot (MO) of 25% in response to a step input [10].

F. Flowchart PID Kontroller

Operation of a PID Control System starts with: Initialization of the system; Selection of an AC motor speed (RPM), from available choices of 400 through 1200 RPMs; Entry of the Kp, Kd, and Ki values obtained through the Ziegler–Nichols method. The LM393 IR speed sensor reads RPM data being produced by the AC motor providing the error which will then be computed using the Kp, Kd and Ki values. The LM393 output speed reading from the AC motor is then matched against the preselected setpoint; If the setpoint and speed reading match, the system holds in that position (remains in that state); If the two values do not match, the procedure starts over again, so that the system will be held in the new position based upon the most recent speed data. This continues iteratively.

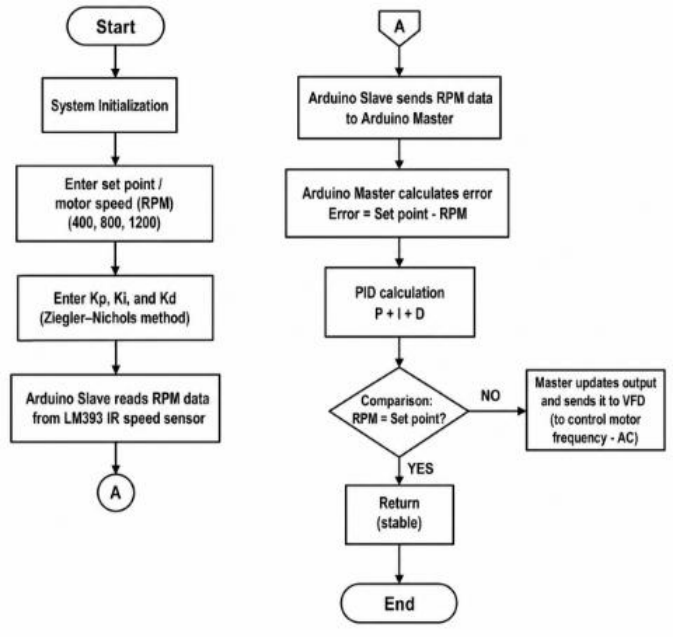

Figure 6. PID Kontroller Operation System

G. Time Domain Analisys

Time-domain analysis is a primary approach for evaluating transient response and system stability, especially for systems subjected to step inputs. Time-domain parameters are used to determine the response speed and the quality of system stability. This approach is widely applied in electric motor control research because it can represent the actual operating conditions of the system more realistically than analyses based purely on linear models [11].

H. Nonlinearity of AC Motors and VFDs

Three-phase induction motors (AC motors) controlled via a Variable Frequency Drive (VFD) are regarded as complex and nonlinear control systems. As a result, conventional linear control approaches such as PID have limitations in guaranteeing optimal performance over the entire operating range [12].

III. RESULT AND DISCUSSION

A. Testing

The process of determining PID parameters using the Ziegler-Nichols method begins by setting the Ki and Kd values to 0, so that the system uses only proportional control. The Kp value is then gradually increased while observing the system's response until the system reaches a state of sustained oscillation. The Kp value at this point is referred to as the ultimate gain (Ku), and the oscillation period is referred to as the ultimate period (Pu).

After obtaining the Ku and Pu values, they are calculated using the formula shown in the table below:

TABLE II
PARAMETER PID ZIEGLERS-NICHOLS

Type of Controller	Kp	Ki	Kd
P	0,5 Kcr	∞	0
PI	0,45 Kcr	1 / 1,1 Pcr	0
PID	0,6 Kcr	0,5 Pcr	0,125 Pcr

Changes in the values of kp, ki, and kd at each setpoint (target speed) actually occur due to the inherent nature of the AC motor and its load, which constantly fluctuate under various operating conditions. Imagine the motor is set to a low speed (400 RPM). Under these conditions, the motor tends to respond more slowly and is more prone to "slipping" (greater motor slip). Therefore, we need a PID parameter tuning that is extra sensitive so the motor responds quickly to small errors, yet remains stable—meaning it doesn't oscillate excessively—when the motor is forced to accelerate to medium (800 RPM) and high (1200 RPM) speeds. As the rotational speed increases, the torque and friction load during automatic cutting also change drastically. It is this change in the motor's behavior that causes the critical threshold values of Ku and Pu to shift. That is why the Ziegler-Nichols calculation formula produces different PID values for each target speed. The goal is to ensure that, regardless of the speed at which the motor is started, the system can still respond instantly, avoid overshooting (low overshoot), and ultimately achieve a speed

that is precisely on target (low steady-state error).

