

Conveyor Speed Control with Fuzzy-PID

Minh-Nam Phan¹, Van-Thao Nguyen², The-An Nguyen³, Thi-Ngoc-Thao Nguyen⁴, Thanh-Binh Nguyen⁵,
Thi-Hong-Lam Le^{6,*}, Van-Hiep Nguyen⁷, Hoang-Lam Le⁸, Van-Phuc Nguyen⁹, Phong-Nam Huynh¹⁰, Vi-Cuong Hong¹¹,
Tuan-Minh Truong¹², Kieu-Vinh Nguyen¹³, Huu-Nhan Tran¹⁴

^{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} Faculty of Electrical and Electronics Engineering (FEEE), Ho Chi Minh City University of
Technology and Engineering (HCM-UTE), Ho Chi Minh City (HCMC), Vietnam

Email: ¹ 22151261@student.hcmute.edu.vn, ² 22151296@student.hcmute.edu.vn, ³ 22151170@student.hcmute.edu.vn,
⁴ thaontn@hcmute.edu.vn, ⁵ binhnt@hcmute.edu.vn, ⁶ lamlth@hcmute.edu.vn, ⁷ hiepnv@hcmute.edu.vn,
⁸ lamlh@hcmute.edu.vn, ⁹ 23142184@student.hcmute.edu.vn, ¹⁰ 22151030@student.hcmute.edu.vn,
¹¹ 22151010@student.hcmute.edu.vn, ¹² 21151425@student.hcmute.edu.vn, ¹³ 22145289@student.hcmute.edu.vn,
¹⁴ 20145086@student.hcmute.edu.vn

*Corresponding Author

Abstract—Conveyor belt systems are essential components in industrial automation, but controlling their speed accurately is a significant challenge. Classic PID controllers, while common, struggle to handle the inherent nonlinearities of these systems, such as varying friction and sudden load changes. The adaptive Fuzzy-PID controller, which auto-tunes its parameters, has been proposed as a superior alternative. However, most existing research is limited to software simulations, leaving a gap between theoretical performance and practical, real-world applicability. This paper addresses this gap by presenting the complete design, construction, and experimental verification of a Fuzzy-PID controller implemented on a physical conveyor belt model. The methodology includes system identification via multi-sine input to extract a baseline transfer function, followed by the deployment of the control algorithm on an embedded Arduino Nano microcontroller. Experimental results are presented and directly compared with those of a conventional PID controller to evaluate its performance. Quantitative findings confirm that the embedded Fuzzy-PID controller provides superior performance, reducing the rise time to 0.16 s (from 0.20 s) and significantly decreasing the settling time to 0.56 s (a 48.1% improvement over the PID's 1.08 s). Furthermore, the steady-state error was reduced by 36.4%, demonstrating its superior stability and efficiency in a practical hardware environment and confirming its feasibility for industrial applications.

Keywords—Fuzzy Control; PID Control; Speed Control; Conveyor Belt

I. INTRODUCTION

In the context of the Industrial Revolution 4.0, the automation process has become a key factor in determining efficiency and precision in production. Among automated lines, the conveyor system is considered the "lifeline," responsible for transporting everything from raw materials to finished products. The efficiency of the entire factory depends directly on the precise and stable operation of these conveyors, especially in stages requiring absolute synchronization, such as robotic assembly, continuous weighing, or high-speed packaging.

Therefore, the core technical problem is controlling the speed of the drive motor. An effective controller must ensure that the conveyor's speed closely follows the setpoint quickly and accurately, while also being robust enough to counteract real-world disturbances like sudden load changes. Any failure in speed control can cause a cascading effect: damage

to products, reduced productivity, and disruption of the entire line's synchronization.

The control of mechatronic systems has undergone a long development process, with the PID (Proportional-Integral-Derivative) controller serving as the core foundation. As analyzed in the overview by C. Knospe in the IEEE Control Systems Magazine [1], the PID controller holds an absolute advantage in industrial applications due to its simple structure, ease of implementation, and high efficiency with linear systems. However, the effectiveness of PID is significantly reduced when faced with the nonlinear and uncertain characteristics of real systems. The motor-conveyor system is a typical example, where factors such as static and dynamic friction, sudden load changes, and the saturation region of the actuator are unavoidable, as has been analyzed in detail in surveys on friction models [2].

