

Discrete-Time PID Speed Control for DC Motor via Z-Transform: Performance Analysis Across Simulation and Embedded Implementation Environments

Andicho Haryus Wirasapta^{1*}, Hamzah Alghifari², Muhammad Damas Fatih³, Mohammad Alfiza Rayesa⁴, Muhammad Taufik Ramadhoni⁵, Tiara Deta Pamungkas⁶

^{1,2,3,4,5}Department of Electrical and Informatics Engineering, Universitas Jambi, Jalan Raya Jambi - Ma. Bulian KM. 15 Mendalo Indah, Muaro Jambi, 36361, Indonesia

⁶Department of Nursing, Jambi Health Polytechnic, Ministry of Health, Jl. Dr. Tazar, Buluran Kenali, Jambi, 361225, Indonesia

Received: 12-05-2026

Accepted: 09-09-2026

Keywords:

DC Motor Speed Control;
Discrete-Time PID;
Z-Transform;
Microcontroller Simulation;
ZOH Discretization.

Correspondent Email:

andicho@unja.ac.id

Abstract. This paper presents the design and comparative performance analysis of a discrete-time Proportional-Integral-Derivative (PID) controller for DC motor speed regulation using the Z-transform method and Zero-Order Hold (ZOH) discretization. The principal contribution is a systematic analysis of controller behavioral differences between an ideal simulation environment (MATLAB/Simulink) and a microcontroller-based simulation platform (Tinkercad Arduino), revealing the quantitative impact of discretization effects on closed-loop dynamic response. System stability is rigorously verified through pole-zero analysis in the Z-plane. Performance evaluation encompasses time-domain metrics rise time, settling time, percent overshoot, and steady-state error supplemented by integral performance indices (ISE, IAE, and ITAE). The Tinkercad implementation yields an underdamped response ($t_r = 0.827$ s, $t_s = 13.74$ s, $M_p = 64.78\%$, $ess = 0$), while the Simulink simulation produces a critically damped response without overshoot. These discrepancies are quantitatively attributed to finite-precision arithmetic, computational delay, and PWM output resolution constraints inherent to the microcontroller platform. The findings underscore the critical necessity of implementation-aware evaluation in the development of reliable embedded control systems.

1. INTRODUCTION

Precise speed control of direct current (DC) motors is a fundamental requirement in a wide range of embedded control applications, including industrial automation, robotics, and electromechanical drive systems [1], [2]. The accelerating adoption of microcontroller-based digital platforms has intensified interest in discrete-time control strategies that are computationally compatible with the architecture of modern embedded processors [3], [4]. Among established control methodologies, the Proportional-Integral-Derivative (PID) controller remains the most widely deployed technique in industrial and

research settings, owing to its structural simplicity, interpretable tuning parameters, and demonstrated effectiveness across diverse plant dynamics [5]. The incorporation of encoder-based velocity feedback further enhances tracking accuracy, making digital PID control a natural choice for microcontroller-based motor drive systems [6], [7].

Despite its maturity, the direct digitization of a continuous-time PID design introduces non-trivial implementation challenges [8]. Discretization artifacts, finite sampling intervals, quantization errors, PWM resolution limits, and constrained computational throughput can substantially alter closed-loop behavior relative to

predictions obtained in ideal simulation environments [9], [10]. These effects frequently manifest as excess overshoot, sustained transient oscillations, and extended settling times performance degradation that is absent from continuous-time or high-precision numerical models [11], [12], [13].

Existing literature predominantly evaluates discrete-time PID controllers under ideal simulation conditions, using continuous-time approximations or high-fidelity numerical environments that do not capture the implementation-specific constraints of microcontroller hardware [14], [15], [16], [17], [18], [19]. The quantitative mechanisms by which finite-word-length arithmetic, computational delay, and actuator quantization alter closed-loop dynamics remain insufficiently characterized a gap with direct consequences for practical embedded system design [20], [21], [22], [23].

This paper addresses the foregoing gap through three principal contributions:

- a. A discrete-time PID controller is designed for DC motor speed regulation using the Z-transform method and ZOH discretization, with formal stability verification via pole-zero analysis in the Z-plane.
- b. A systematic comparative analysis of controller performance is conducted between an ideal simulation environment (MATLAB/Simulink) and a microcontroller-based simulation platform (Tinkercad Arduino), using time-domain performance metrics and integral performance indices (ISE, IAE, ITAE).
- c. Performance discrepancies between environments are quantitatively attributed to specific implementation factors finite-precision arithmetic, computational delay, and PWM resolution providing actionable insights for implementation-aware embedded controller design.

The remainder of this paper is organized as follows. Section II reviews the relevant literature on DC motor control, discrete-time PID design, and simulation-based evaluation methods. Section III details the system

modeling, discretization procedure, stability analysis, controller design, and implementation environments. Section IV presents and analyzes the performance results, including a cross-environment comparison and root-cause attribution. Section V concludes the paper and identifies directions for future research.

2. LITERATURE REVIEW

This section synthesizes prior work on four thematic pillars directly relevant to the present study: (i) DC motor modeling and speed control, (ii) PID controller design and tuning in discrete-time settings, (iii) Z-transform-based discretization methods, and (iv) simulation-based performance evaluation of embedded control systems. The review identifies the specific gap that motivates the contributions of this paper.

2.1. DC Motor Modeling and Speed Control

The DC motor has long served as a canonical benchmark plant for evaluating control strategies, owing to its well-characterized second-order electromechanical dynamics and wide industrial relevance [1]. Comprehensive reviews of DC motor speed control techniques spanning classical proportional control, state-space methods, and modern intelligent approaches consistently affirm the enduring applicability of linear feedback control for speed regulation in embedded motor drive systems [1], [2].

