Indonesian Journal of Engineering and Science, Vol. 7, No. 2, 2026 *Phan, et.al.*
https://doi.org/10.51630/ijes.v7i2.222

DESIGN OF A PID CONTROLLER FOR SPEED OF A CONVEYOR

Truong-Nguyen Phan, Huu-Tai Nguyen, Quy-Kien Tran*, Trung-Nhan Nguyen, Minh-Thanh Nguyen, Hoang-Danh Vu, Thanh-Toan Nguyen, Nguyen-Quoc-Nam Dang, Duc-Quang-Thai Nguyen, Ba-Anh Le, Van-Dong-Hai Nguyen, Thi-Ngoc-Thao Nguyen, Hoang-Lam Le[1]

Ho Chi Minh City University of Technology and Engineering (HCM-UTE), Ho Chi Minh City (HCMC), Vietnam

**Corresponding author: 22151239@student.hcmute.edu.vn*

(Received: 03 March 2026; Accepted: 11 June 2026; Published on-line: 01 July 2026)

ABSTRACT: This study presents a compact and energy-aware conveyor speed control system designed to maintain stable and precise motor operation. A PID controller is implemented on an Arduino Mega, using encoder feedback and a discrete Kalman filter to reduce measurement noise and improve response smoothness. The system incorporates an H-bridge driver for motor actuation and an LCD module for on-site monitoring, together with a Python-based interface that provides real-time visualization of set speed, actual speed, and transient response via UART communication. Experimental results show that the controller achieves fast response, minimal steady-state error, and low overshoot across various reference speeds. Performance at a fixed 25-rpm setpoint under both no-load and light-load conditions further demonstrates good disturbance tolerance. Overall, the system offers a reproducible, low-cost, and energy-efficient solution suitable for small conveyor applications in educational and prototyping environments.

KEY WORDS: PID Control; Speed control; Conveyor; DC motor; Arduino.

1. INTRODUCTION

Conveyor systems, robotic manipulators, electric vehicles, and HVAC equipment all rely on accurate motor speed regulation to ensure stable operation, process reliability, and reduced energy losses. In manufacturing environments, conveyor lines alone account for a significant share of electrical power consumption, and poor speed regulation can lead to excessive acceleration cycles, vibration-induced losses, and premature mechanical wear. These effects directly increase energy waste and maintenance frequency, underscoring the environmental relevance of efficient conveyor motor control. As improving the dynamic stability of conveyor drives contributes to lower electricity usage and more sustainable material-handling operations, this topic aligns with the broader themes of energy efficiency and industrial sustainability.

To address these challenges, a wide range of motor-control strategies has been studied in recent years. Advanced PID-based approaches such as adaptive fuzzy fractional-order PID tuned using Genetic Algorithms for electric-vehicle motors [1], GA-optimized PID tuning for DC motors [2], and PID-based robotic control architectures [3] demonstrate the continued relevance of PID control in modern applications. Iterative feedback tuning represents another pathway for improving PID performance in dynamic environments [4]. Several studies have also highlighted the practicality of PID implementations on low-cost microcontrollers for real-time motor control [5], as well as hybrid PID–fuzzy approaches for enhanced robustness under varying loads [6]. Beyond controller tuning, recent research has examined disturbance modeling and compensation for conveyor drive systems [7], energy-efficient conveyor motor schemes [8], and Arduino-based conveyor automation frameworks [9]. Additional works have focused on embedded PID performance [10], encoder noise suppression using Kalman filters [11], low-cost filtering techniques for speed measurement [12], monitoring conveyor belt dynamics under impact loading [13], and energy consumption characteristics of industrial conveyor drives [14]. These studies collectively indicate that although significant progress has been made, most advanced optimization or hybrid-control approaches require increased computational resources or complex parameterization, limiting their suitability for small, resource-constrained systems.

Despite the extensive literature, there remains a practical gap for simple, low-cost, and energy-aware control solutions tailored to compact conveyor systems used in educational laboratories, prototyping platforms, and lightweight industrial tasks. In particular, many prior works rely heavily on simulation or assume idealized no-load conditions, whereas real conveyor systems experience variations in load, friction, and sensor noise. Moreover, advanced optimization-based controllers, while effective, often exceed the computational capabilities of low-cost microcontroller systems commonly deployed in small conveyors.

