

DC Motor Speed Control and Stabilization using PID Turning Methods (A Comparative analysis of the Ziegler-Nichols (ZN) closed-loop resonance method and the Chien–Hrones–Reswick (CHR) tuning method)

Oyindinpre Bioko¹, Damfebo Franklin Ayebagbalinyo², Fawei Jenny Ebitonere³, Anyalewechi Chika Juliana⁴

1 Department of Electrical and Electronic Engineering, Faculty of Engineering, Niger Delta University, Amassoma, Wilberforce Island, Bayelsa State, Nigeria.
oyins.bioko@ndu.edu.ng

2 Department of Electrical and Electronic Engineering, Faculty of Engineering, Niger Delta University, Amassoma, Wilberforce Island, Bayelsa State, Nigeria.
damfebofranklin@ndu.edu.ng

3 Department of Electrical and Electronic Engineering, Faculty of Engineering, Niger Delta University, Amassoma, Wilberforce Island, Bayelsa State, Nigeria.
faweijenny@ndu.edu.ng

4 Department of Electrical and Electronic Engineering,
Federal University of Technology Owerri, Imo State, Nigeria
chika.anyalewechi.futo.edu.ng

Abstract: Efficient speed control of Direct Current (DC) motors is a fundamental requirement in industrial automation, robotics, and precision engineering. This study presents a comparative performance analysis of two distinct Proportional-Integral-Derivative (PID) tuning strategies: the Ziegler-Nichols (ZN) closed-loop resonance method and the Chien–Hrones–Reswick (CHR) tuning method. While the ZN method remains a benchmark for its ability to achieve aggressive disturbance rejection through sustained oscillation analysis, it often results in high overshoot and oscillatory behavior. In contrast, the CHR method—an evolution of the ZN open-loop technique—is implemented to provide a more tailored response, offering specific tuning parameters for either set-point tracking or disturbance rejection with varying degrees of overshoot (0% vs. 20%). The system was implemented using an experimental setup, the DC motor transfer functions were modelled, and both tuning algorithms were simulated within a MATLAB/Simulink environment. Performance is evaluated based on key temporal parameters, including rise time, settling time, peak overshoot, and steady-state error. Preliminary results indicate that while the ZN method provides a faster initial response, the CHR-tuned PID controller offers smoother operation with enhanced stability, making it superior for applications where mechanical stress from overshoot must be minimized. The findings provide practical guidance for selecting suitable PID tuning methods for DC motor speed control applications based on specific performance requirements.

Keywords—component; DC Motor Control, PID Tuning, Ziegler-Nichols, Chien–Hrones–Reswick

1. INTRODUCTION

A DC motor is a common electromechanical device that converts electrical energy into mechanical motion using direct current. Because of its simplicity, reliability, and ease of control, DC motors are widely used in industrial and technological applications, like automotive systems, robotics, household appliances, and medical equipment [1].

In real operating environments, factors such as load changes, supply voltage fluctuations, and parameter uncertainties can negatively affect motor performance. Effective speed control of DC motors is essential to ensure high performance, stability, and energy efficiency. As a result, developing control strategies that can maintain a stable and

precise motor speed under varying conditions has become an important area of research in both academia and industry [2]. Among various control strategies, the Proportional–Integral–Derivative (PID) controller remains the most commonly adopted because of its simple structure, ease of tuning, low computational requirements, and proven reliability in real-world applications [3]. For this reason, PID control is adopted.

The Proportional–Integral–Derivative (PID) tuning techniques commonly employed for DC motor speed regulation are the Ziegler–Nichols (ZN) closed-loop resonance method and the Chien–Hrones–Reswick (CHR) tuning method [4]. The ZN closed-loop resonance method determines PID parameters by increasing the proportional gain until the closed-loop system exhibits sustained oscillations, from which the ultimate gain and oscillation period are obtained. These

parameters are then used to compute the controller gains, typically resulting in a fast but aggressive system response. In contrast, the CHR tuning method is based on the process reaction curve obtained from the open-loop step response of the system and defines PID parameters using the process gain, time constant, and dead time, with the objective of achieving controlled transient behavior such as zero or limited overshoot [5].

The main objective of this research is to design, implement, and evaluate a closed-loop DC motor speed control system using PID control on a Siemens S7-1200 PLC, focusing on accurate speed measurement using an incremental rotary encoder, systematic PID tuning using the Chien–Hrones–Reswick (CHR) tuning method and Ziegler–Nichols (ZN) closed-loop resonance method and detailed their performance analysis under varying operating conditions.

