

Comparison of PID and Inverter-Based Fuzzy Logic Control in an Induction Motor Blower Load

Lispindah Ayu Lestari

Electrical Engineering, Faculty of Engineering,
State University of Surabaya
Jl. Raya Kampus Unesa, Lidah Wetan, Kec.
Lakarsantri, Surabaya, Jawa Timur 60213
lispindah.23024@mhs.unesa.ac.id

Joko

Electrical Engineering, Faculty of Engineering,
State University of Surabaya
Jl. Raya Kampus Unesa, Lidah Wetan, Kec.
Lakarsantri, Surabaya, Jawa Timur 60213
joko@unesa.ac.id

Article history: Received Jule 27, 2026| Revised August 07, 2026| Accepted August 08, 2026

Abstract – Three-phase induction motors are widely used to drive centrifugal blowers in industrial ventilation and process systems because of their reliability and compatibility with variable-frequency drives. However, speed regulation under centrifugal blower loads is challenging because load torque increases with the square of rotational speed, while power demand increases with its cube, producing nonlinear dynamics that affect controller performance. Previous comparisons of Proportional-Integral-Derivative (PID) and Sugeno Fuzzy Logic controllers have largely focused on simulations, direct-current motors, or constant loads, leaving experimental evaluation on inverter-fed three-phase induction motors with real centrifugal blower loads insufficiently explored. This study experimentally compares PID and Sugeno Fuzzy Logic controllers implemented on an ESP32-based inverter-fed three-phase induction motor driving a centrifugal blower. PID was tuned using the experimentally optimized Ziegler-Nichols method, while Sugeno Fuzzy Logic employed 49 inference rules based on error and delta error. Experiments were conducted at a 1050 rpm setpoint under identical operating conditions. The PID controller achieved a steady-state error of 0.35%. Its overshoot was 7.25%. Its rise time was 19 seconds. Its settling time was 44 seconds. The Sugeno Fuzzy Logic controller achieved a steady-state error of 0.69%. Its overshoot reached 27.95%, notably higher than PID. However, its rise time was only 1.3 seconds. Its settling time was 3.9 seconds. Thus, Sugeno Fuzzy Logic provided faster transient response, whereas PID offered better steady-state accuracy and overshoot suppression. The results demonstrate a practical trade-off that can guide controller selection for inverter-fed induction motor industrial blower applications.

Keywords: *Blower Load, ESP32, Induction Motor, Inverter, PID, Sugeno Fuzzy Logic.*



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

I. INTRODUCTION

Three-phase induction motors are among the most widely used electric motors in industrial

applications because of their simple construction, low maintenance requirements, and high reliability [1][2]. Electric motors account for more than 40% of the world's total electricity consumption, and induction motors make up more than 80% of the industrial electric motor population [3][4], making the efficiency of their control a crucial aspect. One common application of induction motors is as drives for centrifugal blowers in ventilation systems and industrial processes [6][7].

Controlling the speed of an induction motor under a centrifugal blower load presents unique challenges rooted in the nonlinear relationship between load, power, and rotational speed. According to the fan/blower affinity laws, load torque increases proportionally to the square of rotational speed, while power consumption increases proportionally to the cube of rotational speed [7][8]. This nonlinear characteristic implies that even a small deviation in speed produces a disproportionately larger change in load torque and power demand, imposing a rapidly time-varying disturbance on the control loop. Consequently, a control strategy applied to an induction motor blower system must be capable of responding not only to setpoint tracking but also to continuously changing load dynamics, a condition difficult to fully replicate through simulation or through testing on motors with constant or linear loads. This characteristic forms the primary technical motivation for evaluating control performance directly on a real induction motor blower system rather than relying solely on simulated or DC motor based experiments.

Conventional induction motor speed control can be achieved through voltage or frequency regulation. The use of Variable Frequency Drives (VFDs) enables precise control of motor voltage and frequency through Pulse Width Modulation (PWM), allowing motor speed to be adjusted according to process requirements while improving drive efficiency [3], [4]. According to the blower affinity law, a 20%

reduction in rotational speed can theoretically reduce power demand by up to 49% [8]. This highlights the importance of VFD-based speed control for industrial blower applications, although direct energy consumption measurement is beyond the scope of this study, which focuses on transient and steady-state speed-control performance.

Two commonly used control methods for inverter-based motor speed regulation are Proportional Integral Derivative (PID) control and Fuzzy Logic control. PID control is known for its simple implementation and adequate performance in systems with relatively linear characteristics, but it has limitations in systems with nonlinear dynamics and time-varying loads [18][21][11]. Meanwhile, Sugeno Fuzzy Logic control can handle system uncertainties and nonlinearities without requiring an exact mathematical model, offering the advantages of computational efficiency and a smooth output signal [14][15]. Several previous studies have compared the performance of these two methods on various types of motors and drive configurations. Mugheri and Keerio [24] demonstrated that a simplified Fuzzy-PI controller outperforms both open-loop scalar control and a conventional PI controller in induction motor speed regulation. It is important to distinguish the present study's contribution from these advances: whereas Bal [27] demonstrated overshoot reduction through a more sophisticated type-3 fuzzy architecture under simulated or vector-controlled conditions, the present study contributes direct experimental validation of a standard zero-order Sugeno controller on physical hardware under a real, mechanically coupled centrifugal blower load providing empirical baseline evidence of the specific overshoot mechanism that more advanced architectures such as type-3 fuzzy logic are designed to address. Salahuddin et al. [26] applied a Mamdani fuzzy logic controller with tuned membership functions across multiple discrete operating speeds, underscoring the flexibility of fuzzy-based approaches in variable-speed drive applications. However, these studies remain limited in two important respects: they are predominantly conducted through simulation, on scalar/V-F control schemes, or on motor types other than a physically loaded three-phase induction motor–blower system, and none directly address the compounded nonlinearity introduced by a centrifugal blower load acting on a three-phase induction motor under real hardware conditions. Direct experimental evidence on how PID and Sugeno Fuzzy Logic perform under a real, dynamically varying blower load implemented on a practical embedded platform such as the ESP32 remains scarce [4][13]. This gap constitutes the primary scientific contribution of the present study.