To address this gap, the present study develops and experimentally evaluates a manually tuned PID speed-control system implemented on an Arduino Mega platform. The encoder signal is processed through a Kalman filter used solely for noise suppression, improving speed estimation without incurring significant computation overhead. The originality of this work lies in its experimental emphasis on (i) systematically analyzing PID response quality—rise time, overshoot, steady-state error, and settling time—across multiple setpoint speeds under no-load conditions, and (ii) experimentally comparing dynamic performance at a fixed reference speed of 25 rpm under both no-load and loaded conditions, providing insight into disturbance tolerance and energy-related implications. By focusing on measurable performance behavior rather than algorithmic complexity, this study offers a practical, reproducible, and energy-efficient control solution suitable for small-scale conveyor systems.

Overall, this work contributes to the sustainable operation of low-power conveyors by improving speed stability, reducing unnecessary transients, and mitigating mechanical wear—factors that collectively support more energy-efficient material-handling processes. The proposed approach aligns on energy systems and environmental performance while maintaining the simplicity required for low-cost embedded hardware.

2. CONTROL ALGORITHM

2.1. PID

2.1.1. Concept of PID

A proportional–integral–derivative controller (PID controller or three-term controller) is a feedback-based control loop mechanism commonly used to manage machines and processes that require continuous control and automatic adjustment. It is typically used in industrial control systems and various other applications where constant control through modulation is necessary without human intervention. The general form of the PID control law is:

$$u(t) = K_P e(t) + K_i \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \tag{1}$$

where:

$u(t)$: control signal applied to the motor driver

$e(t) = \omega_{ref}(t) - \omega(t)$: speed error

K_p : proportional gain controlling reaction to instantaneous error

K_i : integral gain reducing steady-state error

K_d : derivative gain improving damping and transient response

The controller is implemented in discrete time with sampling period $T_s = 10ms$. The discrete PID equation is therefore:

$$u[k] = K_p e[k] + K_i T_s \sum_{i=0}^{k} e[i] + K_d \frac{e[k] - e[k-1]}{T_s} \tag{2}$$

To prevent actuator saturation and integral wind-up, the control signal is limited to [0,255] (8-bit PWM resolution), and the integral term is clamped within predefined bounds.

Fig. 1 illustrates the structure of a PID controller used in automatic control systems. The diagram shows how the three key components—Proportional (P), Integral (I), and Derivative (D)—work together to compute the control signal $u(t)$, which is then applied to the plant or process.

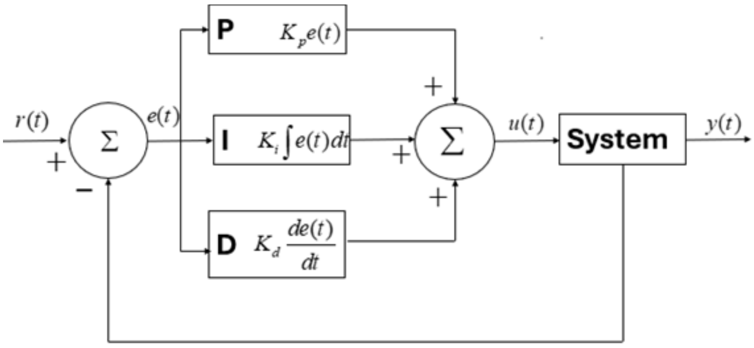

Fig. 1. Structure of PID controller

2.1.2. PID Parameter Tuning Using the Trial and Error Method

The PID gains used in this study were determined through a systematic trial–and–error tuning procedure, which is commonly applied in low-cost embedded motor control systems where plant models are unavailable or difficult to obtain. The objective of the tuning process was to achieve a stable and energy-efficient speed response with minimal overshoot, low steady-state error, and acceptable rise and settling times.

The tuning process consists of the following steps:

Step 1: Initial stabilization by adjusting Kp.

The integral and derivative gains were initially set to zero $(K_i = 0, K_d = 0)$. The proportional gain Kp was gradually increased from a very small value while observing the motor's response to a step command of 25 rpm.

+ When Kp was too low, the motor exhibited slow rise time and large steady-state error.

+ As Kp increased, steady-state error decreased and responsiveness improved.

+ The upper limit of Kp was identified when oscillations began to appear.

The final selected Kp was slightly below the oscillatory threshold to ensure sharp response without instability.

Step 2: Eliminating steady-state error by adjusting Ki

After selecting Kp, the integral gain Ki was gradually increased from zero.

+ Small values of Ki helped eliminate residual steady-state error.

+ Increasing Ki too much caused overshoot, oscillation, and slower settling time due to integral wind-up.

To prevent these effects, the integral term was clamped within a predefined saturation range.

The chosen Ki ensured that steady-state error at all reference speeds remained below 1 rpm.

Step 3: Improving Damping Using Kd

With Kp and Ki fixed, the derivative gain Kd was introduced to reduce overshoot and improve damping.