2. PID CONTROLLER OVERVIEW

The PID controller is a classical control algorithm extensively used for motor speed regulation. It continually adjusts the control signal based on three terms:

- Proportional (P): Reduces present error.
- Integral (I): Eliminates steady-state error.
- Derivative (D): Predicts future error trends.

These actions provide a balanced response in terms of speed regulation, system stability, and disturbance rejection [6].

2.1 PID Control Law

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{d e(t)}{dt} \quad (1)$$

Where $e(t)$ is the error between the reference speed and the actual speed [7].

2.2 PID Tuning Methods

The earliest and simplest PID tuning approaches is the trial-and-error method, as discussed in [8]. This technique involves manually adjusting the proportional, integral, and derivative gains while observing the system response. The gains are iteratively modified until an acceptable dynamic behavior is achieved. Although straightforward, this method can be time-consuming and highly dependent on the experience of the operator.

As control systems evolved, John Ziegler and Nathaniel Nichols proposed a landmark method for PID tuning applicable to both open-loop and closed-loop systems, based on the step response characteristics of the plant [9]. In the closed-loop Ziegler–Nichols method, the controller is first operated in proportional-only mode to determine the ultimate gain K_{uand} and the corresponding oscillation period T_{u} . These parameters are then used to compute the PID gains K_p , K_i , and K_d . The open-loop variant, on the other hand, relies on estimating the process time delay and time constant from the system’s step response [10].

2.3 Ziegler–Nichols Closed-Loop (Ultimate Gain) Method

The Ziegler–Nichols Closed-Loop Resonance Tuning Method is an experimental technique for setting PID controller parameters by finding the point where the system naturally oscillates, called the ultimate gain (K_u) and ultimate period (P_u). This is done by disabling integral/derivative actions, increasing proportional gain (K_p) until stable, continuous oscillations occur (the resonance point), and then using K_u and P_u with Z-N formulas to calculate initial PID settings, providing a good starting point for further fine-tuning. The closed-loop (ultimate gain) method is used when the process can be operated under feedback control [11].

The proportional gain (K_p) is increased continuously in a feedback loop until the system output exhibits sustained oscillations (no growth or decay). The gain at which this occurs is called the ultimate gain (K_u). The period of oscillation is called the ultimate period (T_u).

Steps for the Closed-Loop Method

- Stabilize the Loop: Ensure the process is stable and in automatic mode.
- Disable Integral & Derivative: Set integral time (T_i) to maximum and derivative time (T_d) to zero (or disable these modes).
- Introduce Disturbance: Make a small step change to the set point.
- Find Ultimate Gain (K_u): Gradually increase the proportional gain (K_p) until the system's output shows sustained, constant-amplitude oscillations (resonance). Record this gain as the ultimate gain (K_u).
- Find Ultimate Period (P_u): Measure the time between peaks of these sustained oscillations to find the ultimate period (P_u).
- Calculate PID Settings: Use (K_u) and (P_u) with the Ziegler–Nichols formulas to find initial K_p , T_i , and T_d values for the PID controller [12]. Table 1 described the Closed-Loop PID Gains for Ziegler–Nichols Closed-Loop Method [13]

Table 1. The Closed-Loop PID Gains for Ziegler–Nichols Closed-Loop Method [13]

Controller Type	(K_p)	(T_i)	(T_d)
PID	0.6 (K_u)	0.5 (T_u)	0.125 (T_u)

2.4 PID Chien–Hrones–Reswick (CHR) Tuning Method

The Chien–Hrones–Reswick (CHR) method is an empirical PID tuning technique for DC motor speed control that uses open loop step response parameters (gain K_s , delay t_u , rise time t_d) to calculate PID gains for desired performance, offering specific formulas for no-overshoot or 20% overshoot, resulting in good setpoint tracking making it useful for achieving precise control with less oscillation in applications like motor speed regulation. Although, it requires

a good open-loop step response, which might be harder to get in some processes, it Provides more stable and less oscillatory response, this is because it allows tuning for specific overshoot requirements, which is useful for precision control [14].

This Works by applying a step change to the motor's input (e.g., voltage) and measure the output (speed) to find the system's open-loop response, extract the three key parameters which are; the Steady-state gain (ks) (Ratio of output change to input change), Time delay (tu) which is the time until the response starts, and the Rise time (td) which is the time from 10% to 90% of the final value, then use CHR formulas to calculate PID parameters [15].