The purpose of testing was to compare the performance of the PID controller to that of the Fuzzy Logic Controller when controlling the speed of an AC motor via a Variable Frequency Drive (VFD). Two different loads were tested: an acrylic load that was 3 mm thick and a single-layer PCB load. Speed setpoints for both experimental runs were set to three different RPM values (400, 800, and 1200). The results from both tests were analyzed using three parameters: percent overshoot (PO), rise time (RT), and steady-state error (SSE). The Fuzzy Logic Controller tests were also conducted using two different experimental runs.

1) Testing the effects of Kp, Kd, and Ki values on steady-state error, rise time, and overshoot for the PID controller under load conditions.

TABLE III
PID KONTROLLER TESTING

Applied load	Set point (RPM)	Parameter PID			PO	RT	SSRE
		Kp	Kd	Ki			
1 (3 mm Thick Acrylic)	400	2,1	0,35	0,05	19%	0,06 s	16,88%
	800	1,2	2,34	0,15	6,48%	0,18 s	6,81%
	1200	2,8	4,5	0,45	4,8%	0,5 s	5,33%
2 (PCB Single Layer)	400	2,1	0,35	0,05	15%	0,06 s	14,8%
	800	1,2	2,34	0,15	6,5%	0,16 s	6,18%
	1200	2,8	4,5	0,45	3,5%	0,36 s	4,54%

The Table II summarizes the results of PID controller tests using different loads (3 mm thick acrylic and single-layer PCB). Three setpoints (400, 800, and 1200 RPM) were utilized to provide data on the following performance metrics: Overshoot, Rise Time, and Steady State Ripple Error. The PID controller settings (Kp, Ki, Kd) were adjusted for each setpoint.

2) *Testing of AC Motor RPM Measurement Using a Fuzzy Logic Controller Under Load.*

TABLE IV
COMPARATIVE TESTING OF THE PID CONTROLLER AND FLC UNDERLOAD

Applied load	Set point (RPM)	PID Controller			Fuzzy Logic Controller		
		O (%)	RS (s)	SSRE (%)	O (%)	RS (s)	SSRE (%)
1 (3 mm Thick Acrylic)	400	19	0,06	16,88	30	0,3	28
	800	6,48	0,18	6,81	9,5	0,61	13,43
	1200	4,8	0,5	5,33	9,5	1,02	11,54
2 (PCB Single Layer)	400	15	0,06	14,8	20	0,51	24
	800	6,5	0,16	6,18	8,25	0,68	10,93
	1200	3,5	0,36	4,54	7,5	1,01	7,12

The results from testing the speed response of an AC motor with variable frequency drive (VFD) under two loads (3 mm acrylic and single-layer PCB) are shown as a table 2. Three reference speeds (400, 800 or 1200 RPM) were used for each load. Each of these loads was tested with two trials (Trial 1 and Trial 2), in order to test the system's consistency. The parameters analyzed were: rise time (RT), delay time (D), peak time (P), overshoot (O), and steady state ripple error (SSRE).

3) Comparative Testing of the PID Controller and Fuzzy Logic Controller Under Load.