+ Small increments of Kd helped smooth the transient response.

+ Excessively large Kd amplified encoder noise, causing jitter in the PWM output.

Therefore, the selected Kd value represented a compromise between improved transient performance and noise sensitivity.

2.2. Flowchart

Fig. 2 illustrates the flowchart of a closed-loop motor control system. In this system, an encoder attached to the motor shaft generates pulses as the shaft rotates. These pulses are counted using external interrupts and are then processed to determine the actual motor speed.

To improve accuracy, the measured speed signal is passed through a Kalman filter, which helps reduce noise caused by measurement errors from the encoder. The Kalman filter enhances the stability and precision of the control system by performing the following steps:

– Predicting estimation error
– Calculating the Kalman gain
– Updating the speed estimation
– Updating the estimation variance

After filtering, the actual speed value is used by a PID controller, which calculates the error between the setpoint (desired speed) and the filtered actual speed. Based on this error, the controller generates an appropriate PWM (Pulse Width Modulation) signal to adjust the motor's speed accordingly.

The PWM signal is output through an H-bridge circuit, which not only controls the motor speed but also its direction of rotation. The user can send the desired speed value via Serial communication. Additionally, both the setpoint and the actual speed values are displayed on an LCD screen and sent back over Serial for monitoring purposes.

This structured approach ensures accurate, stable, and noise-resilient motor control, making it suitable for automation and embedded control systems. The integration of the Kalman filter before the PID controller is a key enhancement that enables the system to respond more smoothly to noise and mechanical vibrations.

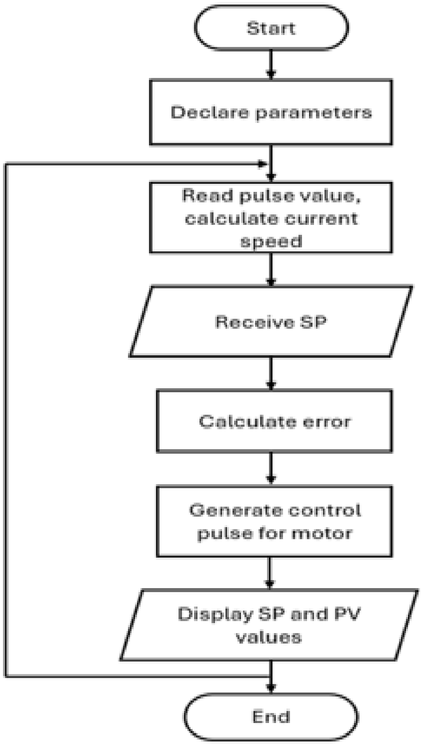

Fig. 2. Flowchart of the system algorithm

3. EXPERIMENTAL MODEL

3.1. Hardware

Fig. 3 illustrates the system consists of six main components: a setpoint input, a controller block, a PID controller, a motor, a mechanical load (conveyor belt), and a speed encoder. The setpoint defines the desired speed of the conveyor system. The controller block receives feedback from the speed encoder and compares it with the setpoint to determine the error. This error is processed by the PID controller, which generates a control signal to drive the motor. The motor delivers torque to the load, typically a conveyor belt. The speed encoder continuously measures the actual speed of the load and sends this data back to the controller block, thereby forming a closed-loop control system.