This study focuses on setpoint-tracking performance under a constant, fully mounted centrifugal blower load, representing the transient and steady-state characteristics of each controller. Dynamic load disturbances, such as abrupt changes in

blower damper position, are beyond the scope of this study and are proposed for future research on hybrid Fuzzy–PID control. The study aims to design, implement, and compare PID and Sugeno Fuzzy Logic controllers for an inverter-fed three-phase induction motor driving a centrifugal blower. Performance is evaluated based on steady-state error, overshoot, rise time, and settling time. An ESP32 microcontroller was used to implement the control algorithms and process the real-time feedback signal [4], [20]. The results are expected to provide a practical reference for selecting suitable speed-control strategies for induction motor–blower applications.

II. METHOD

A. System Design

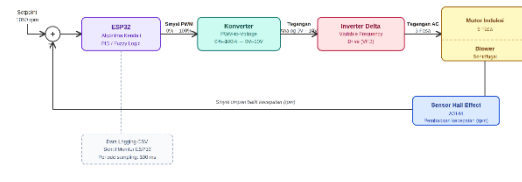


Figure 1. Block Diagram of a VFD-Based Induction Motor Speed Control System

This study employs an experimental approach by implementing two speed control methods directly on actual hardware. The system consists of six main subsystems that are integrated, as shown in Figure 1: an ESP32 microcontroller, a PWM-to-voltage converter, a delta inverter, a three-phase induction motor, a centrifugal blower, and an A3144 Hall-effect sensor.

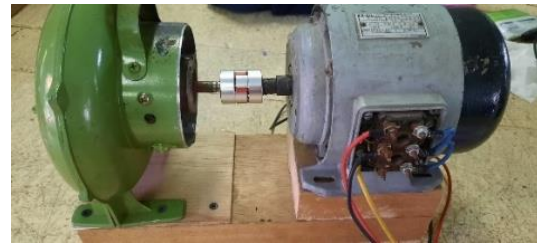


Figure 2. Three-Phase Induction Motor Combined with a Centrifugal Blower



Figure 3. Shaft Coupling Types

Table 1. Hardware Specifications of the Induction Motor and Delta Inverter

Component	Parameter	Value
Three-phase induction motor	Rated power	0.37 kW (0.5 HP)
	Rated voltage	220/380 V (Δ/Y)
	Rated current	2.6/1.7 A (Δ/Y)
	Rated frequency	50 Hz
	Number of poles	4 poles (synchronous speed 1500 rpm)
	Rated speed	1360 rpm
Delta inverter	Model	Delta VFD015M21A (M-Series)
	Input voltage	1-Phase 230 V AC, 50/60 Hz
	Output voltage range	0–230 V AC (3-phase PWM output)
	Output frequency range	0.1–400 Hz
	Control interface	0–10 V analog input (AVI) as frequency reference

A three-phase induction motor is directly coupled to a centrifugal blower on a single shaft without a reduction gear to simulate real-world operating conditions in an industrial blower system. The ESP32 generates a PWM signal with a duty cycle of 0%–100% based on the control algorithm's output. Since the Delta inverter requires an analog voltage reference signal of 0V–10V, a PWM to Voltage converter is used as an interface to convert the PWM signal via a low-pass filter circuit and an operational amplifier (op-amp) into a 0V–10V voltage.

The A3144 Hall Effect sensor reads pulses from a permanent magnet on the motor shaft using the ESP32's interrupt technique, allowing the actual speed in rpm to be calculated in real time as feedback for the closed-loop control system. All data is recorded via the ESP32 Serial Monitor in CSV format with a sampling period of 100 ms.

B. Implementation Of PID Control

PID control is one of the most widely used control methods in industrial applications due to its simplicity and its ability to deliver good performance in systems with linear characteristics [18][21]. The PID control algorithm implemented on the ESP32 is based on the following discrete PID formulation:

$$u[k] = K_p \cdot e[k] + K_i \sum_{j=0}^k e[j] \cdot \Delta t + K_d \frac{e[k] - e[k-1]}{\Delta t} \quad (1)$$

Where $u(k)$ is the control signal at the k th sampling period in the form of a PWM duty cycle

(0%–100%), $e(k)$ is the error at the k th sampling period, calculated as the difference between the setpoint and the actual motor speed measured by the A3144 Hall-effect sensor; T is the sampling period of 0.1 seconds; K_p is the proportional gain; K_i is the integral gain; and K_d is the derivative gain.

The value $u(k)$ generated by the PID algorithm represents the PWM signal duty cycle sent to the PWM-to-Voltage converter. The converter then converts this duty cycle value into an analog voltage of 0 V–10 V as the frequency reference signal for the Delta inverter. Thus, an increase in the duty cycle by the PID algorithm directly results in an increase in the reference voltage at the Delta inverter, which in turn increases the output frequency and the motor's rotational speed [3][4].

The three PID components work synergistically to generate an optimal control signal. The proportional component generates a control signal proportional to the magnitude of the current error; thus, the greater the deviation of the actual speed from the setpoint, the greater the duty cycle correction sent through the converter to the Delta inverter. The integral component accumulates the error over time to eliminate residual steady-state error caused by blower load disturbances or system imperfections. The derivative component responds to the rate of change of the error to dampen oscillations and accelerate system stabilization under dynamic blower load conditions [21][11].

PID parameter tuning is performed in two stages. The first stage determines the initial parameters using the Ziegler-Nichols method. K_p is gradually increased until the system exhibits sustained oscillations, yielding the critical gain (K_u) and critical period (T_u). Then $K_p = 0.6 \cdot K_u$, $K_i = 2 \cdot K_p / T_u$, and $K_d = K_p \cdot T_u / 8$ [18][11]. The second stage involves manual parameter optimization through experimental testing with the blower load installed, observing the speed response via ESP32 Serial Monitor data logging, and iteratively adjusting the parameters in the order of K_p , K_i , and then K_d [11][12]. The final results of the PID parameter tuning are:

Table 2. PID Tuning Results

Parameter	Value
K_p	0,8
K_i	0,04
K_d	0,05

To prevent integrator windup, clamping is applied to the integral component accumulation with upper and lower limits set based on the PWM duty cycle operating range, namely 0% to 100%, which corresponds to the 0V–10V voltage range at the Delta inverter input [12][13].

C. Implementation Of Sugeno Fuzzy Logic Control

The Fuzzy Logic controller was implemented using the zero-order Sugeno inference method. The Sugeno method was chosen for its computational

efficiency and its ability to produce smooth, continuous output signals [15][19]. These characteristics are well-suited for implementation on the ESP32, given that the fuzzy algorithm with 49 rules must be executed within a single 100-ms sampling period, and its output consists of a Δ PWM value that is converted to an analog voltage by a PWM-to-voltage converter before being sent to the Delta inverter [4][20].