TABLE V
COMPARATIVE TESTING OF THE PID CONTROLLER AND FLC UNDERLOAD

Applied load	Set point (RPM)	PID Controller			Fuzzy Logic Controller		
		O (%)	RS (s)	SSRE (%)	O (%)	RS (s)	SSRE (%)
1 (3 mm Thick Acrylic)	400	19	0,06	16,88	30	0,3	28
	800	6,48	0,18	6,81	9,5	0,61	13,43
	1200	4,8	0,5	5,33	9,5	1,02	11,54
2 (PCB Single Layer)	400	15	0,06	14,8	20	0,51	24
	800	6,5	0,16	6,18	8,25	0,68	10,93
	1200	3,5	0,36	4,54	7,5	1,01	7,12

Based on the test results shown in Table V, it can be observed that both the PID controller and the Fuzzy Logic Controller are capable of controlling the AC motor speed with a fast initial response, as indicated by the relatively small rise time values across all setpoint variations.

Calculation to Determine the Rise Time:

$$Tr = (t90\% - t10\%) \times Ts \quad (1)$$

One example of a calculation using PID data with a 1200 RPM setpoint.

$$T90\% = 90\% \times 1200 = 1080$$

$$T10\% = 10\% \times 1200 = 120$$

$$Tr = (40 - 15) \times 0{,}02$$

$$Tr = 0{,}3$$

Based on the example above, t10% is obtained at 15 and t90% at 40, which refer to the sampling indices in the Serial Monitor data. This means that the 10% value of the setpoint occurs at sampling index 15, while the 90% value of the setpoint occurs at sampling index 40. The value 0,02, which represents Ts, is the PID response time.

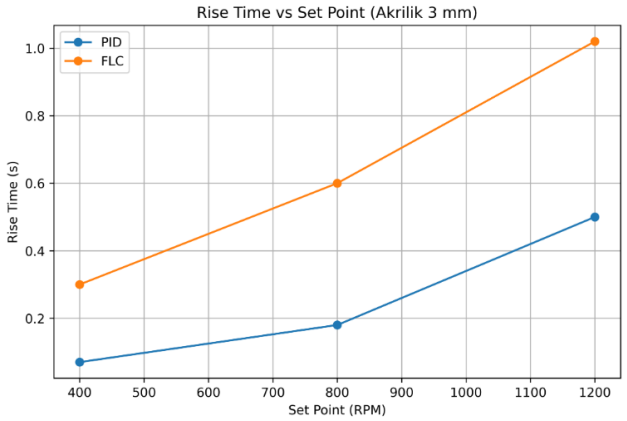

Figure 7. Rise time graph under acrylic load

The figure 7 shows how much time it takes for a machine to reach its original reference speed after you change its motor speed setpoint. This completion timing is referred to as "rise time." The results show that every time you make a change to your motor speed setpoint, the rise time should be the same regardless of what control method you're using. However, in this example, whenever the motor speed setpoint was changed to 400 RPM, the PID controller produced a rise time of about 0,06 seconds, compared to an FLC rise time of about 0,30 seconds. Therefore, the initial response is quicker with the use of a PID controller.

As the setpoint increases to 800 RPM and 1200 RPM, both controllers experience an increased rise time, although the increase is much larger for the FLC than the PID Controller. The PID rise time at the 1200 RPM setpoint is about 0,5 s; the FLC rise time is more than 1 second at the same setpoint. These increased rise times indicate that the higher the setpoint, the longer it takes the system to reach the desired speed, especially for the conservative FLC.

The distinct characteristics show that a PID Controller is a more aggressive and responsive control method for accelerating the initial response of the system; however, the Fuzzy Logic Controller (FLC) is less aggressive and responds smoothly but with lower speed than PID, particularly at higher load settings. Furthermore, the nonlinear characteristics associated with the AC motor/VFD when operating under conditions of an acrylic load contribute to the sensitivity of the FLC relative to the PID Controller, and their corresponding operation is affected by differences in operating environment.