To address these challenges, intelligent control methods have emerged, based on theories of systems capable of "reasoning" and "adapting" [3]. Among them, fuzzy logic, first introduced by Lotfi Zadeh [4], has become an extremely powerful tool. The ability to simulate human thinking through "IF-THEN" linguistic rules allows it to control complex systems without needing a precise mathematical model.

The combination of the classic PID structure and the adaptive capability of fuzzy logic has created the hybrid Fuzzy-PID controller, an approach that has received deep interest. Specifically, recent studies have increasingly focused on applying intelligent controllers to industrial motor-conveyor systems to enhance adaptability [5]. Many works have demonstrated the superiority of this method. The research by Somwanshi et al. in the paper "Comparison of Fuzzy-PID and PID Controller for Speed Control of DC Motor using LabVIEW" [6] and the work of R. Li and P. Gong titled "Fuzzy PID Speed Controller of DC Motor Based on MATLAB" [7] are two typical examples. Both studies, though conducted on different software platforms, reached the same conclusion: the Fuzzy-PID controller is superior to the classic PID in terms of response time, stability, and noise immunity. Many researchers have even gone further by using optimization algorithms such as genetic algorithms to automatically find the best fuzzy rule base or PID tuning parameters, thereby further enhancing system performance [8], [9].

Most of the aforementioned works were verified in the ideal environment of computer software. However, as pointed out in studies on real-time embedded systems [10], converting an intelligent control algorithm from a computer to a low-cost microcontroller (like Arduino) is a major challenge. Issues such as processing speed limits, errors from integer arithmetic, electromagnetic interference from the motor, and delays in sensor reading are often overlooked in simulations.

This project is undertaken not to invent a new algorithm, but to address a profound technical and application-oriented question: "Can an advanced control solution, proven effective in theory, operate stably and efficiently when implemented on a common, low-cost hardware platform?". Stemming from the practical context and the identified research gap, the main contribution of this study is to provide a complete control solution that explicitly narrows the gap between simulation theory and real-world industrial applicability.

The main objectives of the paper include:

- A deep analysis of the sources of nonlinearity in the conveyor system and the need for an adaptive controller.
- Building a mathematical model and performing parameter identification for the motor-conveyor system from experimental data.
- A detailed design of the Fuzzy-PID controller and simulation on MATLAB-Simulink to evaluate its performance against the classic PID controller.
- Constructing the hardware model and programming the control algorithm on an Arduino Nano microcontroller, while also building a monitoring interface with Python.
- Conducting practical experiments to verify the efficiency, stability, and adaptive capability of the designed controller compared to the PID controller.

II. EXPERIMENTAL MODEL

A. Analysis and Modeling of the Controlled Object

The controlled object in this study is a conveyor belt system driven by a direct current (DC) motor (Fig. 1).

1. Theoretical Modeling

The electro-mechanical model of system is shown in Fig. 1. Torque T generated by the motor and back electromotive force (back EMF) e are described by the (1), (2):

$$T = K_t i \quad (1)$$

$$e = K_e \dot{\theta} \quad (2)$$

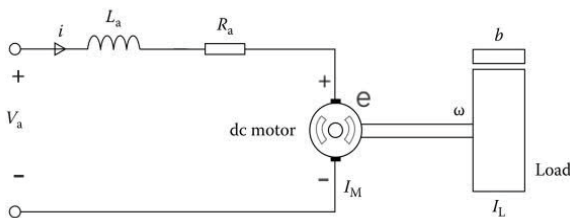


Fig. 1. Electro-mechanical model of DC motor and conveyor belt system

where: T : Motor torque (N.m), K_t : Torque constant of the motor (N.m/A), i : Armature current (A), e : Back electromotive force (V), K_e : Viscous friction constant of the motor (N. m. s), $\dot{\theta}$: Angular velocity of the motor shaft (rad/s)

Applying Kirchhoff's law for the armature circuit and Newton's 2nd law for the mechanical part, we have differential (3), (4) describing system:

$$J\ddot{\theta} + b\dot{\theta} = K_t i \quad (3)$$

$$L_a \frac{di}{dt} + R_a i = V_a - K_e \dot{\theta} \quad (4)$$

where: J : Total moment of inertia of the motor rotor and conveyor system (kg. m^2), b : Viscous friction constant of the motor (N. m. s), L_a : Inductance of the motor winding (H), R_a : Resistance of the motor winding (Ω), V_a : Input armature voltage (V), $\ddot{\theta}$: Angular acceleration (rad/s^2).