1. Input Variables

The Sugeno Fuzzy Logic system uses two input variables calculated from the speed data read by the A3144 Hall-effect sensor at each sampling period. The first input variable is the error (e), which is the difference between the speed setpoint and the actual motor speed:

$$e(k) = \text{setpoint} - \text{RPM}_{\text{aktual}}(k) \quad (2)$$

The second input variable is the delta error (Δe), which is the rate of change of the error between two consecutive sampling periods and reflects the system's dynamic behavior:

$$\Delta e(k) = e(k) - e(k-1) \quad (3)$$

A positive delta error value indicates that the error is increasing, requiring a more aggressive PWM duty cycle correction through the converter to the Delta inverter, while a negative delta error value indicates that the error is decreasing, so the correction can be reduced to prevent overshoot [14][19].

2. Membership Functions

From a theoretical standpoint, a fuzzy set is defined by a membership function that maps each element of a crisp input domain to a degree of membership within the interval $[0, 1]$, thereby allowing a continuous variable to be represented through overlapping linguistic categories rather than discrete, sharply bounded intervals [25]. In the zero-order Takagi–Sugeno–Kang (TSK) fuzzy inference framework adopted in this study, the choice of membership function shape directly influences both the smoothness of the fuzzy partition and the computational cost of the fuzzification process, which is a critical consideration for real-time embedded implementation [20].

In this study, each input variable error (e) and delta error (Δe) is represented by seven fuzzy sets: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), and Positive Big (PB). A triangular membership function is adopted for both input variables. This choice is theoretically justified on two grounds. First, triangular membership functions require only three parameters per set (left boundary, peak, right boundary) and involve solely linear interpolation during fuzzification, resulting in lower computational overhead, which is an important property given that the ESP32 must complete the entire 49-rule inference process within a single 100-ms sampling period, consistent with real-time fuzzy

logic implementations reported on the same microcontroller platform [20]. Second, triangular partitions provide a smooth and continuous mapping between numerical input values and their corresponding linguistic representations, avoiding discontinuities in the resulting control surface a partitioning approach widely adopted in fuzzy-based induction motor speed control [19],[27].

This configuration two input variables, seven linguistic terms per variable, and a resulting 7×7 rule base of 49 rules is consistent with established practice in fuzzy-based induction motor speed control reported in the literature. Shaukat et al. [19] employed an equivalent Takagi–Sugeno structure with error and delta-error inputs for induction motor speed regulation, demonstrating its suitability for embedded real-time control. More recently, Bal [27] adopted an identical two-input, seven-membership-function, 49-rule configuration in a type-3 fuzzy logic controller for induction motor speed control, reporting that this granularity of fuzzy partitioning provides a favorable balance between control resolution and computational tractability. Similarly, Mugheri and Keerio [24] demonstrated that even a simplified fuzzy partition with fewer linguistic terms can achieve satisfactory dynamic performance, indicating that the seven-term configuration adopted in the present study offers additional resolution that may be leveraged for finer control near the zero-error region.

The universe of discourse for each input variable was determined empirically from the observed operating range of the physical system during preliminary testing, following the common practice of deriving fuzzy partition boundaries from the actual dynamic range of the controlled process rather than from arbitrary or normalized limits [19], [27]. For the error variable (e), the observed range spans from approximately -300 rpm corresponding to the largest negative deviation recorded during the transient overshoot phase to approximately $+1050$ rpm, corresponding to the maximum positive error occurring at system start-up when the actual speed is zero and the setpoint is 1050 rpm. For the delta error variable (Δe), the universe of discourse was set to a narrower symmetric range reflecting the maximum rate of change in error observed between two consecutive 100-ms sampling periods under normal transient conditions. Figures 4 and 5 illustrate the triangular membership function plots for both input variables, showing the seven fuzzy sets distributed across their respective universes of discourse.

It is worth noting that the boundaries of the universe of discourse, together with the input and output scaling factors, constitute tunable design parameters that directly affect controller aggressiveness; an inappropriately wide universe of discourse relative to the system's actual operating range can cause the rule base to apply disproportionately large corrective actions even for moderate errors, a factor that is further examined in the

Discussion in relation to the overshoot behavior observed for the Sugeno Fuzzy Logic controller.

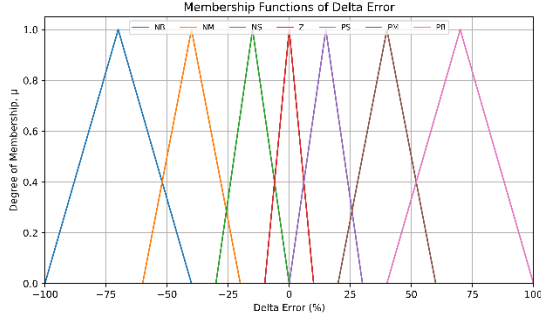


Figure 4. Membership Functions Of Delta Error

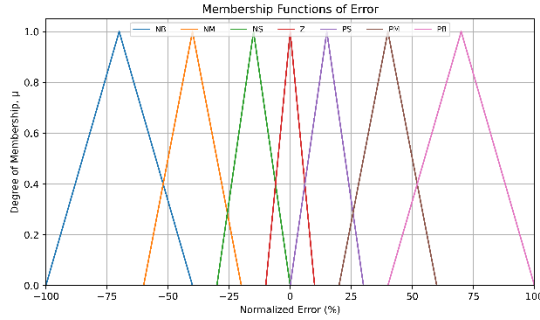


Figure 5. Membership Functions of Error

3. Rule Base

The combination of seven fuzzy sets across two input variables results in a total of 49 IF-THEN rules in the form of a 7×7 rule-base matrix [14][15]. Each rule defines the corresponding Δ PWM output value based on the current error and delta error conditions. The general principle applied is that a large error requires a large Δ PWM correction, causing the reference voltage at the Delta inverter to change significantly, whereas an error approaching zero with a low rate of change requires a minimal Δ PWM correction to maintain motor speed stability.

4. Defuzzification

The defuzzification method used is the weighted average typical of the Sugeno method [15][19]:

$$\Delta PWM = \frac{\sum_{i=1}^n (w_i \cdot c_i)}{\sum_{i=1}^n w_i} \quad (4)$$

Where w_i is the firing strength of the i -th rule and c_i is the singleton output value of the i -th rule. The defuzzified Δ PWM value is added incrementally to the PWM value from the previous period:

$$PWM(k) = PWM(k-1) + \Delta PWM(k) \quad (5)$$

The resulting PWM(k) value represents the duty cycle of the PWM signal sent to the PWM-to-Voltage converter to be converted into a 0V–10V analog voltage as the reference signal for the Delta inverter. This incremental approach produces smooth and gradual changes in the reference voltage at the Delta inverter, thereby preventing overly drastic frequency spikes and protecting the motor and the system's mechanical components [5][20]. The PWM value is limited to a range of 0%–100%, which

corresponds to the 0V–10V voltage range at the Delta inverter's input.