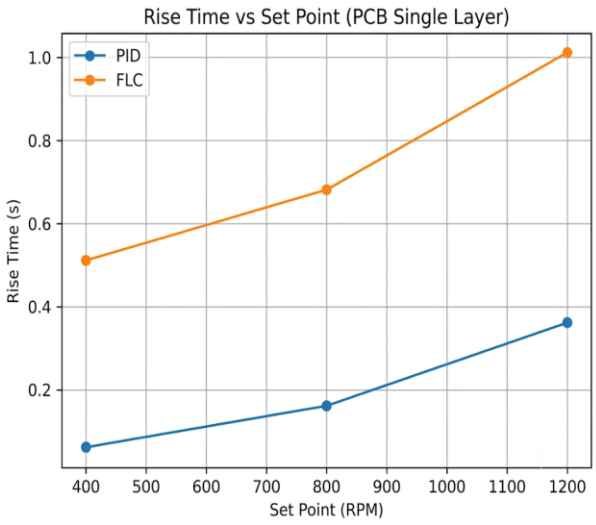

Figure 8. Rise time graph under PCB load

In comparing this graph to the previous one (with an acrylic load of 3mm) it appears that the single-layer PCB load considerably shortened the rise time of the PID controller (precluding the control system response) while providing greater mechanical damping (i.e. a better response system) than did either of the other PCBs. Finally, the fundamental characteristics (i.e. aggressiveness, setpoint speed) between both control types are consistent with each other regardless of the type of PCB that is being tested: the PID has a greater response to commands and reaches the setpoint faster; the FLC has a relatively kind of response but reaches the setpoint slower.

Comparing these two graphs confirms that the controller's characteristics have a greater impact on the rise time differences than the load variations that only affect the rise time magnitude and do not alter the overall response trend of the system; thus confirming this statement is true [13].

In terms of overshoot, the PID controller generally produces smaller overshoot values than the FLC, especially at medium and high setpoints. The formula used to calculate overshoot is as follows [14]:

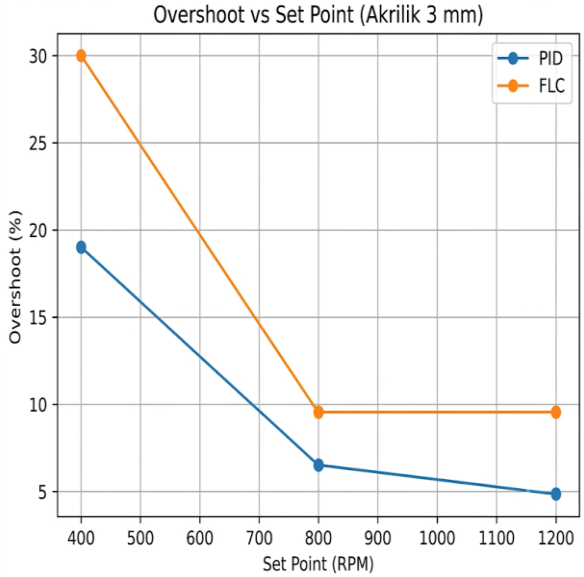

Figure 9. Overshoot Graph under Acrylic Load

$$PO = \frac{Ymax - Yss}{Yss} \; x \; 100\% \quad (2)$$

Example of a Calculation Using PID Data with a 1200 RPM Setpoint:

$$PO = \frac{1258 - 1200}{1200} \; x \; 100\%$$

$$PO = 4{,}8\ \%$$

This graph on figure 9 depicts how much an increase in the setpoint will increase the amount of overshoot of the two types of control used in this system (a PID controller and a Fuzzy Logic Controller (FLC)) as you increase from a low speed to a high speed. At 400 RPM, the amount of overshoot is significant from both controllers, although the FLC is much greater than the PID at about 30% versus about 19%. Therefore, although the system will still be highly influenced by the nonlinearity of the AC motor and the characteristics of the VFD at low speed, there will be an even greater degree of difficulty in controlling the AGV at low speed due to the aggressive nature of the response to changes in the input.

The overshoot experienced at the setpoint of 800RPM has dropped considerably for the PID and FLC Controllers indicating that the system is becoming more stable and linear. At 800RPM, the PID controller's overshoot is approximately 6,48% with respect to the new setpoint while the FLC's maximum overshoot is about 9,5%. When the setpoint is raised to 1200 RPM, both the PID and FLC's maximum overshoots decrease further and show a greater degree of

stability, with the PID's overshoot at an approximate value of 4,8% and the FLC's remaining at approximately 9,5%.