Applying Laplace transform, (3) and (4) can be expressed in terms of Laplace variables as (5), (6) follows:

$$s(Js + b) \cdot \theta(s) = K_t I(s) \quad (5)$$

$$(L_a s + R_a) I(s) = V_a(s) - K_e s \theta(s) \quad (6)$$

Substituting $I(s) = \frac{s(Js+b)}{K_t}$ from (5) into (6), we get (7):

$$(L_a s + R_a) \frac{s(Js + b)}{K_t} \theta(s) = V_a(s) - K_e s \theta(s) \quad (7)$$

The transfer function of the system is obtained in (8):

$$G(s) = \frac{\dot{\theta}(s)}{V_a(s)} = \frac{K_t}{(Js + b)(L_a s + R_a) + K_e K_t} \quad (8)$$

While the real conveyor system inherently exhibits severe nonlinear characteristics such as static friction (stiction), dead zones at low voltages, and actuator saturation limits [3], an approximated linear model is deliberately extracted in this phase. The rationale for choosing a linear baseline over a highly complex nonlinear model is twofold: first, it provides a straightforward mathematical foundation for the initial tuning of the classic PID parameters; second, the unmodeled nonlinearities and dynamic disturbances are precisely the factors that the proposed adaptive Fuzzy-logic block is designed to compensate for in real-time. Therefore, a strictly nonlinear mathematical model is unnecessary for the proposed hybrid control architecture.

2. Experimental System Identification

To obtain a model that is as close to reality as possible for simulation, the method of system identification from experimental input-output data is used.

The process (Fig. 2) is performed using the System Identification Toolbox in MATLAB. A multi-sine wave input voltage signal is chosen because it can excite the system at many different frequencies, helping the identified model to be more accurate over a wide frequency range.

Using the tfest (transfer function estimation) function configured for a continuous-time model with two poles and zero zeros (to match the theoretical 2nd-order structure in (7)), The identification result yields the transfer function (9) of the real system as follows:

$$G(s) = \frac{\theta(s)}{V_d(s)} = \frac{75.02}{s^2 + 7.56s + 94.26} \quad (9)$$

This transfer function has a high correlation with the experimental data (51.33% fit) and will be used to design and simulate the controller in the following sections.

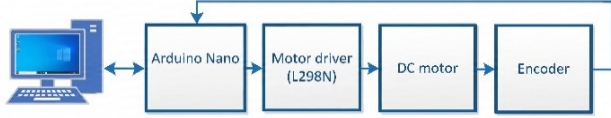


Fig. 2. Block diagram of the system identification process

B. Controller Theoretical Background

1. PID Controller

The PID controller calculates the control signal $u(t)$ based on three components: proportional (Proportional), integral (Integral), and derivative (Derivative) of the error $e(t)$. This algorithm is shown in (10):

$$u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (10)$$

- Proportional component (K_p): Increases the system's response speed, but a large K_p can cause overshoot and oscillation.
- Integral component (K_i): Eliminates steady-state error, but a large K_i can slow down the response and cause integral windup, a common nonlinear issue in practical control.
- Derivative component (K_d): Improves stability, reduces overshoot and oscillation, but is very sensitive to measurement signal noise [11].

2. Fuzzy Logic Background

Fuzzy logic, initiated by Lotfi A. Zadeh, is a method of processing "uncertain" or "vague" information, close to human reasoning. Instead of two true/false values (1/0), fuzzy logic allows values in the range [0, 1].

A fuzzy control system consists of 4 main components structured as follows [11]:

- Fuzzification: Converts (crisp) input values into fuzzy values using membership functions.
- Rule Base: A set of IF-THEN rules built by experts [12], describing the relationship between inputs and outputs.
- Inference Engine: Applies fuzzy rules to the fuzzified input values to produce a fuzzy output result. There are two main inference methods: Mamdani and Sugeno. The Mamdani method is chosen in this study because it is intuitive and close to human linguistic thinking.
- Defuzzification: Converts the fuzzy result back into a crisp value to control the actuator. The Center of Area (CoA) method is used because it provides a smooth result and accounts for the contribution of all activated rules.