Kuczmanski [24] provided an extensive treatment of DC motor modeling and linear controller design, including both theoretical derivations and laboratory validation across multiple operating conditions. The study confirmed that second-order transfer function models offer a sufficient representation of motor dynamics for controller design purposes when motor parameters are well-characterized. Similarly, Mthboob et al. [1] conducted a systematic review of DC motor speed control architectures and concluded that encoder-based closed-loop feedback remains the standard approach for achieving precise speed regulation in embedded applications.

While high-performance alternatives such as field-oriented control (FOC) and direct torque control (DTC) are available for AC drives [2], the DC motor with classical PID control remains the preferred solution in cost-

sensitive, resource-constrained embedded systems particularly in educational and prototyping contexts due to its implementation simplicity and well-established theoretical foundation [6], [25].

2.2. PID Controller Design and Tuning Methods

The PID controller's dominance in industrial and academic applications is well documented. Li [5] and George and Ganesan [9] independently confirmed, through broad literature reviews, that PID remains the controller of choice for a majority of single-input single-output (SISO) process control applications, with its prevalence attributable to transparent parameter semantics and robust performance across a wide range of plant types.

Tuning methodology has attracted extensive research attention. Classical frequency-domain methods Ziegler-Nichols and Cohen-Coon remain widely used as initial design tools, with Huba et al. [12] demonstrating that systematic modifications of the Ziegler-Nichols approach yield significantly more reliable closed-loop responses under realistic operating conditions. Metaheuristic optimization has emerged as a prominent alternative: Joseph et al. [10] surveyed genetic algorithms, particle swarm optimization, and simulated annealing for PID tuning, reporting consistent improvements in transient performance over classical methods. More recently, Shahouni et al. [11] demonstrated that hybrid PSO-deep reinforcement learning approaches can adaptively tune fractional-order PID controllers for nonlinear plants, extending the performance envelope beyond classical integer-order designs.

Despite these advances, the majority of published tuning studies evaluate controller performance under ideal continuous-time or high-precision simulation conditions. The behavior of tuned PID parameters when deployed on finite-precision microcontroller hardware where gain quantization, derivative amplification of arithmetic noise, and integer rounding introduce implementation-dependent dynamics has received comparatively limited attention [9], [12], [13].

2.3. Discrete-Time PID and Z-Transform Digitization

The conversion of analog PID designs to discrete-time implementations via the Z-transform is a well-established procedure in digital control theory [26]. Among available discretization methods including Euler forward, Euler backward, Tustin (bilinear), and Zero-Order Hold the ZOH method is widely regarded as the most physically accurate for control applications, as it preserves the step response of the continuous-time system exactly when the control input is assumed constant between sampling instants [24], [26].

Nguyen et al. [27] applied genetic algorithms to optimize both continuous-time and discrete-time PID parameters for antenna azimuth position control, demonstrating that discrete-domain optimization consistently yields superior closed-loop performance compared to digitizing a continuously tuned controller. This finding underscores a fundamental design principle: controllers should ideally be tuned in the same domain in which they will be implemented. Pokharel [18] reached a similar conclusion in a broader survey of modern control strategies, noting that discrete-time formulation is indispensable for reliable embedded deployment.

Gorel et al. [28] implemented discrete-time PID control on a TMS320F28379D digital signal processor for a DC-DC buck converter, providing high-resolution experimental data that quantified the influence of sampling frequency, word length, and fixed-point representation on controller stability margins. Their results demonstrated that finite-word-length effects can reduce effective phase margin by 10-25° relative to ideal floating-point predictions a finding directly relevant to the present study's analysis of implementation-induced performance discrepancies.

2.4. Microcontroller-Based Simulation and Implementation Effects

Virtual simulation platforms including MATLAB/Simulink, LabVIEW, and browser-based tools such as Tinkercad have gained widespread adoption for rapid prototyping and pedagogical evaluation of embedded control systems [23], [29]. Wainer and Govind [20] traced the historical evolution of discrete-event simulation methodologies and emphasized that

the fidelity of simulation results is fundamentally bounded by the computational model's ability to replicate the timing and arithmetic behavior of the target hardware.

Anees et al. [23] conducted a comparative evaluation of microcontroller simulation platforms for engineering education, finding that Tinkercad accurately replicates the computational constraints of the Arduino Uno including integer arithmetic, PWM resolution, and execution timing within the bounds of its simulation model. This validation supports the use of Tinkercad as a meaningful proxy for embedded implementation behavior in contexts where physical hardware testing is not accessible.

Hobbs et al. [30] provided formal analysis of the impact of implementation platform timing uncertainties on the safety and stability of embedded controllers, demonstrating through hardware-in-the-loop experiments that computational delay equivalent to even one sampling period can shift closed-loop poles sufficiently to degrade stability margins measurably. Saha et al. [22] further characterized the challenge of deploying computationally demanding algorithms on microcontroller-class hardware, noting that fixed-point arithmetic introduces systematic coefficient errors that compound across multi-term control laws such as PID.

Taken together, the literature establishes a consistent finding: ideal floating-point simulation environments systematically overestimate the performance of controllers when deployed on constrained microcontroller platforms. However, quantitative attribution of specific implementation factors finite-word-length arithmetic, computational delay, and PWM quantization to specific degradation in PID transient performance metrics has not been systematically reported for DC motor speed control applications. This gap constitutes the primary motivation for the present study.

2.5. Research Gap and Positioning

Table 1 positions the present work relative to representative prior studies across the four thematic pillars reviewed above. The

comparison highlights that while previous work has addressed continuous-time PID design, discrete-time formulation, and simulation-based evaluation independently, a systematic study combining Z-transform discrete-time PID design, formal Z-plane stability verification, comprehensive ISE/IAE/ITAE performance indexing, and quantitative root-cause attribution of implementation effects within a controlled cross-environment comparative framework has not previously been reported for DC motor speed control.