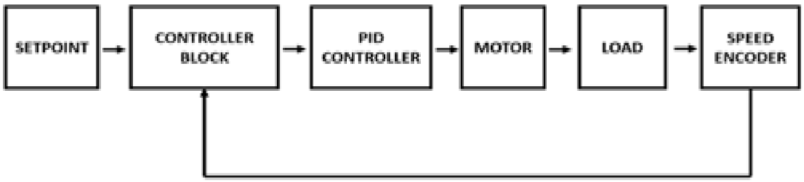

Fig. 3. Block Diagram of the conveyor belt speed control system

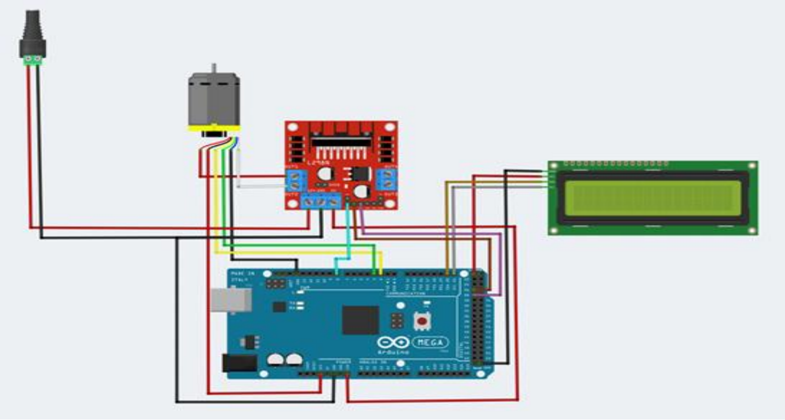

Fig. 4. 3D Hardware Connection Diagram of the System

This system integrates motor actuation, real-time control, and visual feedback into a closed-loop speed regulation application using an Arduino Mega 2560 microcontroller. The primary components include a DC motor, an L298N motor driver module, a speed encoder (integrated with the motor), and an LCD display for real-time monitoring.

The system receives a desired speed setpoint from the user, which is processed by the Arduino. The L298N module functions as the power modulation stage, providing the appropriate voltage and current to the DC motor based on control signals received from the Arduino's digital output pins. This module ensures bidirectional motor control and sufficient current handling capacity.

The motor shaft is coupled with an encoder that provides rotational speed feedback. The encoder's pulses are read through the Arduino's interrupt-capable digital pins to calculate the actual speed of the motor. This real-time feedback is essential for implementing closed-loop control.

A PID control algorithm is executed on the Arduino. It continuously compares the measured speed with the reference setpoint and generates control signals to adjust the motor's PWM duty cycle accordingly. This ensures stable and accurate speed tracking, even under varying load conditions.

An I2C-connected 16×2 LCD module displays system parameters such as the current speed, setpoint, and control output. This provides intuitive user interaction and facilitates debugging or demonstration purposes.

Together, these hardware components form a compact, closed-loop motor speed control system. It is well-suited for educational environments, laboratory experimentation, and prototyping of industrial conveyor control systems.

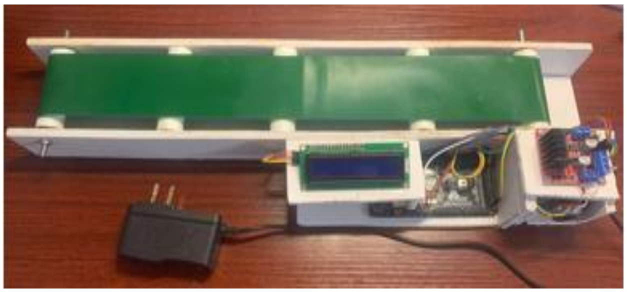

Fig. 5. Completed Hardware Prototype of the Conveyor Belt Speed Control System

The developed system consists of a mini conveyor belt constructed from simple materials, with a total belt length of 40 centimeters. A 12V DC motor with an integrated encoder is used to drive the conveyor and provide real-time feedback on rotational speed. Under load conditions, the motor can reach a maximum speed of 30 revolutions per minute (RPM). The control unit is based on an Arduino microcontroller, which reads encoder pulses, executes a PID control algorithm, and generates corresponding PWM signals to adjust the motor speed. The L298N H-bridge motor driver module allows for bidirectional speed and direction control of the DC motor. System status, including the desired and actual speed values, is displayed on a 16×2 I2C LCD module. The entire system is powered by a 12V DC adapter that supplies both the control and actuation components.

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of the PID controller under different operating conditions, a series of experiments were conducted at three target speeds: 5 RPM, 10 RPM, and 20 RPM. The primary objective was to assess the system's stability, response time, and steady-state accuracy at each speed level. In addition to the no-load condition, a final test was carried out with a mechanical load of 1 kilogram applied to the conveyor belt to examine how external disturbances affect the system dynamics.

During each test, the motor was commanded to reach the target speed from an initial rest state. Experimental data were collected including the rise time (time to reach 90% of the target speed), overshoot, steady-state error, and settling time. These metrics were recorded for all three speed levels and for both load and no-load conditions. The response curves for each test scenario were plotted and analyzed to evaluate the effect of the PID gains on system performance.

4.1. Performance Evaluation at Different Speeds

Through multiple experimental trials, the authors identified the optimal set of PID parameters for the system, with the following values: Kp = 3.5, Ki = 2.6, Kd = 1.

Fig. 6 illustrates the performance of the conveyor system operating at a target speed of 5 RPM using the optimized PID parameters.