C. Proposed Fuzzy-PID Controller Design

To overcome the drawbacks of PID and leverage the advantages of fuzzy logic, a hybrid Fuzzy-PID controller (Fig. 3) is designed. Its structure includes a conventional PID controller and a fuzzy logic controller (FLC). The FLC's task

is to automatically tune the K_p , K_i , and K_d parameters based on the current state of the system, represented by the error E and the change in error CE .

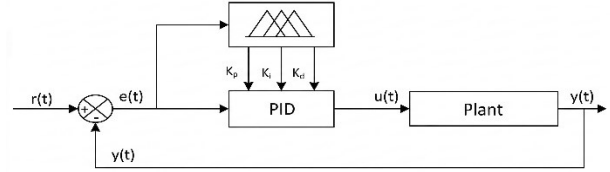


Fig. 3. Proposed Fuzzy-PID controller structure

FLC Inputs: error E and change in error CE , where: Error E is the difference between the setpoint speed and the actual speed. Change in error CE is calculated by the formula $CE[k] = \frac{e[k] - e[k-1]}{t}$.

FLC Outputs: Correction values for K_p , K_i , K_d . The specific FLC design is as Fig. 4.

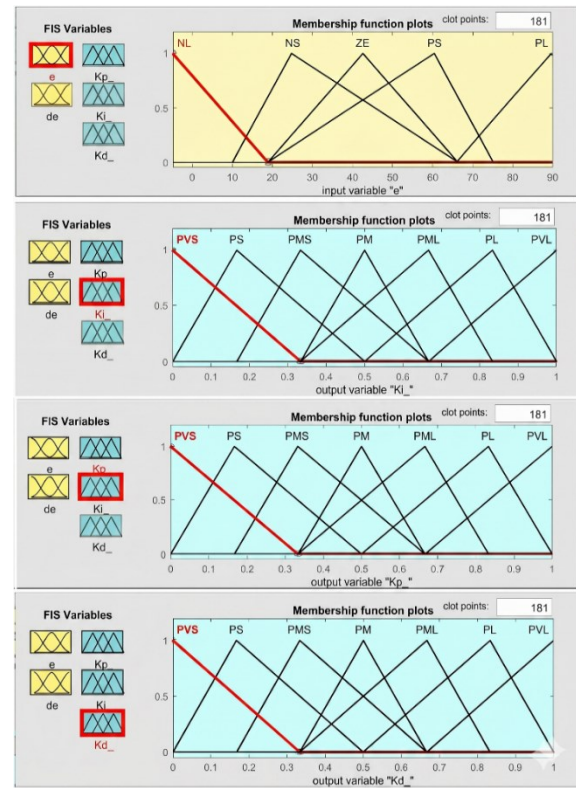


Fig. 4. Membership functions of the linguistic variables

Fuzzification: The two input variables E and CE are divided into 5 fuzzy sets: Negative Large (NL), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Large (PL). The outputs K_p , K_i , K_d are divided into 7 fuzzy sets: Very Small (VS), Small (S), Medium Small (MS), Medium (M), Medium Large (ML), Large (L), Very Large (VL). Triangular and trapezoidal (shoulder) membership functions are used because of their computational efficiency and stable tracking performance in embedded applications [13].

Rule Base: 25 fuzzy rules are constructed for each output based on control experience and principles of parameter self-tuning [14]: increase K_p when the error is large, increase K_i when the error is small, and increase K_d when the change in error is large. The fuzzy rules for the 3 outputs K_p , K_i and K_d are presented in Table 1 to Table 3.