D. Testing And Evaluation Parameters

A setpoint of 1050 rpm was selected as the nominal operating condition, corresponding to approximately 77% of the motor's rated speed under full blower load. Both PID and Sugeno Fuzzy Logic controllers were tested under identical conditions, with the centrifugal blower fully mounted on the motor shaft and the motor initially at rest before each test.

The motor speed was measured in real time using an A3144 Hall-effect sensor and processed by the ESP32. The control signal generated by the ESP32 was converted from PWM to analog voltage and transmitted to the Delta inverter. Speed data were recorded through the ESP32 Serial Monitor in CSV format with a 100-ms sampling period.

The reported results represent a single characterization trial under one operating condition and controlled laboratory conditions. Although this provides an initial evaluation of the transient and steady-state characteristics of both controllers, performance across different speed setpoints was not investigated and is recommended for future work. The comparison was based on four standard performance parameters: steady-state error, overshoot, rise time, and settling time.

1. Steady-state error (E_{ss}), which measures the motor speed accuracy under steady-state conditions relative to the setpoint:

$$E_{ss} = \frac{|RPM_{avg} - setpoint|}{setpoint} 100\% \quad (6)$$

where RPM avg is the average of the actual speeds from five consecutive data samples under steady-state conditions. The system is considered to have good accuracy if the E_{ss} value is below the 1% threshold.

2. The overshoot percentage (M_p), which measures how far the motor speed exceeds the setpoint during the initial transient phase:

$$M_p = \frac{(RPM_{max} - setpoint)}{setpoint} 100\% \quad (7)$$

where max RPM is the maximum speed measured by the A3144 Hall-effect sensor during the transient process.

3. Rise time (T_r), defined as the time required for the system to move from 10% to 90% of the setpoint value, that is, from 105 rpm to 945 rpm [18][20].
4. Settling time (T_s) measures the time required for the system's actual speed to enter and permanently stabilize within a tolerance range of $\pm 5\%$ of the setpoint, specifically between 997.5 rpm and 1102.5 rpm [18][19].

III. RESULTS AND DISCUSSION

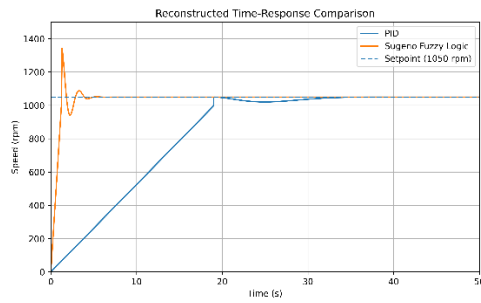


Figure 6. Reconstructed time-response comparison of PID and Sugeno Fuzzy Logic control

Figure 6 presents a reconstructed time-response curve comparing the PID and Sugeno Fuzzy Logic controllers, derived from the reported transient characteristics (overshoot percentage, rise time, and settling time) summarized in Tables 4–6. Panel (a) shows the full transient behavior over a 50-second window, illustrating the markedly slower but smoother approach of the PID controller toward the setpoint, in contrast to the rapid but more oscillatory response of the Sugeno Fuzzy Logic controller. Panel (b) provides a magnified view of the first six seconds, in which the Fuzzy Logic controller's higher overshoot and faster settling become clearly visible relative to the still-rising PID response. This visualization complements the numerical data in Table 6 by allowing direct visual comparison of transient and steady-state behavior, which numerical tables alone cannot convey.

A. Results Of The PID Control Performance Verification

Verification of the PID control performance was conducted by setting a setpoint of 1050 rpm and observing the motor speed response in real time via CSV data logging from the ESP32 Serial Monitor with a sampling period of 100 ms. The control signal generated by the PID algorithm on the ESP32 is sent to a PWM-to-Voltage converter, which converts it into an analog voltage of 0V–10V, serving as the frequency reference for the Delta inverter. The inverter then regulates the rotational speed of the induction motor according to the specified setpoint.

1. Steady-State Analysis

The verification results show an average speed of 1053.72 rpm under steady-state conditions, obtained by averaging five consecutive data samples from the 44th to the 48th second, as presented in Table 3.

Table 3. Steady-State Sampling Data For PID Control

Sample No.	Time (seconds)	Speed (rpm)
1	44	1060,74
2	45	1050,12
3	46	1053,15
4	47	1050,72
5	48	1053,79
Average		1053,72

The steady-state error (Ess) is calculated using (6):

$$Ess = \frac{|1053,72 - 1050|}{1050} 100\%$$

$$Ess = 0,35\%$$

$$Ess = 3,72 \text{ rpm}$$

The Ess value of 0.35% is well below the 1% tolerance threshold, indicating that the PID control implemented on the ESP32 is capable of maintaining the motor speed close to the setpoint with good accuracy under steady-state conditions. This low Ess value is primarily due to the integral component ($K_i = 0.04$), which continuously accumulates and corrects the residual error until it approaches zero. The small fluctuations still visible in the steady-state sample data ranging from 1050.12 to 1060.74 rpm—are caused by the dynamic variation in the centrifugal blower's load torque relative to rotational speed, as well as noise in the A3144 Hall Effect sensor readings due to mechanical vibrations in the system.

2. Transient Response Analysis

The transient response characteristics of the PID system were analyzed based on three main parameters: overshoot, rise time, and settling time.