Table 1. Positioning of the Present Study Relative to Representative Prior Works

Study	DC Motor Model	Discrete PID	ZOH Method
Kuczmam [24]	✓	X	X
Mthboob et al. [1]	✓	X	X
Nguyen et al. [27]	X	✓	✓
Gorel et al. [28]	X	✓	✓
Hobbs et al. [30]	X	✓	X
Anees et al. [23]	X	X	X
This Work	✓	✓	✓

3. RESEARCH METHODS

3.1. System Architecture

The proposed control system employs a closed-loop architecture comprising a reference speed input $r[k]$, a discrete-time PID controller $C(z)$, the DC motor plant $G(z)$, and an encoder-based velocity feedback path. At each sampling instant k , the tracking error $e[k] = r[k] - y[k]$ is computed and processed by the PID algorithm to produce the control signal $u[k]$, which drives the motor via a PWM interface. The full system architecture is illustrated in Fig. 1.

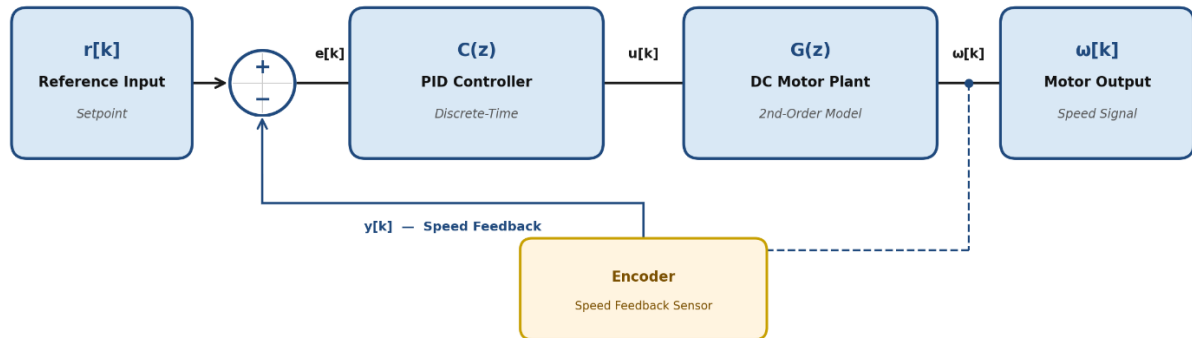


Figure 1. Block Diagram of the Closed-Loop Discrete-Time PID Control System for DC Motor Speed Regulation

3.2. DC Motor Modeling

The DC motor is modeled by coupling its electrical and mechanical subsystems. The electrical dynamics of the armature circuit, described by Kirchhoff's voltage law, are given by:

$$v(t) = L \frac{di(t)}{dt} + R \cdot i(t) + K_e \cdot \omega(t) \quad (1)$$

The rotational mechanical dynamics of the shaft, governed by Newton's second law, are expressed as:

$$J \frac{d\omega(t)}{dt} + B \cdot \omega(t) = K_t \cdot i(t) \quad (2)$$

where $v(t)$ is the applied armature voltage, $i(t)$ the armature current, and $\omega(t)$ the shaft angular velocity. Parameters R and L denote armature resistance and inductance; J and B the rotor inertia and viscous friction coefficient; K_e and K_t the back-EMF and torque constants, respectively. Applying the Laplace transform to Equations (1) and (2) yields the continuous-time transfer function:

$$G(s) = \frac{\Omega(s)}{V(s)} = \frac{K_t}{(J_s + B)(L_s + R) + K_t \cdot K_e} \quad (3)$$

Motor parameters are adopted from the standard benchmark reference of Ogata [26], selected to yield representative second-order dynamics. The complete parameter set is listed in Table 2.

Table 2. DC Motor Model Parameters (Ogata [26])

Parameter	Symbol	Value	Unit
Armature resistance	R	2.0	Ω
Armature inductance	L	0.5	H
Rotor inertia	J	0.02	$\text{kg} \cdot \text{m}^2$
Viscous friction coeff.	B	0.02	$\text{N} \cdot \text{m} \cdot \text{s}$
Torque constant	K_t	0.1	$\text{N} \cdot \text{m/A}$
Back-EMF constant	K_e	0.1	$\text{V} \cdot \text{s/rad}$

Substituting the parameter values from Table 1 into Equation (3) and simplifying yields the reduced continuous-time transfer function:

$$G(s) = \frac{0,1}{0,01s^2 + 0,04s + 0,05} \quad (4)$$

The open-loop poles are located at $s = -2 \pm j1$, confirming an inherently stable plant with continuous-domain damping ratio $\zeta_c = 0.894$.

3.3. ZOH Discretization and Sampling Period Selection

The ZOH method is selected for discretization because it accurately represents the sample-and-hold behavior of practical digital controllers, preserving the held-input response characteristic of embedded systems [24]. The sampling period $T = 0.01$ s corresponds to a sampling frequency $f_s = 100$ Hz. Justification is provided by the Shannon-

Nyquist criterion: the plant's natural frequency $\omega_n \approx 2.13 \text{ rad/s}$ ($\approx 0.34 \text{ Hz}$) requires a minimum sampling frequency of $f_{s,min} = 2 \times 0.34 = 0.68 \text{ Hz}$. The chosen $f_s = 100 \text{ Hz}$ exceeds this bound by a factor of 147, ensuring negligible aliasing and faithful discrete representation of the control signal dynamics [24], [31].

Applying ZOH discretization to Equation (4) with $T = 0.01 \text{ s}$ yields the discrete-time plant model:

$$y[k] = 1.82y[k-1] - 0.83y[k-2] + 0.095u[k] \quad (5)$$

This difference equation is directly implementable on digital hardware and serves as the plant model throughout the evaluation.