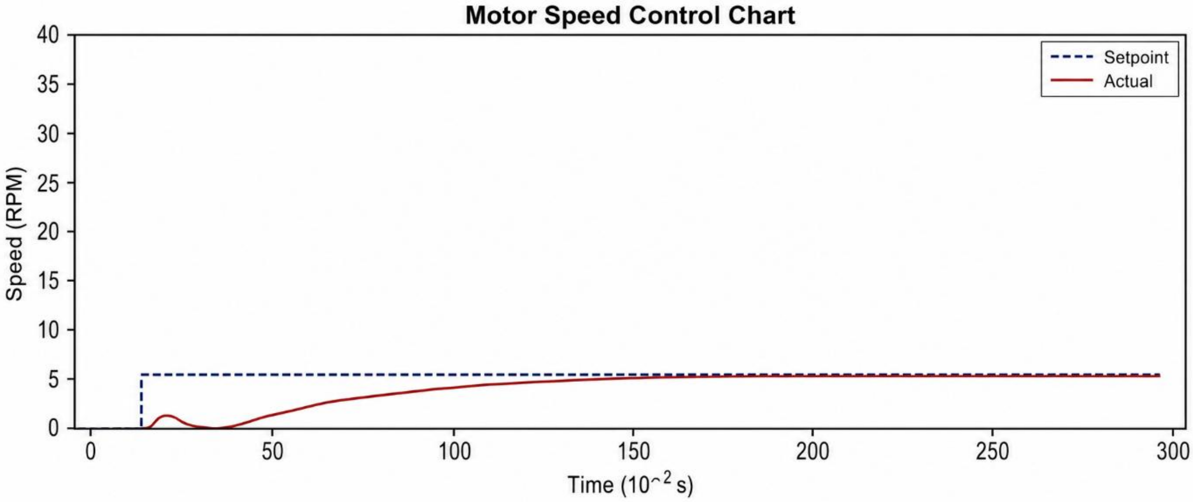

Fig. 6. System response at 5 RPM target speed

Analysis: The system response demonstrates effective speed control performance. The motor speed increases gradually and approaches the setpoint of 5 RPM with a short rise time and without significant overshoot. After approximately 140 time units, the actual speed stabilizes closely around the desired value. Minor oscillations are observed during the transient phase, but they quickly diminish, indicating a well-tuned controller and high system stability. The steady-state error is minimal, proving that the controller can accurately track the reference speed and maintain consistent performance under no-load conditions.

Fig. 7 illustrates the performance of the conveyor system operating at a target speed of 10 RPM using the optimized PID parameters.

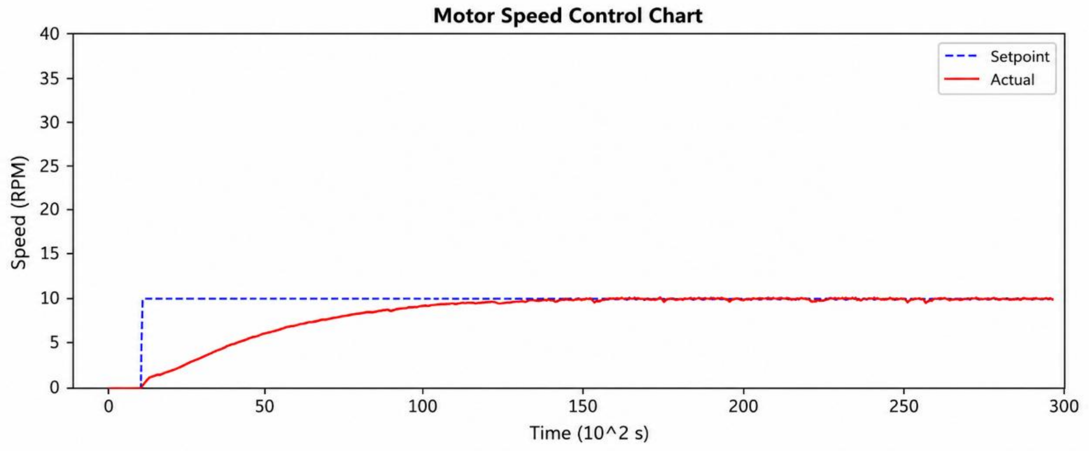

Fig. 7. System response at 10 RPM target speed

Analysis: The motor speed control system performs well at a target speed of 10 RPM. The actual speed steadily increases without overshoot and reaches the setpoint within approximately 140 time units. The response curve is smooth, with minimal oscillations during the transient phase. Once stabilized, the speed remains close to the reference with very small steady-state error, indicating that the PID controller is well-tuned and provides reliable and stable performance under no-load conditions.

Fig. 8 illustrates the performance of the conveyor system operating at a target speed of 20 RPM using the optimized PID parameters.