Table 2: Comparison between Ziegler–Nichols and CHR methods

Feature	Ziegler–Nichols (Closed-Loop)	CHR Method
Approach	Closed-loop, based on oscillation	Open-loop, based on step response
Goal	Quick, aggressive response	Controlled, specific overshoot [16]
Safety during tuning	Requires tolerance of oscillations	Safer (step response can be done slowly)
Overshoot control	high	User-selective (0% or 25% options) [17]
Complexity	Simple	More calculations required

3. REVIEW ON RELATED WORK

Recent works confirm the effectiveness of PID controllers for DC motor speed control. Authors in reference [18] developed an Arduino dc motor speed control with PID Controller using simulation and experimental studies which verified that PID control can improve mean error by over 85 % in speed regulation scenarios.

A specially designed PID algorithm implemented on microcontrollers achieved satisfactory steady-state performance with PWM and encoder feedback was designed and implemented in reference [19]

Optimal PID controller tuning for DC motor speed control using metaheuristic methods such as smell agent optimization algorithm have been applied to PID gains, yielding better

overshoot and settling time characteristics than classical tuning for DC motor speed control as in ref [20].

Reference [21] analyzed DC motor speed control using PID controllers tuned with the Ziegler–Nichols (ZN) and Chien–Hrones–Reswick (CHR) methods by developing a mathematical to evaluate the performance of both tuning techniques. The ZN-tuned PID controller produced a faster response but exhibited higher overshoot and oscillations, while the CHR-tuned controller achieved smoother speed regulation with improved stability and reduced overshoot. The study concluded that the choice of tuning method should depend on whether fast response or stable operation is the primary control objective.

4. METHODOLOGY

4.1 Description of Hardware Used

Figure 1 is the fulcrum of the system is the Siemens SIMATIC S7-1200 PLC (CPU 1215C DC/DC/DC), which is responsible for receiving feedback and issuing control signals. The PLC acquires input signals from the incremental encoder as well as the start and stop push buttons, processes the control logic, including the PID algorithm, and generates analog output signals for the motor controller. Equipped with 24 V DC power supply, 14 digital inputs, 4 analog outputs, high-speed counter capability, and integrated PID function blocks.

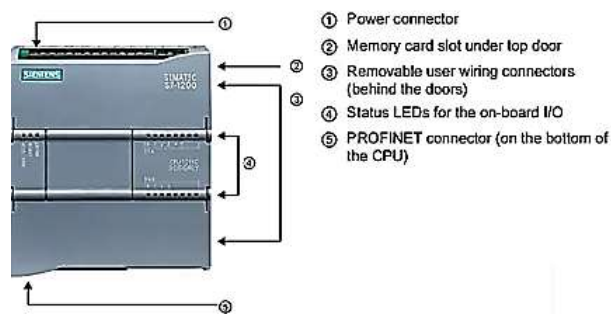


Figure 1. Siemens S7 1200 [22]

The DC motor (CM 82869012) serves as the primary actuator in the system. It operates within a voltage range of 5–24 V and has a nominal maximum speed of 5000 rpm. An integrated gearbox with a reduction ratio of 80:1 lowers the output shaft speed to approximately 62.5 rpm, making the motor suitable for low-speed, high-torque applications. Since the motor speed is directly proportional to the applied supply voltage, effective speed control is achieved through voltage modulation.

Motor speed and rotational direction are measured using a 600 PPR incremental rotary encoder (E38S6G5-600B-G24N). The encoder provides quadrature signals via channels A and B, which are interfaced with the PLC's high-speed inputs. It supports rotational speeds of up to 5000 rpm and operates at a maximum frequency of 100 kHz. The encoder is powered directly from the PLC, ensuring seamless integration and

reliable feedback. The rotary encoder (E38S6G5-600B-G24N) is displayed in Figure 2.



Figure 2. Rotary encoder (E38S6G5-600B-G24N) [23]

The motor controller (RS 313-2122) functions as the power interface between the PLC and the DC motor. It converts the PLC’s analog control output into the appropriate voltage and current required to drive the motor, delivering up to 24 V and 60W as needed to reach the desired speed. Supporting components include pull-up resistors (4.7 kΩ for each encoder channel and 10 kΩ for the motor controller) to maintain proper logic levels, as well as push buttons for system operation, green for start and red for stop. Together, these components form a complete and robust PLC-based DC motor speed control system. The complete setup was described in the first part of this work as in reference [24]

4.2 Software Implementation (TIA Portal)

- System programming and configuration were carried out using Siemens TIA Portal 17 version
- HSC Setup: the High-Speed Counter in the PLC properties is configured to "Frequency" mode to measure speed directly

4.3 Ziegler–Nichols (ZN) closed-loop resonance method

The block diagram shown in Figure 3, illustrates the Ziegler’s closed-loop feedback system for DC speed control. It begins with a summing point that calculates the error by subtracting the actual speed (measured by a tachometer or encoder) from the desired set-point speed. This error signal is processed by the PID controller, which applies proportional, integral, and derivative gains to generate a corrective control signal. This signal is typically sent through a power amplifier to drive the DC motor, and adjusting its voltage to counteract variations. The loop is continuously updated, ensuring the motor maintains the target speed despite external load disturbances or internal parameter changes.