3.4. Discrete-Time PID Controller Design

The discrete-time PID control law, derived from backward-difference approximation of the continuous-time algorithm, is expressed as:

$$u[k] = K_p \cdot e[k] + K_i \cdot T \cdot \sum e[k] + \frac{K_d}{T} (e[k] - e[k-1]) \quad (6)$$

where K_p , K_i , and K_d are the proportional, integral, and derivative gains; $e[k]$ and $e[k-1]$ are the current and previous error signals. Taking the Z-transform of Equation (6), the discrete-time PID transfer function is obtained as:

$$C(z) = \frac{80.902 - 160.900z^{-1} + 80.000z^{-2}}{1 - z^{-1}} \quad (7)$$

Controller gains are determined through iterative time-domain tuning an approach appropriate for resource-constrained embedded targets where model-based optimization may be computationally prohibitive [27]. Tuning follows the classical sequential heuristic used for practical PID design: (i) with K_i and K_d set to zero, the proportional gain K_p is increased from zero in increments of 0.1 until the simulated closed-loop step response reaches an acceptably fast rise time without diverging or producing sustained oscillation under proportional action alone; (ii) the integral gain K_i is then introduced and increased in increments of 0.05 until the steady-state error is

driven to a negligible value, while monitoring for excessive overshoot or integral windup; (iii) the derivative gain K_d is subsequently introduced and increased in increments of 0.1 to improve transient damping. At every increment the resulting rise time, percent overshoot, and settling time are recorded from the simulated step response, and an adjustment is retained only if it improves rise time without an unacceptable increase in overshoot or settling time; iteration stops when no single-gain adjustment yields further net improvement under this criterion. Applying this procedure to the plant model of Equation (5) converges to the reported gains. The selected values $K_p = 0.9$, $K_i = 0.2$, $K_d = 0.8$ represent a deliberate trade-off that prioritizes fast rise time, accepting higher transient overshoot. The significance of this trade-off under finite-precision implementation constraints is examined in Section III.

3.5. Stability Analysis in the Z-Plane

Closed-loop stability is verified by checking that all poles of the discrete-time plant $G(z)$ lie strictly within the unit circle $|z| < 1$, the necessary and sufficient condition for asymptotic stability in discrete-time systems [26]. The characteristic equation of the plant difference equation (5) is:

$$z^2 - 1.82z + 0.83 = 0 \quad (8)$$

Solving Equation (8) yields the discrete-time plant poles:

$$z_{1,2} = 0.9100 \pm j0.0436; |z| = 0.9110 < 1 \quad (9)$$

Since $|z| = 0.9110 < 1$, both plant poles satisfy the stability criterion, confirming that the uncontrolled plant is inherently stable. This plant-only check is necessary but not sufficient to establish stability of the complete feedback system: the closed-loop pole locations are, in general, different from the plant poles and are governed instead by the full closed-loop characteristic equation, which is derived and solved explicitly in the following subsection. The discrete PID controller $C(z)$ contributes two zeros at $z \approx 0.9950$ and $z \approx 0.9927$, and two poles at $z = 0$ and $z = 1$. The integrator pole at $z = 1$ (on the unit circle boundary) is characteristic of a Type-I discrete controller,

which, under idealized infinite-precision arithmetic, eliminates steady-state error for step reference inputs. Whether this property is preserved under the finite-precision conditions of the microcontroller implementation is examined empirically in Section IV. The complete open-loop and closed-loop pole-zero configuration is shown in Fig. 2, following the closed-loop stability derivation below.

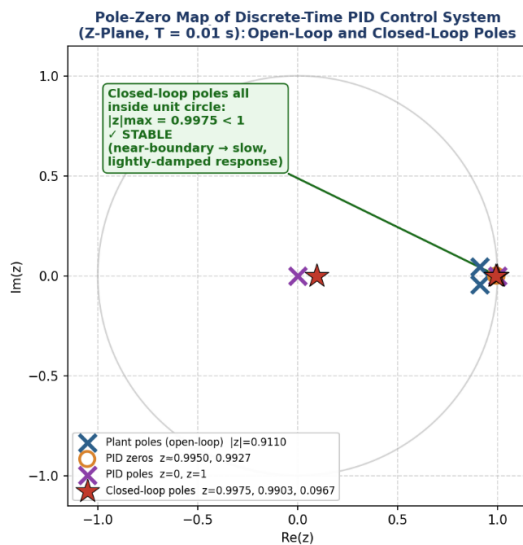


Figure 2. Pole-zero map of the discrete-time PID control system in the Z-plane, showing the open-loop plant poles ($|z| = 0.9110$), the PID controller zeros and poles, and the closed-loop poles obtained from the full characteristic equation $1 + C(z)G(z) = 0$ ($z = 0.9975, 0.9903, 0.0967$). All closed-loop poles satisfy $|z| < 1$, rigorously confirming closed-loop stability

3.6. Implementation Environments

The controller is evaluated in two computationally distinct environments:

- MATLAB/Simulink: Provides high-fidelity simulation using 64-bit IEEE 754 double-precision floating-point arithmetic, deterministic sampling, negligible computational overhead, and a continuous-equivalent PWM model. This serves as the ideal-case reference benchmark.
- Tinkercad (Arduino Simulation): Replicates the computational constraints of an Arduino Uno platform, including 8/16-bit integer arithmetic, fixed-point coefficient approximation, 8-bit PWM output (256 discrete levels), and non-zero execution overhead per control cycle. This represents the implementation-constrained case.

Both environments employ identical plant parameters, PID gains, and sampling period $T = 0.01$ s. Any observed performance differences are therefore attributable exclusively to implementation-level factors rather than design differences.

4. RESULTS AND DISCUSSION

This section presents the step response results, integral performance index evaluation, and root-cause analysis of cross-environment performance discrepancies. All results are obtained under identical initial conditions (zero initial state) with a unit step reference input.

4.1. Step Response: Tinkercad Microcontroller Environment

The step response of the proposed discrete-time PID controller in the Tinkercad environment is shown in Fig. 3. The system achieves a rise time $tr = 0.827$ s, reflecting the fast initial transient driven by the proportional and derivative control actions.