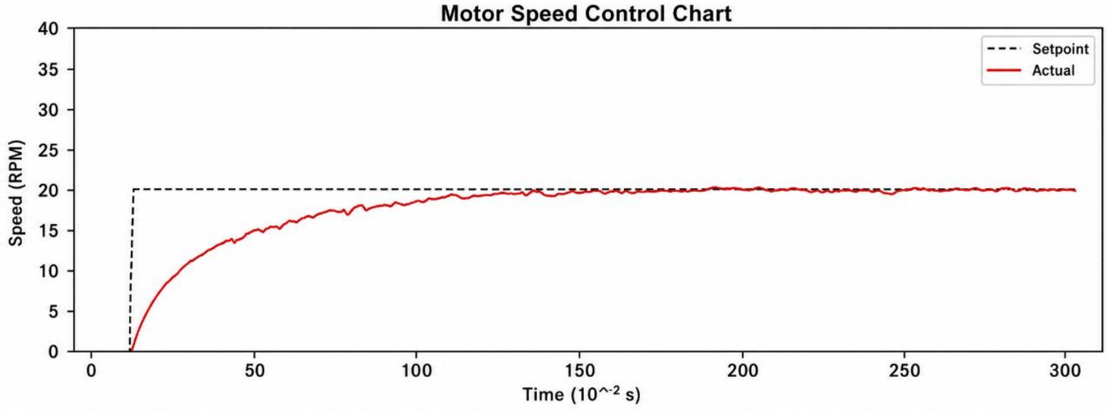

Fig. 8. System response at 20 RPM target speed

Analysis: The system response demonstrates effective speed control performance. The motor speed increases smoothly and approaches the setpoint of 20 RPM with a moderate rise time. Although a slight overshoot is observed, it quickly settles, and the system stabilizes around the desired speed after approximately 150 time units. Minor fluctuations appear in the steady-state phase but remain within an acceptable range, indicating high control accuracy and stability. The steady-state error is negligible, showing that the controller successfully maintains the reference speed even under higher-speed conditions. This confirms the robustness of the PID tuning and its ability to handle dynamic changes effectively.

Table 1. System Performance Evaluation at Different Speeds Using a PID Controller

Evaluation standards	**Speed (rpm)**		
	5	10	20
POT (%)	0	0	0
e_{xl} (rpm)	0	0	0
T_{xl} (s)	140	140	150
Tr (s)	110	120	120

Table 1 summarizes the dynamic performance of the conveyor speed control system at reference speeds of 5 rpm, 10 rpm, and 20 rpm. Across all test conditions, the PID controller achieves zero overshoot (POT = 0%), indicating that the system responds smoothly without exceeding the setpoint. This is a desirable characteristic for low-speed conveyor applications where abrupt acceleration may increase energy consumption or cause mechanical stress.

The steady-state error is consistently 0 rpm, demonstrating that the manually tuned PID gains are sufficient to eliminate long-term tracking error despite encoder noise and low-speed operation. This also confirms the effectiveness of the Kalman filter in stabilizing the feedback signal.

Regarding transient behavior, the settling time (Ts) ranges between 140–150 seconds, while the rise time (Tr) remains within 110–120 seconds. Although these values indicate relatively slow system response—common for low-power DC motors driving small conveyors—they remain stable and repeatable across all speed levels. The increase in Ts at 20 rpm (150 s) suggests a slight reduction in controller effectiveness under higher speed demand, but not enough to cause instability or tracking deviation.

Overall, the results indicate that the PID controller provides stable, overshoot-free, and accurate speed regulation across all tested conditions. The uniformity of the performance metrics also reflects the robustness of the control architecture for low-cost, energy-efficient conveyor applications.

4.2. Performance Evaluation under No-Load and Load Conditions

Fig. 9 illustrates the system operating at a speed of 25 RPM under no-load conditions.