Table 1. Fuzzy Rules Applied to The Kp Output

E / CE	NL	NS	ZE	PS	PL
NL	VL	L	ML	M	MS
NS	L	ML	M	MS	S
ZE	ML	M	MS	S	VS
PS	M	MS	S	VS	S
PL	MS	S	VS	S	MS

Table 2. Fuzzy Rules Applied to The Ki Output

E / CE	NL	NS	ZE	PS	PL
NL	VS	S	MS	M	ML
NS	S	MS	M	ML	L
ZE	MS	M	ML	L	VL
PS	M	ML	L	VL	L
PL	ML	L	VL	K	ML

Table 3. Fuzzy Rules Applied to The Kd Output

E / CE	NL	NS	ZE	PS	PL
NL	MS	S	VS	S	MS
NS	M	MS	S	VS	S
ZE	ML	M	MS	S	VS
PS	L	ML	M	MS	S
PL	VL	L	ML	M	MS

III. CONTROL METHOD

A. Simulation Implementation on MATLAB-Simulink

The Fuzzy-PID controller was built and simulated on MATLAB-Simulink using the identified transfer function. The results are compared with the open-loop system (no controller) and the system using a classic PID controller at a setpoint of 70 RPM. This comparison is shown in Fig. 5.

The simulation results (Table 4) show that the Fuzzy-PID controller achieves the best results. The system is not only stable (no overshoot, no steady-state error) but also has a fast response time (settling time of only 0.82s), almost twice as fast as the PID, demonstrating the intelligent self-tuning capability of the fuzzy block.

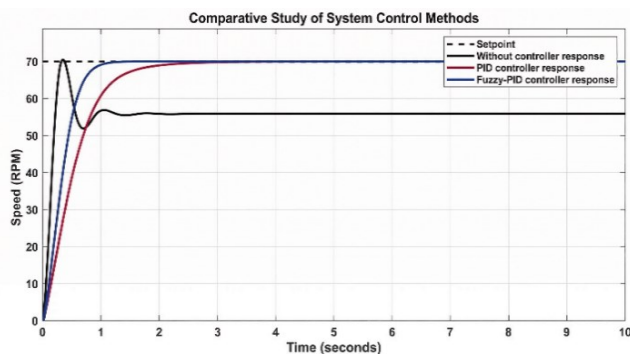


Fig. 5. Comparison of system responses in simulation (Setpoint = 70 RPM)

Table 4. Comparison of Response Quality Indicators in Simulation

Parameters	No Controller	PID	Fuzzy-PID
Overshoot (%)	26.49	0	0
Rise Time (s)	0.15	0.87	0.44
Steady-State Error (RPM)	14.29	0	0
Settling Time (s)	0.86	1.67	0.82

B. Model Construction and Experimental Verification

1. Hardware and Software Setup

To verify theory, a practical conveyor belt model was constructed as in Fig. 6.

Hardware: The main components include an Arduino Nano V3 Microcontroller, a GA25-370 12V DC gear motor, an L298N H-bridge module, and a 12V-3A switching power supply.

Software and Real-time Implementation Challenges: The control algorithm is programmed in C/C++ on the Arduino IDE. Implementing continuous control laws on digital microcontrollers requires careful consideration of sampling time and signal discretization to ensure stability. Specifically, deploying a Fuzzy Inference System on a low-cost, 16 MHz microcontroller like the Arduino Nano presents significant computational complexity. The evaluation of 75 fuzzy rules (25 rules for each parameter Kp, Ki, Kd) and the execution of Center of Area (CoA) defuzzification consume considerable CPU cycles. To address this real-time challenge, the algorithm was heavily optimized using efficient discrete mapping and conditional branching. This optimization ensured that the execution time of the entire control loop was constrained strictly within the 10 ms sampling period, preventing buffer overflows, execution overruns, and maintaining the robustness of the system against dynamic parameter variations.

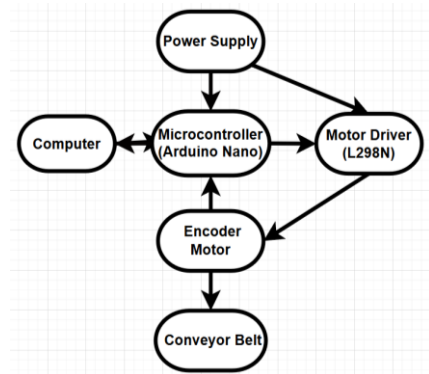


Fig. 6. Block diagram of the experimental system

The user interface (GUI) (Fig. 7) is built using Python with the Tkinter library for control and Matplotlib for real-time plotting. To ensure consistency throughout the manuscript, all linguistic labels, buttons, and parameters on the GUI have been standardized to English. Communication between the computer and the Arduino is implemented via the UART serial standard.