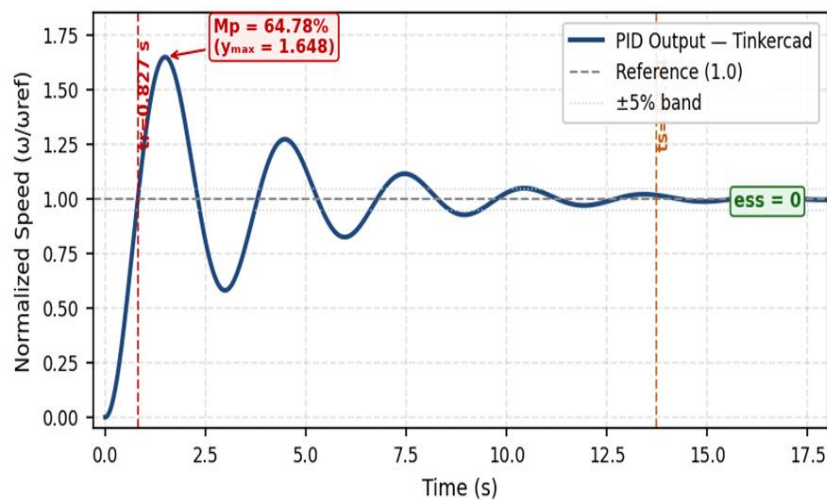


Figure 3. Step response of the DC motor controlled by the proposed discrete-time PID controller in the Tinkercad microcontroller simulation environment ($K_p = 0.9$, $K_i = 0.2$, $K_d = 0.8$, $T = 0.01$ s)

The response is markedly underdamped, characterized by a peak overshoot $M_p = 64.78\%$ and sustained oscillations that decay gradually until steady state is reached at $t_s = 13.74$ s. The closed-loop damping ratio, estimated from the measured overshoot using $\zeta = -\ln(M_p/100)/\sqrt{\pi^2 + \ln^2(M_p/100)}$, yields $\zeta = 0.137$, confirming operation well into the underdamped regime.

Despite the oscillatory transient, no measurable steady-state error is observed within the numerical resolution of the simulation ($ess \approx 0$), consistent with the discrete integral action eliminating accumulated tracking error even under finite-precision arithmetic. This observation should be interpreted with some caution: it reflects the behavior of the Tinkercad simulation model rather than a formal proof that steady-state error is exactly zero under all finite-precision conditions, and a small residual offset below the simulation's measurement resolution cannot be ruled out. With that caveat, the result nonetheless suggests a practical implication: the

integral channel of the discrete PID appears comparatively resilient to implementation constraints, tending to preserve elimination of steady-state error across the word-length limitations captured by this simulation study. Whether this resilience holds on physical hardware, where additional non-idealities such as encoder quantization, static friction, and back-EMF variation are present, has not been verified here and remains to be confirmed experimentally.

4.2. Step Response: Tinkercad Microcontroller Environment

The step responses of both environments, obtained under identical controller parameters, are overlaid in Fig. 4 for direct comparison. The Simulink simulation produces a critically damped response without overshoot ($M_p = 0\%$), with a rise time $t_r = 10.0$ s and settling time $t_s = 12.0$ s. The estimated damping ratio $\zeta > 1$ confirms overdamped transient behavior under ideal conditions.

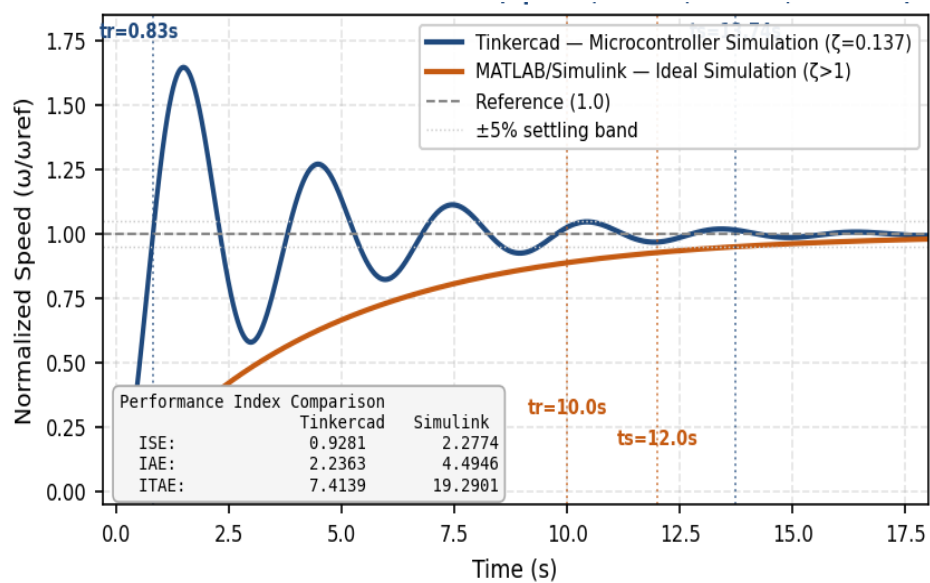


Figure 4. Overlay of step responses: Tinkercad microcontroller simulation vs. MATLAB/Simulink ideal simulation, using identical controller parameters

4.3. Comprehensive Performance Comparison

Table 3 presents the complete time-domain performance metrics and integral performance indices for both environments. ISE, IAE, and ITAE values are computed

numerically by integrating the squared or absolute error signal over a 20-second evaluation window from the simulated step response data.