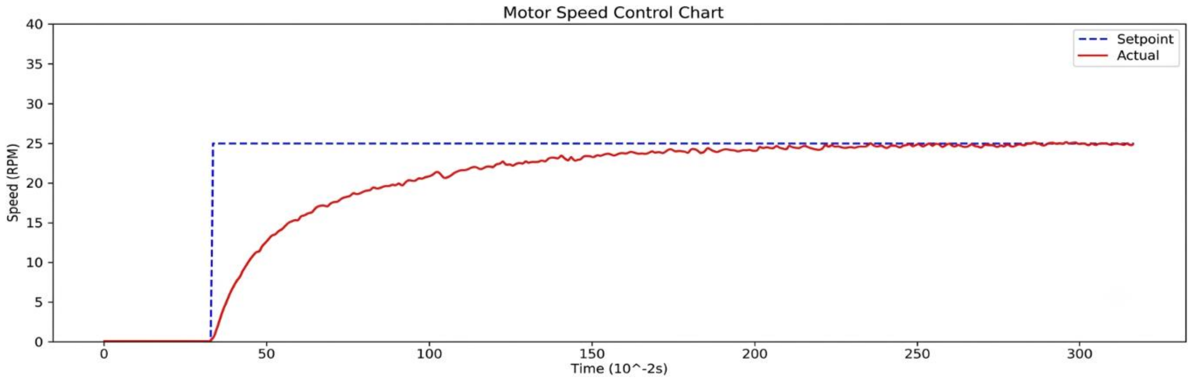

Fig. 9. Speed response at 25 RPM under no load

Fig. 10 illustrates the system operating at a speed of 25 RPM under a load of 0.5 kilograms.

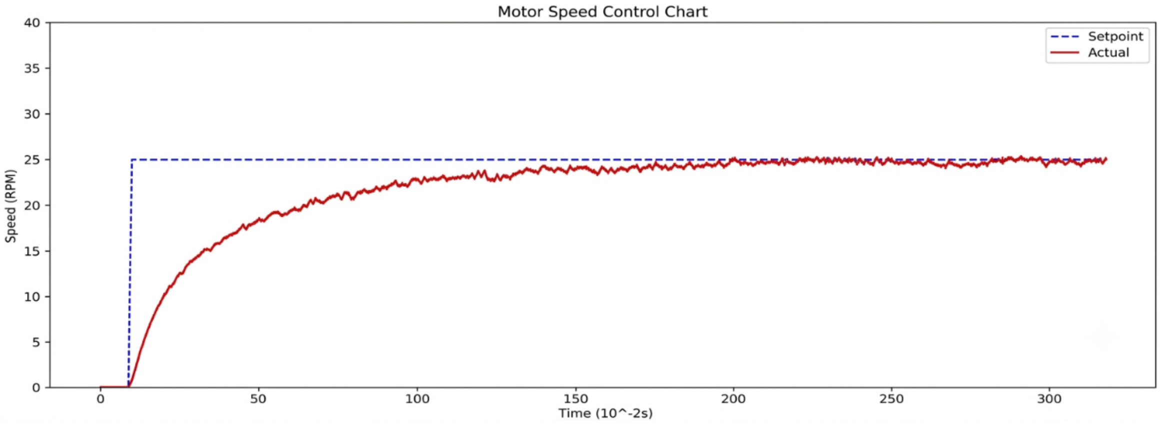

Fig. 10. Speed response at 25 RPM under load 0.5KG

Table 2. System Performance Evaluation at 25 RPM under No-Load and Load Conditions.

Evaluation standards	No-Load	Load (0.5KG)
POT (%)	0	0
e_{xl} (rpm)	0	0
T_{xl} (s)	170	210
Tr (s)	150	180

Analysis: At a target speed of 25 RPM, the system exhibits reliable and consistent performance under both no-load and loaded (0.5 kg) conditions. In the no-load scenario, the motor reaches the desired speed with a rise time of approximately 150 time units and stabilizes at around 170 units. When a 0.5 kg load is applied, the rise time increases slightly to 180 units, with the system settling at approximately 210 units. Despite this added load, the controller effectively maintains the setpoint without significant overshoot or steady-state error. The smooth convergence of the actual speed to the reference in both cases confirms the effectivenessof the PID control strategy in handling varying load demands while ensuring stability and accuracy.

Table 2 presents the system's performance when maintaining a fixed reference speed of 25 rpm under no-load and 0.5 kg load conditions. Similar to the previous experiments, the controller achieves zero overshoot (POT = 0%) and zero steady-state error, confirming that the PID parameters remain effective even when the system experiences additional mechanical load. This demonstrates good robustness and disturbance rejection capability.

When comparing dynamic performance, the introduction of a 0.5 kg load causes a noticeable increase in both settling time (Ts) and rise time (Tr). Specifically, Ts increases from 170 s to 210 s, and Tr increases from 150 s to 180 s. This behavior is expected, as the added load increases inertia and friction, requiring more time for the controller to accelerate the conveyor to the target speed. Despite this, the controller maintains a smooth response without oscillations, reflecting stable control behavior.

The results confirm that the proposed PID-based control system is capable of maintaining accurate speed tracking under varying operating conditions, with performance degradation under load remaining within acceptable limits. The ability to preserve zero overshoot and zero steady-state error under load conditions highlights the practicality of the design for energy-efficient and mechanically safe conveyor operation.