Fig. 7. Hardware model and Monitoring interface

IV. EXPERIMENTAL RESULTS AND DISCUSSION

Experiments were conducted directly on the hardware model. The experimental results show that the Fuzzy-PID controller performs very well on the real model. The average speed of the conveyor belt closely tracks the setpoint, with no overshoot, fast response time, and small steady-state error across all speed ranges (30, 50, 70 RPM) (Fig. 8). This consistency across different operating points highlights the controller's robustness against varying dynamic friction, a common nonlinear disturbance in physical conveyor belts.

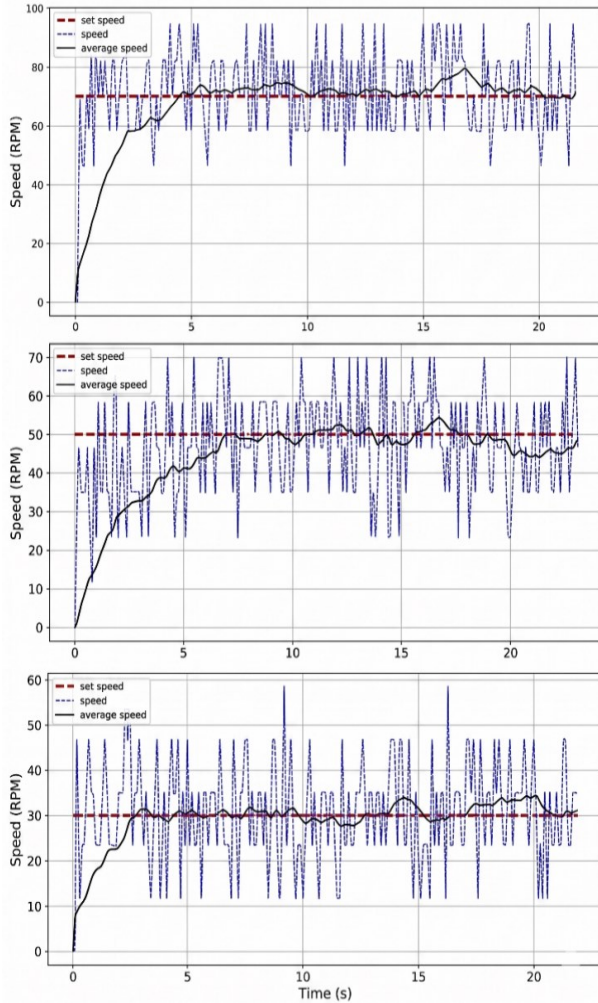


Fig. 8. Experimental response of the Fuzzy-PID controller at 70, 50, and 30 RPM speed levels

To evaluate further, a direct comparison experiment between the PID controller and the Fuzzy-PID controller was performed at a setpoint of 50 RPM (Fig. 9 and Table 3).

The comparison results on the real model once again confirm the superiority of the Fuzzy-PID controller. Specifically, Fuzzy-PID provides a 20% faster rise time (0.16s vs. 0.20s), a nearly 50% shorter settling time (0.56s vs. 1.08s), and significantly lower steady-state error (2.99 RPM vs. 4.70 RPM) compared to the PID controller, as shown in Table 5.

This superior performance can be explained by analyzing the adaptive tuning mechanism in real-time (illustrated in Fig. 10). A critical limitation in physical conveyor systems is actuator saturation; the L298N motor driver has a maximum voltage limit (PWM = 255). During the initial start-up phase,

when the speed error is maximum, a conventional PID's integral term accumulates rapidly, often leading to integral windup and subsequent overshoot. Conversely, the Fuzzy inference system intelligently mitigates this. According to the rule base (Table 2), when the error is Large, the Fuzzy block assigns a Very Small (VS) value to K_i , effectively preventing windup while maximizing K_p to accelerate the motor safely. As the speed approaches the setpoint (Error becomes Small), the Fuzzy block automatically increases K_d to increase system damping (preventing overshoot) and adjusts K_i upwards to eliminate the steady-state error caused by mechanical friction. This continuous, context-aware adjustment allows the Fuzzy-PID to navigate the physical constraints of the hardware far more gracefully than a fixed-gain controller.