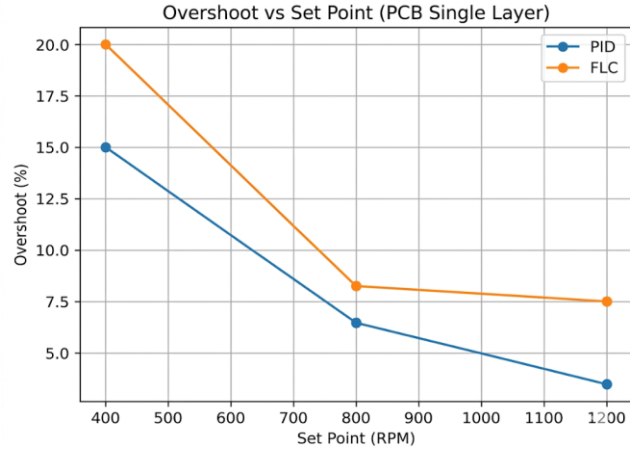

Figure 10. Overshoot Graph under PCB Load

The overall results from the graph show that for all setpoint changes, surpassing the PID controller is better than with the FLC controller for eliminating overshoot; especially at lower speeds. Increasing your setpoint reduces overshoot, showing that system non-linearity becomes a less influential factor. Additionally, the difference between both controllers indicates that while a FLC response is more aggressive (and less damped) than the PID response, their effect on damping is similar under a load of 3 mm using acrylic material.

The comparison of the 3 mm acrylic load versus the single layer PCB load shows a lower level of overshoot in the single layer PCB for all setpoint variations for both PID and FLC controllers. The PCB load provides greater mechanical damping therefore causing less overshoot to occur at the initial response peak compared to the 3 mm acrylic load. However, the general trend of overshoot variation for both loads is consistent; the lowest speed results in the highest degree of overshoot and the degree of overshoot decreases as the speed of the setpoint to be reached increases.

Additionally, both graphs demonstrate that the PID controller has a lower level of overshoot than the FLC, supporting the superiority of the PID controller in damping. However, these comparisons show that the loads have an influence on the amount of overshoot produced. Furthermore, the overall response characteristics of the system are primarily determined by the type of control system.

In terms of steady-state error, the PID controller consistently yields smaller SSE values than the FLC across all setpoint variations and load conditions. The formula used to calculate the steady-state error is as follows:

$$\text{SSRE} = \frac{Ymax - Ymin}{2SP}\ x\ 100\% \quad (3)$$

The output of the motor speed in the developed system will be non-constant (i.e., have some small fluctuations), even though it is at steady state with regards to the output of the AC motor, VFD, and the resolution of the sensor used to measure it). As such, the conventional method of calculating steady-state error does not accurately provide a representation of the situation as it currently is today. To provide a for more precise definition on the stability of the output in steady state, an additional method called "steady-state ripple error" (SSRE) has been developed. The calculation of SSRE is accomplished through verifying the amplitude of the fluctuations of the output with respect to the setpoint [15].

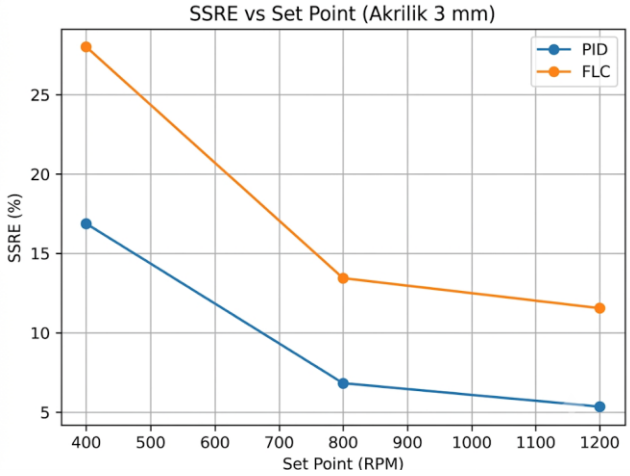

Figure 11. SSRE Graph under Acrylic Load

According to the graph in figure 11, the PID controller outperformed the fuzzy logic controller (FLC) in controlling speed variation under steady-state conditions for every setpoint. The SSRE continued to decrease as the setpoint increased. This implies that non-linearities in the system play a larger role at low speeds than they do at higher speeds and, therefore, the system becomes progressively more stable at high speeds.