Table 3. Comprehensive Performance Comparison of The Proposed Discrete-Time PID Controller Across Both Simulation Environments

Performance Metric	Tinkercad	MATLAB / Simulink
Rise Time, t_r (s)	0.827	10.0
Settling Time, t_s (s)	13.74	12.0
Peak Overshoot, M_p (%)	64.78	0
Delay Time, t_d (s)	0.41	4.55
Steady-State Error, e_{ss}	0	0
Damping Ratio, ζ	0.137 (underdamped)	> 1.0 (overdamped)
ISE	0.9281	2.2774
IAE	2.2363	4.4946
ITAE	7.4139	19.2901

An important and counterintuitive result emerges from the integral index analysis: the Tinkercad implementation achieves lower ISE ($0.9281 < 2.2774$), lower IAE ($2.2363 < 4.4946$), and lower ITAE ($7.4139 < 19.2901$) compared to the Simulink simulation. Although the Tinkercad response exhibits significant overshoot, its dramatically faster initial response means it accumulates substantially

less integrated error over the full 20-second evaluation window than the slowly converging Simulink response.

This finding challenges the implicit assumption that high overshoot necessarily implies inferior overall performance. When the evaluation criterion is total accumulated tracking error over a fixed time horizon as captured by ISE, IAE, and ITAE, a fast-

underdamped response may outperform a slow overdamped one. The appropriate performance criterion therefore depends on the specific application requirements: regulatory processes where transient smoothness is critical will favor the overdamped Simulink response, while time-critical positioning applications may prefer the faster, albeit oscillatory, Tinkercad behavior.

4.4. Root-Cause Analysis of Implementation Induced Discrepancies

The substantial differences in transient behavior between the two environments are systematically attributed to four well-defined implementation factors, summarized in Table 4. This attribution is analytical and qualitative in nature: it is constructed by comparing the

known computational characteristics of each environment (Table 4) against established finite-word-length control theory and against the observed step-response discrepancies, rather than from isolated experiments that vary each factor independently. A controlled ablation study for example, systematically varying arithmetic precision, injected computational delay, and PWM resolution one at a time while holding the others fixed would be required to quantitatively isolate and rank the individual contribution of each factor; this is identified as a direction for future work in the Discussion below.

Table 4. Root-Cause Analysis of Performance Discrepancies: MATLAB/Simulink Vs. Tinkercad Microcontroller Simulation

Implementation Factor	MATLAB / Simulink	Tinkercad (Microcontroller)
Arithmetic precision	64-bit double (negligible error)	8/16-bit fixed (quantization errors)
Sampling jitter	Deterministic (none)	Non-zero (scheduler overhead)
Computational delay	Negligible	≈ 1 sampling period (0.01 s)
PWM output resolution	Continuous equivalent	8-bit (256 discrete levels)
Coefficient quantization	None	Present in $K_d/T = 80$ term
Resulting damping	Critically damped ($\zeta > 1$)	Underdamped ($\zeta = 0.137$)

The dominant factor is arithmetic precision. MATLAB/Simulink employs 64-bit double-precision floating-point computation, ensuring that all PID gain values are applied with negligible quantization error. The Arduino platform in Tinkercad operates with 8/16-bit integer arithmetic, which introduces coefficient quantization in the derivative term. Specifically, the discrete derivative coefficient $K_d/T = 0.8/0.01 = 80$ is a large integer that, when multiplied by the error difference ($e[k] - e[k-1]$), amplifies any quantization noise present in the error signal. This effectively reduces the net damping contribution of the derivative channel and is analytically identified as a principal contributor to the underdamped behavior observed in Fig. 3, alongside the computational delay and PWM quantization effects discussed below.

A secondary factor is computational delay. At a sampling period of $T = 0.01$ s, even a few milliseconds of execution overhead per control cycle constitute a meaningful phase

delay. This delay effectively shifts the closed-loop phase margin leftward, reducing stability margins beyond those predicted by the ideal Z-domain analysis and increasing the tendency toward oscillation [30].

The 8-bit PWM resolution (256 discrete voltage levels) introduces a third factor: a quantization deadband that prevents the application of arbitrarily small corrective voltages. This quantization of the control effort contributes to the residual oscillation observed during the settling phase of the Tinkercad response, as the controller cannot smoothly fine-tune the applied voltage near the reference point.

These findings are consistent with established theory on finite-word-length effects in digital control systems [12], [30], and demonstrate that implementation-aware analysis is indispensable for accurate prediction of real-world embedded controller behavior.

4.5. Discussion: Design Implications

The results yield three actionable design insights for engineers implementing PID controllers on microcontroller-based platforms. First, the integral channel is robust to finite-precision constraints and can be reliably deployed even in resource-constrained embedded systems for steady-state error elimination. Second, the derivative channel is the most vulnerable PID component to word-length effects: when K_d/T is large, coefficient quantization significantly degrades damping. Practitioners should consider (i) reducing K_d to limit the magnitude of the discrete derivative coefficient, or (ii) incorporating a first-order derivative filter to attenuate quantization noise amplification [12]. Third, performance assessment should always include implementation constrained simulations alongside ideal benchmarks, and the choice of performance index (ISE vs. IAE vs. ITAE) must reflect the specific application priority (fast convergence vs. smooth transient vs. early-error emphasis).

A key limitation of the present work is the exclusive use of simulation environments. While Tinkercad faithfully replicates several computational constraints of the Arduino Uno, it remains a simulation model, and physical hardware introduces additional nonlinearities including motor friction, thermal drift, encoder quantization, and back-EMF variations that are not captured in the simulation model. Consequently, all claims regarding implementation-induced performance degradation, root-cause attribution, and the resilience of steady-state accuracy in this study should be understood as applying specifically to the Tinkercad simulation environment, and should not be extrapolated to physical embedded hardware without direct experimental confirmation. Hardware-in-the-loop (HIL) validation on a physical Arduino-controlled DC motor testbed is planned as the primary direction for future work to verify whether the trends identified here persist under real hardware conditions.