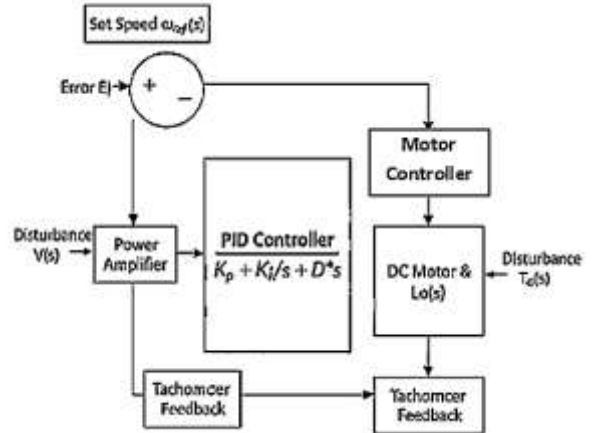


Figure 3. block diagram of Ziegler–Nichols (ZN) closed-loop resonance method

4.4 Flow chart for Ziegler–Nichols (ZN) closed-loop resonance method

The flowchart (figure 4), outlines the systematic procedure for the Ziegler–Nichols closed-loop tuning process, beginning with the system in a purely proportional state where integral and derivative gains are set to zero. The process follows an iterative loop where the proportional gain K_p is gradually increased until the motor’s speed output reaches a state of sustained oscillation with a constant amplitude. The Ultimate Gain K_u , and the Ultimate Period P_u of the oscillations will be identified and recorded. The final stage involves applying these two experimental values to the standard ZN formulas to calculate the specific K_p , K_i and K_d parameters, effectively transitioning the controller from a manual tuning phase to an optimized PID configuration.

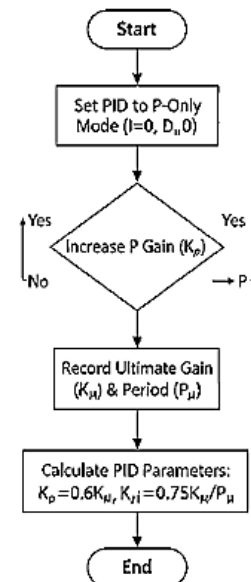


Figure 4. flow chart for Ziegler–Nichols (ZN) closed-loop resonance method

4.5 Chien–Hrones–Reswick (CHR) tuning method

This method is based on the system's Open-Loop Step Response. This was achieved by applying a step change to the motor's input and measure the output (speed) to find the system's open-loop response, the three key parameters which are such as the Steady-state gain (ks), Time delay (tu), and the Rise time were extracted, then the PID parameters were calculated. The block diagrams for the Ziegler–Nichols (ZN) and Chien–Hrones–Reswick (CHR) methods appear identical because they both use the same PID Control Architecture.

4.6 Flow chart for the Chien–Hrones–Reswick (CHR) tuning method

The flowchart (figure 5) describes the systematic, step-by-step process begins with an open-loop step response test of the DC motor. this method starts by applying a constant voltage to the motor and recording the resulting speed until it reaches a steady state. the user identifies three critical parameters: the process gain (K), the delay time (L), and the time constant (τ). The next decision point in the flowchart. next involves selecting the specific control objective (set-point) and choosing a desired damping level, categorized as 0% overshoot or 20% overshoot. Finally, the process parameters are multiplied by specific CHR constants to determine the optimal Proportional (Kp), Integral (Ki), and Derivative (Kd) values for the final controller implementation.

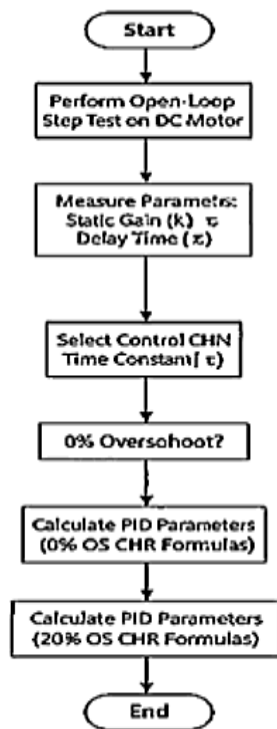


Figure 5. Flow chart for the Chien–Hrones–Reswick (CHR) tuning method

4.7 Description of PLC Control Program

The PLC control software is structured into four distinct program units. These include a Data Block, which defines and stores all program tags and variables; a Function Block, responsible for computing motor speed; a Cyclic Interrupt Block, which executes the speed computation and PID control logic at a defined interval; and a Main Block, which coordinates and supervises the overall program operation.