- The maximum overshoot measured by the A3144 Hall-effect sensor was 1125.93 rpm. The overshoot percentage (M_p) was calculated using Equation (7):

$$M_p = \frac{(1125,93 - 1050)}{1050} 100\%$$

$$M_p = 7,25\%$$

- An overshoot of 7.25% occurs during the initial acceleration phase of the motor because, under conditions of very large error, the PID proportional component generates a high control signal, causing the PWM duty cycle to increase significantly. This causes the reference voltage at the Delta inverter to rise rapidly and the output frequency to increase sharply before the derivative component can effectively dampen the rate of speed increase. Nevertheless, an overshoot value of 7.25% is considered small and safe for the mechanical components of the blower system. This confirms that the set K_d value of 0.05 is sufficiently effective in damping the initial speed surge.
- The system's rise time is defined as the time required to reach 90% of the setpoint value, which is 945 rpm. Based on the ESP32 Serial Monitor logging data, the motor speed reached 918.79 rpm at $t = 18$ seconds and 957.95 rpm at $t = 19$ seconds, meaning the system reached 90% of the setpoint within the range of $t = 18.7$ –19 seconds. Thus, the PID system's rise time is set at 19 seconds. This relatively long rise time is a consequence of PID tuning parameters that prioritize minimizing overshoot and ensuring stability; the selected K_p value of 0.8 results in a

moderate proportional response. Additionally, the high mechanical inertia of the motor-blower system and the blower's quadratic load characteristics further slow the motor's acceleration during the initial phase, requiring more time to reach the setpoint operating range.

- d. The settling time is determined based on a tolerance criterion of $\pm 5\%$ of the setpoint, with a lower limit of $1050 - 0.05(1050) = 997.5$ rpm and an upper limit of $1050 + 0.05(1050) = 1102.5$ rpm. Based on observations of the ESP32 Serial Monitor data, the actual motor speed enters and consistently stabilizes within this tolerance range at $t \approx 44$ seconds. The obtained settling time of 44 seconds indicates that the PID system takes a relatively long time to stabilize the motor speed under the load of the centrifugal blower. This condition is caused by the interaction between the integral component which takes time to gradually eliminate residual error the high mechanical inertia of the motor-blower system, and the quadratic load characteristics of the blower, which cause the load torque to vary significantly during the stabilization process [6][7]. All parameters from the PID control verification are summarized in Table 4.

Table 4. Results Of The PID Control Performance Verification

Parameter	Value	Unit
Steady-state	1053,72	rpm
Steady-state error (Ess)	0,35	%
Steady-state error (Ess)	3,72	rpm
Maximum overshoot	1125,93	rpm
Overshoot percentage (Mp)	7,25	%
Rise time (Tr)	19	second
Settling time (Ts)	44	second

These results are consistent with previous research showing that PID control with Ziegler-Nichols tuning yields good steady-state accuracy in inverter-based motor systems, albeit at the cost of longer rise and settling times.

B. Verification Results Of The Sugeno Fuzzy Logic Control Performance

The performance of the Sugeno Fuzzy Logic controller was verified using the same setpoint, namely 1050 rpm, with a sampling period of 100 ms. The actual motor speed was monitored in real time by the A3144 Hall-effect sensor, and the defuzzified Δ PWM value was converted to an analog voltage of 0 V–10 V by a PWM-to-voltage converter before being sent to the Delta inverter.

1. Steady-State Analysis

Based on the test results, the average speed under steady-state conditions was found to be 1042.79 rpm. The steady-state error (Ess) was calculated using Equation (6):

$$Ess = \frac{|1050 - 1042,79|}{1050} 100\%$$

$$Ess = 0,69\%$$

$$Ess = 7,21 \text{ rpm}$$

The Ess value of 0.69% remains below the 1% threshold, indicating that the Sugeno Fuzzy Logic control can maintain motor speed close to the setpoint with acceptable accuracy under steady-state conditions, although it is slightly lower than that of the PID control. The difference in Ess values between PID and Sugeno Fuzzy Logic stems from the fundamental differences in the error correction mechanisms of the two algorithms. PID has an integral component that explicitly accumulates and corrects the residual error until it approaches zero, whereas Sugeno Fuzzy Logic lacks an explicit error accumulation mechanism and relies entirely on the rule base to correct small deviations around the setpoint. When the error is small (close to the fuzzy set zero), the Δ PWM correction generated by the rule base is also very small, so there is still a residual speed deviation from the setpoint under steady-state conditions.

Analysis of the control output signal under steady-state conditions shows that the Δ PWM value varies between 53.91 and 55.87, with a Δ PWM difference of 1.96. This very small variation in the Δ PWM value indicates that the changes in the control signal converted to an analog voltage by the PWM-to-Voltage converter and fed to the Delta inverter occur very smoothly without extreme oscillations. This condition proves that the 49-rule rule base is capable of producing proportional and stable control corrections in response to small error changes under steady-state conditions, and demonstrates that there is no hunting or chattering in the control signal that could cause unwanted speed fluctuations.

2. Transient Response Analysis

The transient response characteristics of the Sugeno Fuzzy Logic control system were analyzed based on three main parameters: overshoot, rise time, and settling time.

- a. The maximum overshoot measured by the A3144 Hall-effect sensor was 1343.43 rpm. The overshoot percentage (Mp) was calculated using Equation (7):

$$Mp = \frac{(1343,43 - 1050)}{1050} 100\%$$

$$Mp = 27,95\%$$

An overshoot of 27.95% occurred during the initial acceleration phase of the motor and can be explained by three main interacting factors, namely:

- 1) The high mechanical inertia of the three-phase induction motor, which prevents the speed from responding instantly to control corrections, even though the voltage

reference signal at the Delta inverter has increased rapidly.

- 2) The dynamic load characteristics of the centrifugal blower, where the load torque varies with the square of the speed; thus, during the initial acceleration phase, the load is still very low, and the motor tends to accelerate much faster than desired before the load increases significantly.
 - 3) The characteristics of the Fuzzy Logic algorithm, which does not explicitly use a mathematical model of the motor; thus, under conditions of large error (fuzzy set PB), the rule base applies the maximum Δ PWM correction without directly considering the system's momentum and inertia. Although the overshoot of 27.95% is greater than that of the PID controller, this value is still acceptable in the context of controlling the speed of an induction motor with a blower load because the system does not enter a dangerous state, and the speed stabilizes again within a short time. This is consistent with the findings of previous research reporting that Fuzzy Logic without hybrid optimization tends to produce greater overshoot in motor systems with dynamic loads.
- b. Rise time is defined as the time required for the system to move from 10% to 90% of the setpoint. The 10% setpoint value is 105 rpm, reached within the range of $t = 2\text{--}3$ seconds, while the 90% setpoint value is 945 rpm, reached within the range of $t = 15\text{--}16$ seconds. Thus, the rise time of the Sugeno Fuzzy Logic system is calculated as:

$$T_s = (40 - 1) \times 0,1 = 3,9 \text{ second}$$

The settling time of 3.9 seconds indicates that although the Sugeno Fuzzy Logic system experiences greater overshoot in the initial phase, it is able to stabilize the motor speed within the tolerance range much faster than the PID system. This is due to the adaptive capability of the fuzzy rule base to provide precise and proportional Δ PWM corrections in accordance with the error and delta error conditions that change during the stabilization process. When the motor speed exceeds the setpoint and the error becomes negative (the NB or NM sets), the rule base automatically generates a negative Δ PWM that reduces the reference voltage at the Delta inverter, thereby decreasing the output frequency and causing the motor speed to quickly return to the setpoint. All parameters from the verification of the Sugeno Fuzzy Logic control are summarized in Table 5.