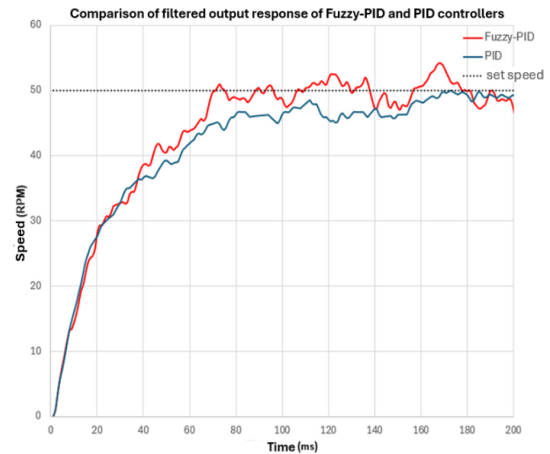


Fig. 9. Comparison of experimental output response between PID and Fuzzy-PID at a setpoint of 50 RPM

Table 5. Comparison of Experimental Quality Indicators (Setpoint =50 rpm)

Controller	Rise Time t_r (s)	Overshoot POT(%)	Settling Time t_{sl} (s)	Steady-State Error e_{sl} (RPM)
Fuzzy-PID	0.16	0	0.56	2.99
PID	0.20	0	1.08	4.70

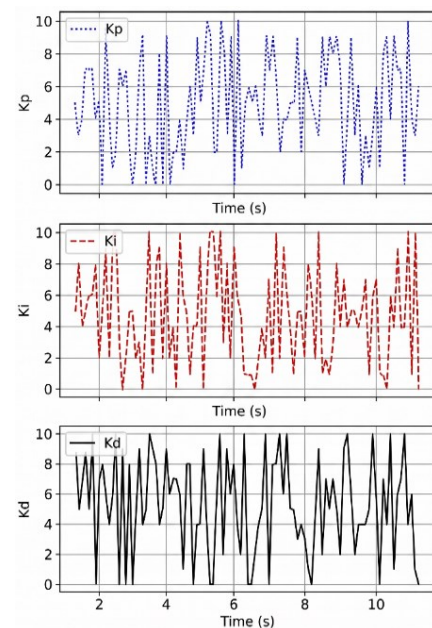


Fig. 10. Real-time change of PID parameters adjusted by the fuzzy block (K_p , K_i , K_d)

However, a multidimensional analysis reveals a necessary trade-off. While the Fuzzy-PID controller demonstrates clear superiority in control quality and robustness, it imposes a significantly higher computational burden. The real-time fuzzification, evaluation of 75 rules, and defuzzification processes consume considerably more CPU cycles and SRAM on the Arduino Nano compared to the simple arithmetic of a standard PID. In this specific conveyor application with a 10 ms sampling time, the hardware handles the load successfully. Yet, in highly dynamic industrial environments requiring extremely fast control loops (e.g., < 1 ms) or concurrent processing of multiple sensors, this computational overhead could become a hardware bottleneck, necessitating a migration to a more powerful microprocessor architecture (such as an ARM Cortex-M).

V. CONCLUSIONS AND FUTURE DEVELOPMENT

This paper successfully presents the design, simulation, and hardware implementation of an advanced conveyor speed control system using a hybrid Fuzzy-PID algorithm. The proposed system effectively overcomes the limitations of traditional PID controllers in nonlinear environments, demonstrating superior responsiveness and stability. Quantitative experimental results show significant performance enhancements: the rise time was reduced by 20% (to 0.16 s), the settling time was nearly halved to 0.56 s (a 48.1% improvement), and the steady-state error decreased by 36.4%. Furthermore, embedding the entire control algorithm on a microcontroller ensures practical applicability by enabling independent operation without computer-based computational support.

Despite these promising results, the current research relies heavily on empirical trial-and-error for designing fuzzy rules and membership functions. Additionally, the Arduino Nano hardware limits processing capabilities for high-speed applications, and the system is currently restricted to a laboratory scale. To address these limitations, future work will focus on integrating intelligent optimization algorithms, such as Genetic Algorithms (GA) or Particle Swarm Optimization (PSO), to systematically determine optimal fuzzy parameters. Upgrading to more powerful microcontrollers like the ESP32 or STM32 will also be necessary to support floating-point operations, faster sampling rates, and more complex computations.