- Counter-to-Speed Function Block: This function block forms the initial stage of the control program. It acquires encoder pulse signals from the PLC analogue input port addressed as %ID1000 and converts these pulses into rotational speed using the standard relationship:

$$\text{Speed (RPM)} = \frac{\text{No. of Encoder Pulses Per Second} \times 60}{\text{Encoder Resolution}} \quad (2)$$

During each execution cycle, the current counter value is transferred to a memory variable named Record Counter. In the subsequent cycle, the difference between the new counter value (Current Counter) and the stored value (record Counter) is computed to determine the number of pulses within the cycle. Since the program operates with a cycle time of 200 ms, the measured pulse count is first scaled to a one-second equivalent (1000 ms) before applying Equation (2).

- Cyclic Interrupt Block: This program unit executes the counter-to-speed function block at a predefined cycle time and manages the real-time control process. The computed motor speed, expressed in revolutions per minute (rpm), is stored in a variable labeled Motor Speed. This variable serves as the process feedback input to the PID controller.

The PID controller, implemented as a technology object in the Simatic S7 environment, is highly versatile and requires appropriate configuration for specific applications. Key configuration settings adopted in this paper include:

The controller type is configured for speed control, with rpm selected as the unit of measurement.

The control action is set to inverted, and the CPU operating mode is configured as Automatic to enable continuous autonomous operation.

Two input source options are available: *Input* for signals originating from internal memory locations, and *Input_PER* for signals obtained directly from PLC hardware. As shown in Figure 6, the *Input* source is selected in this paper because the motor speed signal is an internal variable. On the output side, *Output_PER* is employed as the controller output is transmitted directly to the PLC hardware.

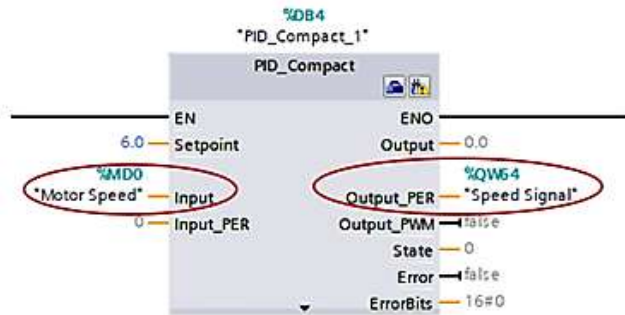


Figure 6: PID Block showing input and output types

5. RESULTS AND DISCUSSION

5.1 Current–Speed Relationship

The PLC generates an analog output current signal in the range of 0–20 mA. Within the TIA Portal environment, this analog signal is internally represented as a digitized integer value ranging from 0 to 27,648 to facilitate software processing. To examine the relationship between the PLC output current and the DC motor speed, selected integer values within this range were manually applied to the motor controller, and the corresponding motor speeds were recorded. A graph of motor speed versus output current was then plotted for analysis. This is displayed in Figure 7.

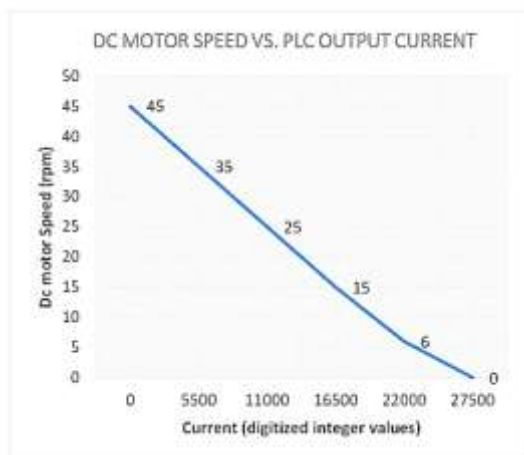


Figure 7: Plot of DC Motor Speed vs. PLC Output Current

In this representation, an integer value of 0 corresponds to 0 mA, while a value of 27,648 represents the full-scale output of 20 mA. Experimental observations revealed that the motor speed varies inversely with the PLC output current. This behavior arises because the PLC’s analog output does not directly supply sufficient power to drive the DC motor. Instead, the signal is first routed to a motor controller, which serves as a power amplifier and converts the low-power control signal into an appropriate voltage and current level for the motor. Consequently, the effective relationship between the PLC output current and motor speed is governed by the

internal design and signal processing characteristics of the motor controller.

