

Article

Design and Performance Analysis of an Adaptive PID Controller for Brushless DC Motor Systems in Electric Vehicles

Md Mahmud ^{1,2,*} , S. M. Rakibul Islam ^{1,2}  and S. M. A. Motakabber ³ ¹ Phillip M. Drayer Department of Electrical and Computer Engineering, College of Engineering, Lamar University, Beaumont, TX 77710, USA; irakibul281@gmail.com² Resa Power, 50613 Varsity Ct, Wixom, MI 48393, USA³ Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia; amotakabber@iiu.edu.my

* Correspondence: mmahmud12@lamar.edu

Abstract

Brushless DC (BLDC) motors are now the dominant propulsion choice for electric vehicles (EVs) because of their high torque density, efficiency and reliability, but their nonlinear dynamics, electronic commutation, and wide load and speed range make fixed-gain control difficult. A single set of proportional–integral–derivative (PID) gains tuned at one operating point degrades when inertia, back-EMF, or load torque change. This paper presents a hybrid adaptive PID speed controller for a BLDC EV drive that couples an online PID auto-tuner that re-estimates the gains from a frequency response estimate of the plant, with a fast fixed-structure PID that supplies the rapid corrective action that the auto-tuner cannot provide during its estimation interval. The novelty of this work is this explicit two-element decomposition operating on a cascaded speed/voltage loop driven by Hall sensor feedback, which removes the need for an exact analytical feedback model while retaining the transparency of classical PID. A full analytical model of the BLDC machine and the closed-loop transfer functions is derived and implemented in MATLAB/Simulink. Across step references of 1000–1800 rpm and load steps to 10 N·m, and against a conventional fixed-gain PID and a Flower Pollination Algorithm (FPA)-tuned PID, the proposed controller holds overshoot below 1% at low-to-mid speed and a consistently lower torque ripple, while a 12.4% transient undershoot at 1800 rpm under sudden load identifies the present operating limit and a direction for future work.

Keywords: PID controller; electric vehicles; adaptive PID; BLDC; PWM; fuzzy logic

Academic Editor: Grzegorz Sierpiński

Received: 4 February 2026

Revised: 9 June 2026

Accepted: 21 June 2026

Published: 12 August 2026

Copyright: © 2026 by the authors.

Published by MDPI on behalf of the World Electric Vehicle Association.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Electric vehicles (EVs) have moved from a niche technology to a central element of road transport decarbonisation, and the electric machine at the heart of the drivetrain largely determines the vehicle's efficiency, range, and drive quality. Among the available machines, the brushless DC (BLDC) motor is widely used in EV traction and auxiliary drives because it combines the high torque-to-weight ratio and controllability of DC machines with the durability of an electronically commutated, brushless construction. By replacing mechanical brushes with solid-state commutation based on rotor position feedback, the BLDC motor offers high efficiency, a favourable torque–speed characteristic, and low maintenance, which is why it has become a standard choice for EV and industrial drive systems. Figure 1 shows Schematic of an electric vehicle (EV) powertrain showing the

location of the BLDC motor controller with adaptive PID, the battery pack, and the power electronic stages [1–3].

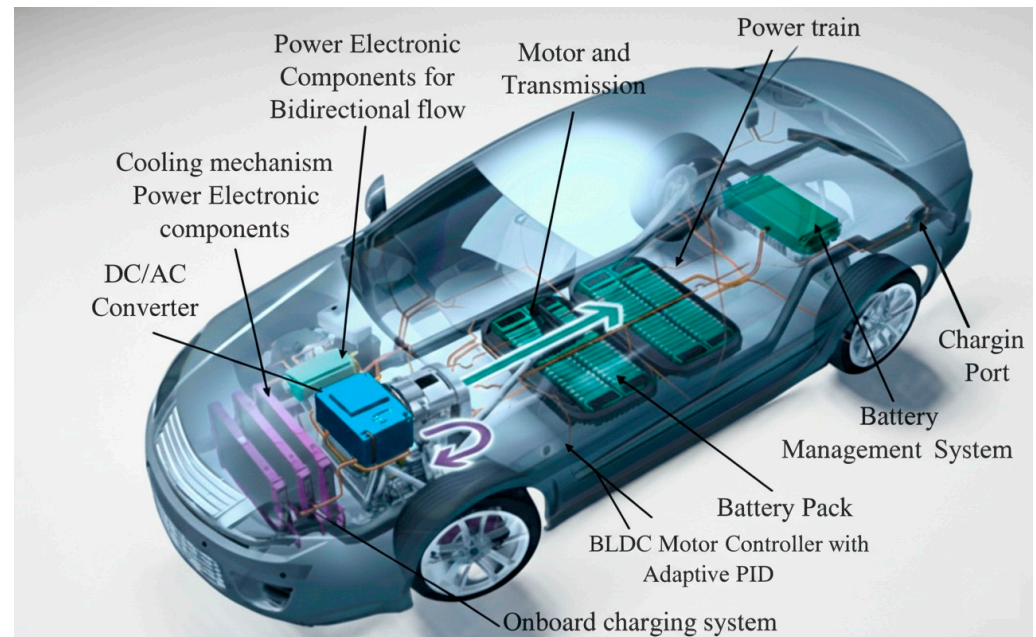


Figure 1. Schematic of an electric vehicle (EV) powertrain showing the location of the BLDC motor controller with adaptive PID, the battery pack, and the power electronic stages.