5. CONCLUSION

This study presented the development and experimental evaluation of a low-cost and energy-efficient conveyor speed control system based on a manually tuned PID controller implemented on an Arduino Mega platform. Using encoder feedback together with a Kalman filter for noise reduction, the system achieved stable and precise speed regulation across multiple reference levels. The controller exhibited zero overshoot, negligible steady-state error, and consistent transient behavior at 5, 10, and 20 rpm under no-load conditions. Additional tests at 25 rpm demonstrated that the system maintained accurate tracking even under a 0.5-kg load, confirming good robustness against external disturbances.

The results highlight that simple, well-tuned PID control remains an effective solution for small conveyor systems, particularly where energy efficiency and mechanical reliability are required. By minimizing unnecessary acceleration and maintaining steady operation, the proposed approach contributes to reducing power consumption and mechanical wear—factors directly relevant to sustainable material-handling processes.

Overall, the work demonstrates the feasibility of constructing a reproducible, low-cost, and environmentally conscious speed control system using commonly available components. Future research may explore automatic PID tuning methods, adaptive or intelligent control strategies, and integration with industrial monitoring frameworks to further enhance performance and applicability in modern energy-aware manufacturing environments.

ACKNOWLEDGEMENT

The research was funded by Ho Chi Minh City University of Technology and Engineering, Vietnam, under grant No. SV2026-242. We want to give thanks to Assoc. Prof. Minh-Tam Nguyen (HCM-UTE) due to his supervision for this sudy. We, authors, are grateful for these supports.

REFERENCES

[1] T. L. Sime, P. Aluvada, S. Habtamu, and Z. Tolosa, “Modeling of genetic algorithm tuned adaptive fuzzy fractional order PID speed control of permanent magnet synchronous motor for electric vehicle,” SN Applied Sciences, vol. 6, p. 83, 2024, doi: 10.1007/s42452-024-06183-8.

[2] E. W. Suseno and A. Maarif, “Tuning of PID controller parameters with genetic algorithm method on DC motor,” IJEA, vol. 1, no. 1, pp. 41–53, 2021.

[3] L. Lam, “PID Controller Human Detection Robot With Visual Processing on Alphabot-2,” International Journal of Engineering Science, vol. 6, no. 2, 2025, doi: 10.51630/ijes.v6i2.176.

[4] J. Sobota and M. Schlegel, “Iterative feedback tuning of PID controller,” Process Control 2004 Conference, Czech Republic, pp. 1–16.

[5] K. Patel and J. Patel, “PID based speed control of DC motor using Arduino microcontroller,” IJEDR, vol. 6, no. 2, pp. 389–392, 2018.

[6] A. Kumar, “Speed control of DC motor using PID and Fuzzy Logic controller,” IJETAE, vol. 4, no. 3, pp. 61–67, 2014.

[7] S. Sahoo, P. Mishra, and R. Nayak, “Load disturbance modeling and control of belt conveyor drive systems,” Measurement, vol. 190, p. 110686, 2022, doi: 10.1016/j.measurement.2022.110686.

[8] H. Mohammed, A. Bakar, and M. Ismail, “Energy-efficient speed control of industrial conveyor systems using optimized DC motor drive,” Energy Reports, vol. 6, pp. 198–207, 2020, doi: 10.1016/j.egyr.2019.12.020.

[9] M. Ibrahim, R. Ahmed, and S. Osman, “Design and implementation of an Arduino-based conveyor control system,” IOP Conf. Ser.: Materials Science and Engineering, vol. 1212, p. 012014, 2022, doi: 10.1088/1757-899X/1212/1/012014.

[10] R. Santos and M. Alves, “Embedded PID control of DC motors using low-cost microcontrollers,” Electronics, vol. 10, no. 4, p. 451, 2021, doi: 10.3390/electronics10040451.

[11] N. Zhao, H. Wang, and T. Guo, “Noise-resilient encoder speed estimation using Kalman filtering,” Sensors, vol. 21, no. 12, p. 4021, 2021, doi: 10.3390/s21124021.

[12] R. Fadil, S. Khalid, and A. Ramli, “Low-cost sensor filtering techniques for DC motor speed measurement,” Measurement and Control, vol. 55, no. 3–4, pp. 365–374, 2022, doi: 10.1177/00202940211066056.

[13] M. Andrejiova, A. Grincova, and D. Marasova, “Monitoring dynamic loading of conveyor belts by measuring peak impact forces,” Measurement, vol. 158, p. 107690, 2020, doi: 10.1016/j.measurement.2020.107690.

[14] K. Patel and J. Patel, “Energy consumption characteristics of industrial conveyor drives,” Energy, vol. 214, p. 118858, 2021, doi: 10.1016/j.energy.2020.118858.