Table 5. Verification Results of the Sugeno Fuzzy Logic Control Performance

Parameter	Value	Unit
Steady-state	1042,79	rpm
Steady-state error (Ess)	0,69	%
Steady-state error (Ess)	7,21	rpm
Maximum overshoot	1343,43	rpm
Overshoot percentage (Mp)	27,95	%
Rise time (Tr)	1,3	second
Settling time (Ts)	3,9	second

C. Comparison Of The Performance Of PID Control And Sugeno Fuzzy Logic

A comprehensive comparison between PID control and Sugeno Fuzzy Logic is presented in Table 6, while the corresponding transient responses are illustrated in Figure 6. The experimental evaluation was conducted under identical operating conditions, namely a speed setpoint of 1050 rpm with a centrifugal blower operating under full-load conditions. The combined presentation of numerical performance indices and time-response curves provides a clearer interpretation of the transient and steady-state characteristics of both controllers.

Table 6. Comparison Of The Performance of PID Control and Sugeno Fuzzy Logic

Parameter	PID	Fuzzy Logic Sugeno	Advantages
Steady-state speed (rpm)	1053,72	1042,79	—
Steady-state error (%)	0,35	0,69	PID
Steady-state error (rpm)	3,72	7,21	PID
Maximum overshoot (rpm)	1125,93	1343,43	PID
Overshoot percentage (%)	7,25	27,95	PID
Rise time (seconds)	19	1,3	Fuzzy
Settling time (seconds)	44	3,9	Fuzzy

In terms of steady-state accuracy, the PID control achieved an Ess of 0.35% (3.72 rpm), which is better than the Sugeno Fuzzy Logic control's 0.69% (7.21 rpm). This advantage of PID stems from the integral component, which continuously corrects the residual error until it approaches zero, whereas Sugeno Fuzzy Logic does not have an explicit integral mechanism. In terms of transient stability, PID control is also superior, with an overshoot of only 7.25% compared to 27.95% for the Sugeno Fuzzy Logic. This difference is due to the PID's derivative component, which dampens the initial speed surge, whereas the

Sugeno Fuzzy Logic applies maximum Δ PWM correction in the early phase without accounting for system momentum.

Conversely, Sugeno Fuzzy Logic is significantly superior in transient response speed, with a rise time of 1.3 seconds versus 19 seconds for PID (14.6 times faster), and a settling time of 3.9 seconds versus 44 seconds for PID (11.3 times faster). These advantages stem from the nonlinear and adaptive nature of Fuzzy Logic, which is capable of instantly generating large Δ PWM corrections when the error is high.

This trade-off pattern is consistent with several prior comparative studies. Almawla et al. [14] similarly reported that fuzzy-based controllers excel in transient response while PID controllers remain more stable at steady-state, corroborating the present findings under a physically loaded induction motor system rather than a simulated environment, and reinforcing the notion that this trade-off is a structural characteristic of the two control paradigms rather than an artifact specific to a particular hardware setup. Mugheri and Keerio [24] likewise found that a simplified Fuzzy-PI controller substantially outperformed a conventional PI controller in dynamic response, though at the cost of reduced steady-state refinement a pattern mirrored in the present results and suggesting that this trade-off persists across different motor types and control architectures. Notably, the magnitude of overshoot observed here (27.95%) is considerably higher than the overshoot reported by Bal [27] for a type-3 fuzzy logic controller (below 5% across six test scenarios) applied to an indirect vector-controlled induction motor. This discrepancy suggests that the performance gap is not an inherent limitation of fuzzy-based control in general, but rather a consequence of the specific inference architecture (zero-order Sugeno with fixed triangular membership functions) and the absence of scaling-factor optimization in the present design, indicating clear room for methodological refinement in future iterations of this research.

The high overshoot exhibited by the Sugeno Fuzzy Logic controller (27.95%) can be attributed to three interacting factors, as discussed in the Transient Response Analysis: the high mechanical inertia of the motor-blower system, the quadratic load characteristics of the centrifugal blower during the low-load initial acceleration phase, and the absence of a derivative-like damping mechanism in the zero-order Sugeno inference structure. Because these three factors act simultaneously during the earliest moments of the transient response, their combined effect is considerably more pronounced than any single factor would produce in isolation. To mitigate this limitation, two concrete technical strategies are proposed for future refinement of the controller design:

First, input/output scaling-factor tuning. The present design maps the full observed error range directly onto the fixed universe of discourse without

an adjustable gain. Introducing scaling factors K_e , $K_{\Delta e}$, and K_u applied respectively to the error input, delta-error input, and the Δ PWM output allows the effective sensitivity of the rule base to be tuned independently of the membership function boundaries. Reducing K_u during the high-error region (fuzzy sets PB/NB) would proportionally attenuate the magnitude of the initial Δ PWM correction, thereby reducing the aggressiveness of the response during start-up without requiring redesign of the rule base itself, and without sacrificing the fast recovery behavior that characterizes the controller once the error approaches the Zero region.

Second, membership function remapping. The current triangular membership functions distribute the fuzzy sets uniformly across the universe of discourse. An asymmetric remapping widening the membership functions near the PB/NB extremes while narrowing those near Zero would reduce the rate at which maximum Δ PWM correction is triggered as the error approaches its peak value, effectively introducing a soft-limiting behavior analogous to the derivative damping present in PID control. This approach is consistent with scaling-factor optimization strategies reported in prior fuzzy-PID hybrid studies [23], and could be further enhanced through metaheuristic tuning methods such as genetic algorithm or particle swarm optimization, which would allow the scaling factors to be determined systematically rather than through manual trial and error, and adapted specifically to the inertia and load profile of the motor-blower system used in this study.

Both strategies preserve the core advantage of Sugeno Fuzzy Logic its fast transient response while directly targeting the mechanism responsible for excessive overshoot, and are proposed as concrete, implementable directions for the hybrid Fuzzy-PID development discussed in the Conclusion.