5. CONCLUSION

This paper has presented the design and cross-environment performance analysis of a discrete-time PID controller for DC motor

speed regulation using the Z-transform method. The following conclusions are established:

- a. The proposed discrete-time PID controller shows no measurable steady-state error in either the Tinkercad microcontroller simulation or the MATLAB/Simulink ideal simulation, an observation consistent with the integral control action remaining effective under the finite-precision arithmetic constraints represented in these simulation environments; confirmation on physical hardware is left to future work.
- b. Formal closed-loop stability is verified through the Z-plane characteristic equation of the complete feedback system, $1 + C(z)G(z) = 0$, rather than from plant poles alone. All three closed-loop poles are located at $|z| < 1$ ($z = 0.9975, 0.9903, 0.0967$), satisfying the necessary and sufficient condition for discrete-time asymptotic stability; the two poles closest to the unit circle explain the slow, lightly-damped settling behavior observed in both simulation environments.
- c. Significant cross environment performance discrepancies are observed: the Tinkercad implementation yields an underdamped response ($M_p = 64.78\%$, $t_r = 0.827\text{ s}$, $t_s = 13.74\text{ s}$, $\zeta = 0.137$), while the Simulink simulation produces a critically damped response ($M_p = 0\%$, $t_r = 10.0\text{ s}$, $t_s = 12.0\text{ s}$, $\zeta > 1$).
- d. Root-cause analysis analytically attributes the performance gap to four implementation factors, consistent with established finite-word-length control theory: (i) finite-precision arithmetic reducing derivative damping, (ii) computational delay reducing effective phase margin, (iii) 8-bit PWM resolution introducing control signal quantization, and (iv) coefficient quantization in the discrete derivative term ($K_d/T = 80$).
- e. Integral performance index analysis (ISE, IAE, ITAE) reveals that the Tinkercad implementation despite its higher overshoot accumulates less total error over a 20-second horizon than the Simulink simulation, demonstrating that performance rankings between environments are criterion-dependent.

These findings collectively establish that ideal simulation-based evaluation is insufficient as the sole validation tool for microcontroller-based control systems. Implementation-aware analysis, incorporating finite-word-length effects, computational delay, and actuator quantization as identified through the root-cause framework developed in Section IV-D is essential for reliable embedded control system development. Future research will pursue hardware-in-the-loop validation on a physical DC motor testbed, a controlled ablation study that independently varies arithmetic precision, computational delay, and PWM resolution to quantitatively isolate their individual contributions, systematic PID re-tuning using metaheuristic optimization methods adapted to the discrete implementation environment, and investigation of derivative filtering strategies to mitigate finite-precision damping degradation.

REFERENCES

- [1] Muhammad Hilal Mthboob, H. ALRikabi, and I. A. Aljazaery, "A concepts and techniques related to the DC motor speed control system design: Systematic Review," *Wasit Journal of Computer and Mathematics Science*, vol. 2, no. 1, pp. 59–73, Mar. 2023, doi: 10.31185/wjcm.121.
- [2] M. Cheng, J. Zhou, W. Qian, B. Wang, C. Zhao, and P. Han, "Advanced Electrical Motors and Control Strategies for High-quality Servo Systems - A Comprehensive Review," *Chinese Journal of Electrical Engineering*, vol. 10, no. 1, pp. 63–85, Mar. 2024, doi: 10.23919/CJEE.2023.000048.
- [3] M. Carminati and G. Scandurra, "Impact and trends in embedding field programmable gate arrays and microcontrollers in scientific instrumentation," *Review of Scientific Instruments*, vol. 92, no. 9, Sep. 2021, doi: 10.1063/5.0050999.
- [4] M. R. Idris, M. N. Irdyanti, and M. Ramlee, "A review of microcontroller design in future industrial revolution," in *AIP Conf. Proc.*, 2023, p. 050031. doi: 10.1063/5.0149680.
- [5] Z. Li, "Review of PID control design and tuning methods," *J. Phys. Conf. Ser.*, vol. 2649, no. 1, p. 012009, Nov. 2023, doi: 10.1088/1742-6596/2649/1/012009.
- [6] J. P. Carvalho Henriques, E. Henrique de Paiva Silva, J. H. Silva Delfino, E. Raimundo Neto, J. P. Magalhães de Paula Paiva, and A. Baratella Lugli, "Embedded Velocity Control System for Very Small Size Soccer Robot," in *2025 13th International Conference on Control, Mechatronics and Automation (ICCMA)*, IEEE, Nov. 2025, pp. 6–11. doi: 10.1109/ICCMA67641.2025.11369570.
- [7] P. Shen, "Advancements in integrated circuit encoders: design, optimization, and applications in high-precision control systems," *IET Conference Proceedings*, vol. 2024, no. 24, pp. 525–530, Jan. 2025, doi: 10.1049/icp.2024.4576.
- [8] Y. Suzantry Handayni, "Otomatisasi Pengontrol Temperatur Suhu Untuk Mesin Pengereng Pelet Menggunakan Kontroler Autonics TK4S Dengan Metode PID," *Electrician : Jurnal Rekayasa dan Teknologi Elektro*, vol. 17, no. 1, pp. 20–25, Jan. 2023, doi: 10.23960/elc.v17n1.2396.
- [9] T. George and V. Ganesan, "Optimal tuning of PID controller in time delay system: a review on various optimization techniques," *Chemical Product and Process Modeling*, vol. 17, no. 1, pp. 1–28, Feb. 2022, doi: 10.1515/cppm-2020-2001.
- [10] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems," *Heliyon*, vol. 8, no. 5, p. e09399, May 2022, doi: 10.1016/j.heliyon.2022.e09399.
- [11] R. Shahouni, M. Bahraini, M. Abrofarakh, and M. Abbasi, "Adaptive tuning of fractional order PID controllers for nonlinear processes using hybrid PSO DQN reinforcement learning," *Sci. Rep.*, vol. 15, no. 1, p. 38545, Nov. 2025, doi: 10.1038/s41598-025-22509-x.
- [12] M. Huba, S. Chamraz, P. Bistak, and D. Vrancic, "Making the PI and PID Controller Tuning Inspired by Ziegler and Nichols Precise and Reliable," *Sensors*, vol. 21, no. 18, p. 6157, Sep. 2021, doi: 10.3390/s21186157.
- [13] G. Yükses, "A novel delay-affected two degree of freedom PID controller using physics-inspired optimization for robust control applications," *Sci. Rep.*, vol. 16, no. 1, p. 3453, Dec. 2025, doi: 10.1038/s41598-025-33481-x.
- [14] J.-H. Urrea-Quintero, J. N. Fuhg, M. Marino, and A. Fau, "PI/PID controller stabilizing sets of uncertain nonlinear systems: an efficient surrogate model-based approach," *Nonlinear Dyn.*, vol. 105, no. 1, pp. 277–299, Jul. 2021, doi: 10.1007/s11071-021-06431-1.