Ultimately, future developments will aim to upgrade the system into a fully independent applied product by integrating load cells for dynamic weighing, vision systems for product classification, and standalone user interfaces. The results achieved in this study not only provide a highly effective solution for conveyor belt speed control but also highlight the significant potential of intelligent control methods in solving complex challenges within the field of industrial automation.

ACKNOWLEDGEMENT

This research was funded by Ho Chi Minh City University of Technology and Engineering, Vietnam, under grant No. SV2026-284. We want to give thanks to Ms. Eng. Thi-Thanh-Hoang Le (HCM-UTE) for her supervision.

REFERENCES

- [1] R. K. Mudi and N. R. Pal, "A robust self-tuning scheme for PI- and PD-type fuzzy controllers," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 1, pp. 2–16, 1999, <https://doi.org/10.1109/91.746295>.
- [2] C. Knospe, "PID control system analysis and design," *IEEE Control Systems*, vol. 26, no. 1, pp. 32–41, 2006, <https://doi.org/10.1109/MCS.2006.1580152>.
- [3] B. Armstrong-Hélouvry, P. Dupont, and C. C. De Wit, "A survey of models, analysis tools and compensation methods for the control of machines with friction," *Automatica*, vol. 30, no. 7, pp. 1083–1138, 1994, [https://doi.org/10.1016/0005-1098\(94\)90209-7](https://doi.org/10.1016/0005-1098(94)90209-7).
- [4] J. S. Albus, "Outline for a theory of intelligence," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 3, pp. 473–509, 1991, <https://doi.org/10.1109/21.97471>.
- [5] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965, [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X).
- [6] L. Liu and X. Ye, "Simulation research on vague PID control of conveyor belt based on MATLAB," in *Proceedings of SPIE 12981, Ninth International Symposium on Sensors, Mechatronics, and Automation System*, 2024, p. 70, <https://doi.org/10.1117/12.3014811>.
- [7] D. Somwanshi, M. Bunde, G. Kumar, and G. Parashar, "Comparison of fuzzy-PID and PID controller for speed control of DC motor using LabVIEW," in *Procedia Computer Science*, 2019, pp. 252–260, <https://doi.org/10.1016/j.procs.2019.05.019>.
- [8] L. Sha, Y. Song, and Z. Li, "Permanent Magnet Synchronous Motor Control Method Based on Fuzzy PI and Sliding Mode Observer," *Journal of Physics: Conference Series*, vol. 2218, no. 1, p. 12053, 2022, <https://doi.org/10.1088/1742-6596/2218/1/012053>.
- [9] Y. Yating, D. Pingan, and M. Shi, "Research on effect of measured curvature on eddy current evaluation by finite element analysis," in *Proceedings of the IEEE International Conference on Industrial Technology*, 2008, <https://doi.org/10.1109/ICIT.2008.4608447>.
- [10] E. W. Suseno and A. Ma'Arif, "Tuning of PID Controller Parameters with Genetic Algorithm Method on DC Motor," *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 41–53, 2021, <https://doi.org/10.31763/ijrcs.v1i1.249>.
- [11] K. Kroičs and A. Būmanis, "BLDC Motor Speed Control with Digital Adaptive PID-Fuzzy Controller and Reduced Harmonic Content," *Energies*, vol. 17, no. 6, p. 1311, 2024, <https://doi.org/10.3390/en17061311>.
- [12] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller—Part I," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 2, pp. 404–418, 1990, <https://doi.org/10.1109/21.52551>.
- [13] P. Chotikunann, R. Chotikunann, Y. Pitietheraphab, T. Puttasakul, A. Wongkamhang, and N. Thongpance, "Comparative Analysis of Fuzzy Membership Functions for Step and Smooth Input Tracking in a 3-Axis Robotic Manipulator," *Journal of Fuzzy Systems and Control*, vol. 3, no. 1, pp. 39–50, Jan. 2025, <https://doi.org/10.59247/jfsc.v3i1.278>.
- [14] Z.-L. Gaing, "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System," *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 384–391, 2004, <https://doi.org/10.1109/TEC.2003.821821>.