Overall, the experimental results demonstrate a clear trade-off between transient response speed and steady-state performance. PID control provides higher steady-state accuracy and lower overshoot, making it suitable for applications prioritizing speed regulation precision and mechanical safety. In contrast, Sugeno Fuzzy Logic offers substantially faster dynamic response, making it advantageous for applications requiring rapid adaptation to operating-point changes. The proposed scaling-factor tuning and membership-function remapping provide practical approaches for improving the transient performance of Sugeno Fuzzy Logic while reducing excessive overshoot. Therefore, integrating both approaches through a hybrid Fuzzy-PID strategy constitutes a promising direction for future research.

IV. CONCLUSION

This study experimentally evaluated PID and Sugeno Fuzzy Logic controllers for speed regulation of an inverter-fed three-phase induction motor driving a centrifugal blower. The results demonstrate a clear

trade-off between transient and steady-state performance. PID provides better steady-state accuracy and lower overshoot, whereas Sugeno Fuzzy Logic provides a significantly faster transient response due to its nonlinear rule-based control mechanism. Therefore, controller selection should be based on the specific operational requirements of the blower application. The main contribution of this study is the experimental evaluation of both controllers on a real induction motor centrifugal blower system, providing practical evidence of their performance under nonlinear loading conditions. This study is limited to a single-trial characterization under one operating condition; repeated trials with statistical analysis, such as mean and standard deviation across multiple runs, would further strengthen the reliability of the reported performance metrics and are therefore recommended for future work. Future studies should also investigate scaling-factor optimization and membership-function adjustment to reduce the relatively high overshoot of the Sugeno Fuzzy Logic controller. Furthermore, The main contribution of this study is the experimental evaluation of both controllers on a real induction motor-centrifugal blower system, providing practical evidence of their performance under nonlinear loading conditions. This study is limited to a single-trial characterization under one operating condition; repeated trials with statistical analysis, such as mean and standard deviation across multiple runs, would further strengthen the reliability of the reported performance metrics and are therefore recommended for future work. To address the high overshoot of the Sugeno Fuzzy Logic controller, future work should specifically investigate a hybrid Fuzzy-PID architecture in which the fuzzy inference system continues to govern the initial transient phase (large error, fuzzy sets PB/NB) for its fast response, while a PID compensator is activated once the error enters the near-setpoint region (fuzzy sets PS/Z/NS) to eliminate residual steady-state error and suppress oscillation, similar to gain-scheduled switching strategies. Within this architecture, three parameters are proposed as concrete optimization targets: (1) the input/output scaling factors K_e , $K_{\Delta e}$, and K_u , which govern the sensitivity of the fuzzy rule base; (2) the crossover error threshold at which control authority transitions from Fuzzy to PID; and (3) the PID gains (K_p , K_i , K_d) activated during the compensation phase. These parameters could be tuned using Particle Swarm Optimization or a Genetic Algorithm, with a multi-objective fitness function that jointly minimizes overshoot percentage and settling time, evaluated against the actual motor-blower load profile characterized in this study. The 0.37 kW induction motor used in this study represents a scaled laboratory prototype consistent in operating principle with larger industrial blower drives; while absolute power ratings differ, the fundamental control challenges nonlinear load torque proportional to the square of speed and the associated transient dynamics are structurally identical across power scales, supporting the relevance of these findings to larger industrial blower systems, though

direct validation at industrial power ratings remains a subject for future work.

V. ACKNOWLEDGMENTS

The authors would like to thank the Electrical Engineering Laboratory, Faculty of Engineering, Surabaya State University, for its technical support and facilities during this research. We would also like to thank our colleagues who provided constructive feedback during the system development and manuscript preparation.

REFERENCES

- [1] Ihsan Faturrohman and Mohammad Fatkhurrokhman, "Pengaturan Kecepatan Motor Induksi 3fasa Dengan Mengatur Frekuensi Menggunakan VSD di PERUMDAM Tirta Madani Serang," *J. Sains dan Teknol.*, vol. 2, no. 1, pp. 46–56, 2023, doi: 10.58169/saintek.v2i1.135.
- [2] R. Rasyid, M. Muhammad, and R. R. Rasyid, "Pengaturan Kecepatan Motor Induksi Tiga Fasa Menggunakan Spwm Inverter dan Kontrol Pid dengan Metode Hibrid Volt/Herzt Konstan-Sensor Arus," *PROtek J. Ilm. Tek. Elektro*, vol. 9, no. 1, p. 57, 2022, doi: 10.33387/protk.v9i1.4201.
- [3] R. S. Widagdo, B. Hariadi, and K. Setyadjit, "Modelling and Analysis of Ziegler-Nichols and Chien-Hrones-Reswick Tuning PID on DC Motor Speed Control," *J. Teknol. Elektro*, vol. 14, no. 1, p. 23, 2023, doi: 10.22441/jte.2023.v14i1.005.
- [4] D. K. P. M. F. S. Adhisuwigno, "KONTROL PID PADA PENGATURAN KECEPATAN MOTOR AC BERBASIS ESP32," *Kohesi J. Sains dan Teknol.*, no. Vol. 2 No. 6 (2024): Kohesi: Jurnal Sains dan Teknologi, pp. 51–60, 2024, [Online]. Available: <https://ejournal.warunayama.org/index.php/kohesi/article/view/2216/2073>
- [5] S. Yahya, A. Fauzi, A. R. Al Tahtawi, and S. M. Ilman, "Kendali Kecepatan Motor Induksi 3-Fasa Dengan Metode Adaptif Fuzzy-PID," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 10, no. 1, pp. 01–11, 2024, doi: 10.15575/telka.v10n1.01-11.
- [6] A. Tikkani and P. V. N. Prasad, "Type-1 and type-2 fuzzy logic-based space vector modulation for two-level inverter fed induction motor," *Telkomnika (Telecommunication Comput. Electron. Control)*, vol. 20, no. 4, pp. 901–913, 2022, doi: 10.12928/TELKOMNIKA.v20i4.22454.
- [7] B. Subarta, H. Eteruddin, and D. Setiawan, "Analisis Perbandingan Sistem Kendali PID dan Sistem Kendali Fuzzy PID pada Pengontrolan Kecepatan Motor DC," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 10, no. 2, pp. 178–192, 2024, doi: 10.15575/telka.v10n2.178-192.
- [8] K. Sekarsari, O. A. Rozak, A. Sunardi, J. Raya Puspitek, and T. Selatan, "Pemodelan dan Simulasi Sistem Kendali PID dan Fuzzy pada Motor DC Menggunakan LabView PID and Fuzzy Control System Modeling and Simulation on DC Motor Using LabView," *Telka*, vol. 10, no. 3, doi: <https://doi.org/10.15575/telka.v10n3.241-250>.
- [9] R. Afrawira, R. Fajar Gumilang, S. Amalia, and S. Bandri, "Analisa Perbandingan Pengendali PID pada Motor DC Menggunakan Metode Ziegler-Nichols dan