This inverse relationship between the control signal and the resulting motor speed was carefully accounted for during PID controller configuration. Specifically, the control logic was inverted by selecting the *invert control logic* option in the PID configuration window, ensuring correct controller action and stable speed regulation.

5.2 Results Using PID Controller

Table 3 summarizes the PID gain parameters obtained using the Chien–Hrones–Reswick (CHR) and Ziegler–Nichols (ZN) tuning methods. At the same time. The results highlight clear differences in controller behavior arising from the tuning philosophy adopted by each method.

Table 3: PID Gain Parameters obtained from each of the tuning methods.

Property	Chien-Hrones-Reswick (CHR)	Ziegler-Nichols (ZN)
Proportional gain (K_p)	0.33	0.80
Integral action time (T_i)	0.45	0.30
Derivative action time (T_d)	0.06	0.10
Derivative coefficient delay	0.1	0.085
Proportional action weighting	1.0	0.9
Derivative action weighting	0.0	0.0
Sampling time of PID algorithm	0.2	0.2

From Table 3, the CHR-tuned controller exhibits a relatively low proportional gain ($K_p = 0.33$) and longer integral action time ($T_i = 0.80$ s), together with a moderate derivative action ($T_d = 0.30$ s). These parameter values are consistent with the CHR method’s emphasis on reduced overshoot and improved disturbance rejection. The smaller proportional action weighting and the inclusion of derivative action weighting further contribute to smoother transient behavior and enhanced stability. As reflected in Figure 8, the CHR-tuned response demonstrates a gradual rise to the setpoint with minimal oscillation, making it suitable for applications where smooth speed regulation is prioritized over rapid response.

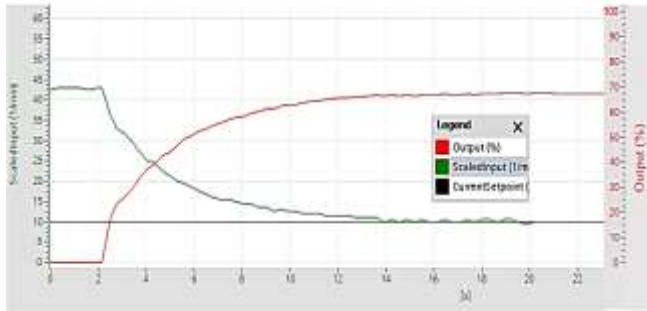


Figure 8: DC Motor Speed Response using Chien-Hrones-Reswick Tuning Method

The Ziegler–Nichols tuning method yields a more aggressive controller configuration. The shorter integral action time ($T_i = 0.45$ s) and shorter derivative time ($T_d = 0.06$ s), combined with full proportional action weighting, results in faster error correction and a more responsive control action. However, the absence of derivative weighting and the relatively higher sensitivity of the gains increase the likelihood of overshoot and oscillatory behavior. This is shown in Figure 9, where the ZN-tuned response reaches the setpoint more quickly but with noticeable overshoot and transient oscillations before settling.

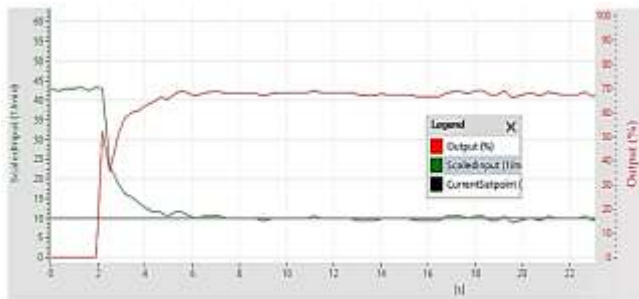


Figure 9: DC Motor Speed Response using Ziegler-Nichols Closed-Loop Resonance Method

Figure 8 and Figure 9 depict the dynamic speed response of the DC motor when the PID controller is tuned using the Chien–Hrones–Reswick (CHR) method and the Ziegler–Nichols (ZN) closed-loop resonance method, respectively. In both figures, the reference speed (setpoint) is fixed at 10 rpm, indicated by the black horizontal line. The motor speed response is shown by the green curve (Scaled Input), while the red curve represents the controller output expressed as a percentage of the maximum available output current (20 mA), with its corresponding scale shown on the right-hand axis.

For both tuning approaches, the controller output stabilizes at approximately 67.5%, indicating that this level of output current is required to maintain the motor at the desired speed of 10 rpm. The fact that the motor speed converges accurately to the setpoint in both cases confirms the correctness of the control design, signal scaling, and PID configuration.