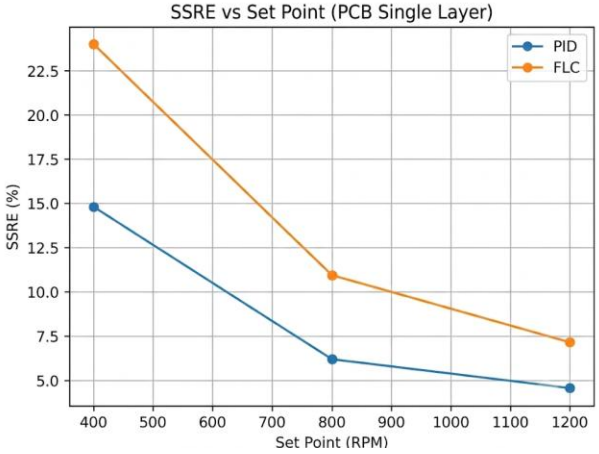

Figure 12. SSRE Graph under PCB Load

As shown on the graph in Figure 12, motor speed fluctuations vary based on SSREs and are related to PID and Fuzzy Logic Controllers' (FLC) performance with respect to a constant SSRE. When looking at a SSRE of 400 RPM, both PID and FLC have a high amount of fluctuation in their performance, while at SSREs of approximately 23%, FLCs have a large amount of speed fluctuation differences at low speed; as compared to PID operating at approximately 14% SSRE. Therefore, at low speeds, there is still significant motor speed fluctuation under loaded operating conditions.

With the increase in setpoint from 800 to 1200 RPM, the SSRE for both controllers shows dramatic decreases.

Specifically, the PID controller SSRE value decreases to approximately 6% and the FLC decreases between 10% and 11%. The PID controller SSRE value reaches a minimum around 4% to 5% and the FLC SSRE value reaches a minimum around 7% to 8% at 1200 RPM indicating that steady state speed stability improves with increased speed.

Comparing the data in Figure 16, it can be observed that a single-layer PCB load has lower SSRE values than either type of load (PID or FLC) for all setpoints. A lower SSRE value indicates increased mechanical damping of the PCB load, resulting in reduced amplitude of steady-state speed variations. Notably, both load configurations exhibit a similar pattern of variation in SSRE data, whereby SSRE is maximized at low speeds and decreases as setpoint increases. Lastly, SSRE data for both types of controllers show that a PID controller produces consistently lower SSRE than FLC, further validating the superior performance of the PID controller relative to FLC for maintaining continuous stable speed of the motor in steady state operation.

IV. CONCLUSION

Based on the test data, the response to changes in AC motor speed shows that the PID controller has a faster rise time and smaller overshoot compared to the FLC under nearly all test conditions. This indicates that the PID controller provides a faster and more damped response to setpoint changes. Meanwhile, the FLC tends to provide a slower response and produces greater fluctuations, especially at low speeds. Overall, PID exhibits better transient response characteristics, while FLC provides a smoother response but is less optimal under nonlinear conditions and at low speeds. This study was conducted using motor speeds of 400, 800, and 1200 rpm. Some improvements that need to be made include developing hybrid controllers, such as Fuzzy-PID or gain-scheduling PID, to combine the fast response of PID with the adaptive stability of FLC in the face of system nonlinearities. In carrying out and completing this final project, there were certainly some shortcomings and limitations, both in the system and in the equipment that was developed. To address this issue, the following corrective measures need to be taken: Develop hybrid controllers, such as Fuzzy-PID or gain-scheduling PID, to combine the fast response of PID with the adaptive stability of FLC in the face of system nonlinearities, the fuzzy rules (rule base) and membership functions need to be refined so that the FLC's response is not too aggressive in nonlinear regions, dding a digital filter or a velocity observer can help reduce noise, thereby making the SSRE evaluation more accurate.