- Trial and Error," *Ranah Res. J. Multidiscip. Res. Dev.*, vol. 5, no. 3, pp. 210–218, 2023, doi: 10.38035/rj.v5i3.758.
- [10] N. L. Manuel, N. İnanç, and M. Lüy, "Control and performance analyses of a DC motor using optimized PIDs and fuzzy logic controller," *Results Control Optim.*, vol. 13, no. May, 2023, doi: 10.1016/j.rico.2023.100306.
- [11] A. M. Prasetya and M. N. Ramadani, "Implementation of Induction Motor Speed Control Using a PID Controller," *Fidel. J. Tek. Elektro*, vol. 6, no. 1, pp. 12–20, 2024, doi: 10.52005/fidelity.v6i1.195.
- [12] S. Chandrasekaran, S. Durairaj, and S. Padmavathi, "A Performance Improvement of the Fuzzy Controller-Based Multi-Level Inverter-Fed Three-Phase Induction Motor with Enhanced Time and Speed of Response," *J. Electr. Eng. Technol.*, vol. 16, no. 2, pp. 1131–1141, 2021, doi: 10.1007/s42835-020-00649-6.
- [13] A. Tikkani *et al.*, "Fuzzy-2 deployment in indirect vector control and hybrid space vector modulation for a two-level inverter fed induction motor drive," *Sci. Rep.*, vol. 15, no. 1, p. 13379, 2025, doi: 10.1038/s41598-025-96600-8.
- [14] A. M. Almawla, M. J. Hussein, and A. T. Abdullah, "A Comparative Study of DC Motor Speed Control Techniques Using Fuzzy, SMC and PID," *J. Eur. des Syst. Autom.*, vol. 57, no. 2, pp. 397–406, 2024, doi: 10.18280/jesa.570209.
- [15] I. Irianto, F. D. Murdianto, E. Sunarno, and D. D. Proboningtyas, "Comparison Method of PI, PID and Fuzzy Logic Controller to Maintain Speed Stability in Single Phase Induction Motors," *INTEK J. Penelit.*, vol. 8, no. 1, pp. 7–16, 2021, doi: 10.31963/intek.v8i1.2687.
- [16] R. Shenbagalakshmi, S. K. Mittal, J. Subramaniyan, V. Vengatesan, D. Manikandan, and K. Ramaswamy, "Adaptive speed control of BLDC motors for enhanced electric vehicle performance using fuzzy logic," *Sci. Rep.*, vol. 15, no. 1, p. 12579, 2025, doi: 10.1038/s41598-025-90957-6.
- [17] N. Rohadi, L. K. Men, and A. Hidayat, "PLC Mitsubishi : Studi Eksperimental," *J. Nas. Tek. Elektro dan Teknol. Inf. (JNTETI)*, vol. 15, no. 1, pp. 26–33, 2026, pp. 26–33, 2026, doi: 10.22146/jnteti.v15i1.22694.
- [18] A. D. M. Africa, J. O. Q. Chua and J. L. H. Solis, "PID Tuning of Speed Controller Using Ziegler-Nichols and Manual Method DC Motor," 2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Coron, Palawan, Philippines, 2023, pp. 1–6, doi: 10.1109/HNICEM60674.2023.10589041.
- [19] N. Shaukat, B. Khan, C. A. Mehmood and S. M. Ali, "Takagi-Sugeno Fuzzy Logic Based Speed Control of Induction Motor," 2016 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2016, pp. 280–285, doi: 10.1109/FIT.2016.058.
- [20] Y. Pamungkas, N. Cahya Romadhona, Z. Satria Alam Syahputra, K. Nazwa Rayhani, A. Sakti Wibowo and F. Akbar Parinduri, "Implementation of Fuzzy Logic Control on the ESP32 Microcontroller for an Automatic Infusion Monitoring System," 2023 International Conference on Modeling & E-Information Research, Artificial Learning and Digital Applications (ICMERALDA), Karawang, Indonesia, 2023, pp. 289–294, doi: 10.1109/ICMERALDA60125.2023.10458205.
- [21] D. Santoso and D. Izci, "Control Strategy Design for DC Motor Speed Regulation Using PID Controller," *SEED J. Sci. Res.*, vol. 2, no. 2, pp. 58–66, 2026, doi: 10.70716/seed.v2i2.396.
- [22] R. Nutenki and B. Varma, "PID Controller Design for DC Motor Speed Control," in *Proc. 2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 2024, pp. 1–6, doi: 10.1109/ICAECT60202.2024.10469124.
- [23] T. Prasoenphon, K. Selamassakul, T. Sittiwanchai, and W. Jitviriya, "Fuzzy-PID vs. PID Controllers: A Comparative Study on DC Motor Speed Control," in *Proc. 2024 1st International Conference on Robotics, Engineering, Science, and Technology (RESTCON)*, 2024, pp. 64–68, doi: 10.1109/RESTCON60981.2024.10463588.
- [24] N. H. Mugheri and M. U. Keerio, "An Optimal Fuzzy Logic-based PI Controller for the Speed Control of an Induction Motor using the V/F Method," *Eng. Technol. Appl. Sci. Res.*, vol. 11, no. 4, pp. 7399–7404, Aug. 2021, doi: 10.48084/etasr.4255.
- [25] L. A. Zadeh, "Similarity relations and fuzzy orderings," *Inf. Sci. (Ny)*, vol. 3, no. 2, pp. 177–200, 1971, doi: [https://doi.org/10.1016/S0020-0255\(71\)80005-1](https://doi.org/10.1016/S0020-0255(71)80005-1).
- [26] H. Salahuddin, K. Imdad, M. U. Chaudhry, D. Nazarenko, V. Bolshev, and M. Yasir, "Induction Machine-Based EV Vector Control Model Using Mamdani Fuzzy Logic Controller," *Appl. Sci.*, vol. 12, no. 9, p. 4647, 2022, doi: 10.3390/app12094647.
- [27] C. Bal, "Type-3 Fuzzy Logic-Based Robust Speed Control for an Indirect Vector-Controlled Induction Motor," *Appl. Sci.*, vol. 15, no. 22, p. 11994, 2025, doi: 10.3390/app152211994.