5.3 A comparison between the two tuning methods.

With the Ziegler–Nichols tuning, the motor exhibits a much faster response, achieving the setpoint with a rise time of approximately 5 seconds and settling within about 6 seconds. However, small oscillations of roughly $\pm 0.5\%$ are observed around the steady-state value. These fluctuations are primarily attributed to minor disturbances and noise in the encoder feedback signal, which are amplified by the more aggressive gain settings characteristic of the ZN method. The CHR-tuned response demonstrates smoother and more conservative behavior. Although the rise time (approximately 12 seconds) and settling time (around 14 seconds) are significantly longer, the speed response shows virtually no oscillations at steady state. This confirms the CHR method’s strength in disturbance rejection and overshoot minimization, making it well-suited for applications where smooth operation and stability are more critical than speed of response.

The observed behaviors are consistent with the PID gain parameters presented in Table 3. The CHR method employs a smaller proportional gain (K_p) and derivative action time (T_d), along with a larger integral action time (T_i), resulting in a well-damped response with minimal oscillations. Conversely, the Ziegler–Nichols method uses relatively higher proportional and derivative gains and a shorter integral time, which explains its faster response but increased sensitivity to disturbances. The larger integral action time in the CHR method also contributes to effective elimination of steady-state error, in agreement with established PID control theory.

6. CONCLUSIONS

This study presents a comparative analysis of DC motor speed control using two classical PID tuning techniques: the Ziegler–Nichols (ZN) closed-loop resonance method and the Chien–Hrones–Reswick (CHR) tuning method. The ZN closed-loop resonance method determines controller parameters by increasing the proportional gain of the closed-loop system until sustained oscillations occur, from which the ultimate gain and oscillation period are obtained to calculate the PID gains.

The CHR tuning method is based on the open-loop process reaction curve and utilizes system parameters such as process gain, time constant, and dead time to compute PID gains, with the objective of achieving improved transient performance and reduced overshoot. The PID controller was implemented on a Siemens S7-1200 PLC, while the DC motor model and control algorithms were developed and simulated using MATLAB/Simulink, while system analysis and parameter extraction were carried out using built-in control and simulation toolboxes. Performance evaluation is conducted based on time-domain specifications including rise time, settling time, overshoot, steady-state error, and speed tracking accuracy.

The results show that the ZN-tuned PID controller provides a faster dynamic response but with higher overshoot and oscillatory behavior, whereas the CHR-tuned PID controller offers improved stability and smoother speed regulation. The

comparative analysis demonstrates that the choice of tuning method should be guided by the desired trade-off between response speed and transient performance in DC motor control applications.