REFERENCES

[1] Y. T. K. Priyanto, A. R. Utami, M. R. Dewanto, D. S. Santaki, dan D. Wulandari, "3 Phase Synchronous Motor Speed Control System Using PID Control," *J. Sistim Inf. dan Teknol.*, vol. 4, hal. 180–185, 2022, doi: 10.37034/jsisfotek.v4i4.149.

[2] S. Roughness, "CNC Milling of Medical-Grade PMMA : Optimization of Material Removal Rate and Surface Roughness," vol. 12, no. 1, hal. 1–15, 2022, doi: 10.4018/IJMMME.293226.

[3] Z. Chen, "Chip removal mechanism and hole plugging in back-drilling of high-speed printed circuit board," hal. 0–25, 2024.

[4] A. M. Prasetia, T. Hariyanto, A. Huda, L. Sartika, dan F. Fitriani, "Monitoring dan Kendali Kecepatan Motor Universal Menggunkan Human Machine Interface (HMI)," *Elektr. Borneo*, vol. 9, no. 1, hal. 28–35, Apr 2023, doi: 10.35334/eb.v9i1.3502.

[5] Endryansyah, P. wanarti Rusimanto, dan Fendi achmad, "Analisis Penggunaan Alat Pengatur Kecepatan Motor Ac Satu Phase Menggunakan Bidirectional Triode Thytistor (Triac)," *J. Tek. Elektro*, vol. 10, hal. 315–323, 2021.

[6] Deni Irawan, Prihadi Murdiyat, dan Rusdiansyah, "Variable Frequency Drive (VFD) Berbasis Arduino Mega 2560 Sebagai Pengendali Motor Induksi 3 Fase," *PoliGrid*, vol. 4, no. 2, hal. 52–61, 2023, doi: 10.46964/poligrid.v4i2.30.

[7] R. Saatchi, "Fuzzy Logic Concepts, Developments and Implementation," 2024.

[8] W. Raza, D. Adzikya, S. Mehmood, dan S. R. Wasti, "Fuzzy Logic Speed Regulator for D . C . Motor Tuning," vol. 8, no. 1, hal. 36–49, 2024.

[9] J. Schöning, "Safe and Trustful AI for Closed-Loop Control Systems," vol. 12, no. 16, 2023.

[10] S. Bassi, E. Gbenga, A. Abidemi, D. Opeoluwa, dan B. Mohammed, "Heliyon Metaheuristic algorithms for PID controller parameters tuning : review , approaches and open problems," *Heliyon*, vol. 8, no. January, hal. e09399, 2022, doi: 10.1016/j.heliyon.2022.e09399.

[11] Q. Wu, L. Fellow, Y. Lin, C. Hong, dan Y. Su, "Transient Stability Analysis of Large-scale Power Systems : A Survey," vol. 9, no. 4, hal. 1284–1300, 2023, doi: 10.17775/CSEEJPES.2022.07110.

[12] T. Khurshaid dan S. Kamel, "Three Phase Induction Motor Drive : A Systematic Review on Dynamic Modeling, Parameter Estimation, and Control Schemes," vol. 15, no. 21, 2022.

[13] M. B. R. Maulana1, D. Dewatama2, dan Mila Fauziyah3, "Kontrol Kecepatan Motor Induksi 3 Fasa Berbasis Arduino Mega 2560 Pada Lift 4 Lantai," vol. 2, no. 6, hal. 71–80, 2024.

[14] R. Budiarto Hadiprakoso dan N. Qomariasih, "Deteksi Masker Wajah Menggunakan Deep Transfer Learning Dan Augmentasi Gambar," *JIKO (Jurnal Inform. dan Komputer)*, vol. 5, no. 1, hal. 12–18, 2022, doi: 10.33387/jiko.v5i1.3591.

[15] Y. Bae dan J. Kim, "Real-Time PI Gain Auto-Tuning for SPMSM Drives Based on Time-Domain Response Characteristics," vol. 18, no. 18, 2025.