- [15] E. Ryzty, "Pengembangan Mini Drone Berbasis ESP32 dengan Roll dan Pitch Control Menggunakan Algoritma PID," *Electrician : Jurnal Rekayasa dan Teknologi Elektro*, vol. 18, no. 2, pp. 171–182, May 2024, doi: 10.23960/elc.v18n2.2597.
- [16] I. S. Leal, C. Abeykoon, and Y. S. Perera, "Design, Simulation, Analysis and Optimization of PID and Fuzzy Based Control Systems for a Quadcopter," *Electronics (Basel)*, vol. 10, no. 18, p. 2218, Sep. 2021, doi: 10.3390/electronics10182218.
- [17] A. Horvath, P. Beda, and D. Takacs, "Modelling and stability analysis of a longitudinal wheel dynamics control loop with feedback delay," *Vehicle System Dynamics*, vol. 60, no. 11, pp. 3765–3787, Nov. 2022, doi: 10.1080/00423114.2021.1975775.
- [18] B. Pokharel, "A comprehensive study of modern control strategies: From classical PID to optimal, robust, adaptive and intelligent methods," *International Research Journal of Modernization in Engineering Technology & Science*, Sep. 2025, doi: 10.56726/IRJMETS83160.
- [19] V. B. S. Cassano, E. S. Vitor Junior, F. K. Kaida, W. P. N. dos Reis, and O. Morandin Junior, "Simulating Automated Guided Vehicles in Unity: A Case Study on PID Controller Tuning," *Applied System Innovation*, vol. 8, no. 6, p. 170, Nov. 2025, doi: 10.3390/asi8060170.
- [20] G. Wainer and S. Govind, "100 volumes of SIMULATION—20 years of DEVS research," *Simulation*, vol. 100, no. 12, pp. 1297–1318, Dec. 2024, doi: 10.1177/00375497241291871.
- [21] U. W. Pooch and J. A. Wall, *Discrete Event Simulation*. Boca Raton: CRC Press, 2024. doi: 10.1201/9781003575054.
- [22] S. S. Saha, S. S. Sandha, and M. Srivastava, "Machine Learning for Microcontroller-Class Hardware: A Review," *IEEE Sens. J.*, vol. 22, no. 22, pp. 21362–21390, Nov. 2022, doi: 10.1109/JSEN.2022.3210773.
- [23] M. Anees, N. B.G., T. Y. G, and C. M. Patil, "Experimental Based Learning: Microcontroller Simulation Using Virtual Simulation Machine," *Journal of Engineering Education Transformations*, vol. 38, no. 3, pp. 199–209, Jan. 2025, doi: 10.16920/jeet/2024/v38i3/24255.
- [24] M. Kuczmann, "Review of DC Motor Modeling and Linear Control: Theory with Laboratory Tests," *Electronics (Basel)*, vol. 13, no. 11, p. 2225, Jun. 2024, doi: 10.3390/electronics13112225.
- [25] Rukmini, M. Rathi, and M. S. Aspalii, "Speed Control of DC Motor using PWM Technique," in *2024 5th IEEE Global Conference for Advancement in Technology (GCAT)*, IEEE, Oct. 2024, pp. 1–6. doi: 10.1109/GCAT62922.2024.10923985.
- [26] Katsuhiko. Ogata, *Modern control engineering*. Prentice Hall, 2022.
- [27] B.-H. Nguyen *et al.*, "Application of Genetic Algorithm for Optimizing Continuous and Discrete PID to Control Antenna Azimuth Position," *Journal of Fuzzy Systems and Control*, vol. 2, no. 1, pp. 1–5, Jan. 2024, doi: 10.59247/jfsc.v2i1.154.
- [28] G. Görel, Y. Bekteş, F. Korkmaz, and H. Özkaya, "PID Controller Design in Buck Converters: Real-Time Implementation and Performance Evaluation with TMS320F28379D," *Alfa Mühendislik ve Uygulamalı Bilimler Dergisi*, vol. 3, no. 1, pp. 1–14, Apr. 2025, doi: 10.70988/ajeas.1583799.
- [29] A. H. Wirasapta, T. D. Pamungkas, and S. S. Mishi, "Simulation-Based Analysis of Packet Scheduling Strategies and Traffic Load Effects on Network QoS," *Energy Insights*, vol. 1, no. 2, pp. 49–58, Apr. 2026, doi: 10.59562/ei.v1i2.11311.
- [30] C. Hobbs, B. Ghosh, S. Xu, P. S. Duggirala, and S. Chakraborty, "Safety Analysis of Embedded Controllers Under Implementation Platform Timing Uncertainties," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4016–4027, Nov. 2022, doi: 10.1109/TCAD.2022.3198905.
- [31] D. Tessal *et al.*, "Designing a Google Voice-Based Solar Panel Cleaner Control System," *Jurnal Edukasi Elektro*, vol. 9, no. 2, pp. 202–213, Oct. 2025, doi: 10.21831/jee.v9i2.90041.