7. REFERENCE

- [1] El-Sousy, F. F. M., & Abdel-Khalik, A. S. (2023). Adaptive control strategies for electric motor drives: A comprehensive review. *IEEE Access*, 11, 39412–39436. <https://doi.org/10.1109/ACCESS.2023.3267819>
- [2] Mohammed, O. A., & Liu, S. (2022). Intelligent control of DC motor drives: Trends and industrial challenges. *Energies*, 15(3). <https://doi.org/10.3390/en15031021>
- [3] Saputra, R. A., Nugroho, A., & Wibowo, S. (2022). PLC-based DC motor speed control using PID controller with MATLAB tuning. *Journal of Electrical Systems and Information Technology*, 9(1), 1–11. <https://doi.org/10.1186/s43067-022-00063-5>
- [4] Azeez, H. A., Mohd Hafiz A. Jalil @ Zainuddin, & Lukmon Owolabi Afolabi. (2024). DC motor speed control and real-time evaluation using PID controller. *Evolution in Electrical and Electronic Engineering*, 5(2), 180–189. <https://publisher.uth1m.edu.my/periodicals/i2ndex.php/eee/article/view/147536>
- [5] Subrata, R. H., Gozali, F., & Djuana, E. (2022). Computational and intelligent optimization tuning method for PID controller. *Journal of Theoretical and Applied Information Technology*, 100(7). Little Lion Scientific.
- [6] Al-Husainy, M. A. F., Al-Janabi, A. S., & Hasan, A. A. (2021). PLC-based temperature and level control system using PID algorithm. *Journal of Physics: Conference Series*, 1973, 012109. <https://doi.org/10.1088/1742-6596/1973/1/012109>
- [7] Hamoodi, S., Mohammed, R., & Salih, B. (2018). DC motor speed control using PID controller implementation by Simulink and practical. *International Journal of Electrical Engineering*, 11(1), 39–49.
- [8] Syahrini, S., Akmaludin, M., Syahrul Mubarak, M., & Anggorowati, A. A. (2025). Optimization of DC motor PID control using Ziegler–Nichols method with encoder and op-amp (LM324). *Academia Open*, 10(2), Article 12875. <https://doi.org/10.21070/acopen.10.2025.12875>
- [9] Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, 64, 759–768. <https://doi.org/10.1115/1.2899060>
- [10] Åström, K. J., & Hägglund, T. (2020). *PID controllers: Theory, design, and tuning* (3rd ed.). Instrument Society of America (ISA). <https://doi.org/10.1002/9781119549138>
- [11] Sailaja, V., and Nagabhushan K. R. (2019). PID Controller Tuning using Ziegler-Nichols Method for Temperature control of Thermal Cycler. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)*, Vol. 8, Issue 4. DOI:10.15662/IJAREEIE.2019.0804024 1289
- [12] Patel, Vishakha. (2020). Ziegler-Nichols Tuning Method: Understanding the PID Controller. *Resonance*. 25. 1385-1397. 10.1007/s12045-020-1058-z.
- [13] 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,” *Jurnal Teknologi Elektro*, vol. 14, no. 1, p. 23, Feb. 2023, doi: <https://doi.org/10.22441/jte.2023.v14i1.005>.
- [14] Soe, A. N., Myint, T., Htwe, P. M. M., & San, N. S. S. (2019). Design of PID controller for DC motor speed control system with Chien-Hrones-Reswrich method. *International Journal of Scientific Engineering and Technology Research*, 8, 581–584.
- [15] Hassan, A. A., Al-Shamaa, N. K., & Abdalla, K. K. (2018). *Comparative study for DC motor speed control using PID controller* [Unpublished student report]. Studocu. <https://www.studocu.com/row/document/aljamaa%D8%A9-altkny%D8%A9-alost%D9%89/electrical-technology/comparative-study-for-dc-motor-speed-con/99159102>
- [16] Åström, K. J. (2025). The evolution of heuristic tuning: From ZN to AI. *Journal of Industrial Process Control*, 42(3), 115–129.
- [17] Åström, K. J. (2025). The evolution of heuristic tuning: From ZN to AI. *Journal of Industrial Process Control*, 42(3), 115–129. *IEEE Industrial Electronics Magazine*. (2024). Comparative analysis of PID tuning for robotic actuators. *IEEE Industrial Electronics Magazine*, 18(2), 44–52.
- [18] Samsudin, S. I., Osman, K., Sulaiman, S. F., Salim, S. I. M., & Mohd Aliashak, N. I. (2024). Simulation and experimental study of Arduino DC motor speed control with PID controller. *Advanced and Sustainable Technologies (ASET)*, 3(2), 133–143. <https://doi.org/10.58915/aset.v3i2.1265>
- [19] Ahmad, H., & Zubair, A. (2023). To control the speed of a DC motor using a specially designed PID algorithm. *Pakistan Journal of Engineering and Technology*, 6(3), 25–33. Retrieved from <https://journals.uol.edu.pk/index.php/pakjet/article/view/256>
- [20] Abba, A. M., Karataev, T., Thomas, S., Ali, A. M., Yau, I., & Mikail, S. A. (2022). Optimal PID controller tuning for DC motor speed control using Smell Agent Optimization Algorithm. *FUOYE Journal of Engineering and Technology*, 7(1). <https://doi.org/10.46792/fuoyejet.v7i1.740>
- [21] Widagdo, R. S., Hariadi, B., & Setyadjit, K. (2023). Modelling and analysis of Ziegler–Nichols and Chien–Hrones Reswick tuning PID on DC motor speed control. *Jurnal Teknologi Elektro*, 14(1). 23 – 27

<https://publisher.uthm.edu.my/periodicals/index.php/eee/article/view/17536>

- [22] Aotewell. (2023). SIMATIC S7-1200 (Siemens S7-1200 PLC + HMI). <https://www.aotewell.com/siemens-automation/simatic-s7-1200-siemens-s7-1200-plchmi>
- [23] Fabian Enterprises Ltd. (2023). *Rotary encoder E38S6G5-400B-G24N*. <https://www.fabian.com.my/en/products/webshop/20127/rotary-encoder-e38s6g5-400b-g24n.htm>
- [24] Bioko, O., Ayebagbalinyo, D. F., Fawei, J. E., & Emmanuel, I. (2025). A PLC-based DC motor speed control system using Model Reference Adaptive Control Schemes (MRACS). *International Journal of Academic and Applied Research (IJAAR)*, 9(12), 291–298.