Controlling a BLDC drive in a vehicle is, however, demanding. The machine is nonlinear; its parameters (winding resistance, inductance, effective inertia, and load torque) vary with temperature, state of charge, road grade, and payload; and the drive must remain stable and well damped across a wide speed range. The proportional–integral–derivative (PID) controller remains the workhorse of motion control because it is simple, transparent, and easy to implement, but a single fixed set of gains tuned at one operating point cannot remain optimal as the plant changes. Under real driving conditions, finding and maintaining a good fixed tuning is therefore difficult, and the result is either sluggish response or excessive overshoot and torque ripple [4–7].

To overcome the limitations of fixed-gain control, a substantial body of work has developed adaptive and self-tuning PID schemes, several of which are also described in the literature as gain-scheduled PID control [8–11]. In gain scheduling, a family of controllers is designed offline for several operating conditions and the active gains are interpolated according to a measurable scheduling variable; this approach has been applied successfully to nonlinear plants such as fixed-wing and rotary UAVs and power converters, and is supported by a mature analysis and design theory [12,13]. Vitelli [12] derived small-signal models of DC–DC converters and showed that a gain-scheduling strategy whose PI gains are adapted to the operating point clearly outperforms a single controller tuned only at one point. Saeki [13] characterised the set of stabilising PID gains in parameter space, providing the theoretical basis for safely scheduling or adapting PID parameters. Closely related families include fuzzy PID and adaptive fuzzy schemes—fuzzy PID being, in effect, a form of adaptive PID in which the gains are adjusted by a rule base—which have been used for EV speed control and drivetrain coordination [4,8,14–20], metaheuristically tuned PID, such as the Flower Pollination Algorithm (FPA) and particle-swarm variants [3,5], and robust and reinforcement-learning-based tuning [7,8].

Despite this maturity, two practical gaps remain for EV BLDC drives. First, gain scheduling and many adaptive schemes depend on either a measurable scheduling variable

or an accurate analytical plant model; obtaining and maintaining such a model across the full automotive operating envelope is difficult. Second, model-free metaheuristic tuners (FPA, PSO) and online estimators typically need a finite estimation or convergence interval during which the closed loop is more weakly controlled, which can show up as overshoot or ripple at the very transients that matter in a drivetrain.

This paper addresses both gaps with a hybrid adaptive PID controller that explicitly separates the slow adaptation from the fast regulation. An online PID auto-tuner re-estimates the PID gains from a frequency response estimate of the plant—so no exact analytical feedback model is required while a fast fixed-structure PID runs in parallel and supplies the immediate corrective action that bridges the auto-tuner's estimation interval. The two elements act as mutual correction layers on a cascaded speed/voltage loop driven directly by Hall sensor feedback. Unlike a pure scheduling table, the scheme does not require a pre-built family of controllers or a measured scheduling variable; unlike a pure metaheuristic tuner, it does not leave the transient weakly controlled while the optimiser converges.

The main contributions of this work are summarised as follows:

- A hybrid adaptive PID architecture for BLDC EV drives that combines a frequency-response-based PID auto-tuner with a parallel fast PID, removing the need for an explicit analytical feedback model while keeping the interpretability of classical PID.
- A complete analytical model of the BLDC machine—phase voltage equations, electromagnetic torque, mechanical dynamics, and the resulting speed-to-voltage transfer function—together with the closed-loop transfer functions and the adaptation law, given at a level of detail sufficient to reproduce the design.
- A cascaded speed/voltage control structure, presented as a control-system block diagram, in which the outer loop regulates speed and the inner loop regulates the DC-link/PWM command, with the auto-tuner updating both PID blocks online.
- A comparative MATLAB/Simulink evaluation against a conventional fixed-gain PID and an FPA-tuned PID over 1000–1800 rpm and load steps to 10 N·m, including the identification of the high-speed operating limit (a 12.4% undershoot at 1800 rpm) and the corresponding direction for future work.

The remainder of this paper is organised as follows. Section 2 presents the system model and the proposed control method, including the BLDC analytical model, the PID transfer functions, and the adaptive tuning algorithm. Section 3 describes the simulation platform and reports the comparative results. Section 4 concludes the paper and outlines future work, including experimental validation.

2. System Model and Proposed Method

This section presents the analytical model of the BLDC machine and its drive, derives the closed-loop transfer functions, defines the PID transfer functions that were adapted, and specifies the adaptive tuning algorithm. The control architecture is shown as a block diagram in Figure 2; the corresponding MATLAB/Simulink realisation, used for all results in Section 3, is shown in Figure 3.

2.1. Control Architecture

The proposed drive uses a cascaded (nested) control structure, as shown in Figure 2. The outer loop regulates rotor speed: the speed error between the reference ω^* and the Hall sensor speed estimate ω drives the adaptive speed PID, $C_\omega(s)$, the output of which is a voltage command V^* . The inner loop regulates the DC-link/PWM command: the voltage error between V^* and the measured DC-link voltage drives the adaptive voltage PID, $C_V(s)$, the output u of which is converted by the PWM generator into gate signals for the buck

converter and three-phase inverter. The inverter feeds the BLDC machine $G(s)$, and the Hall effect and current/voltage sensors close both loops. A PID auto-tuner runs alongside the controllers: it forms a frequency response estimate of the plant from the measured signals and updates the gains of both PID blocks online (Figure 2). This decomposition lets the fast inner PID reject voltage and commutation disturbances quickly while the auto-tuner slowly re-optimises the gains as the operating point drifts.

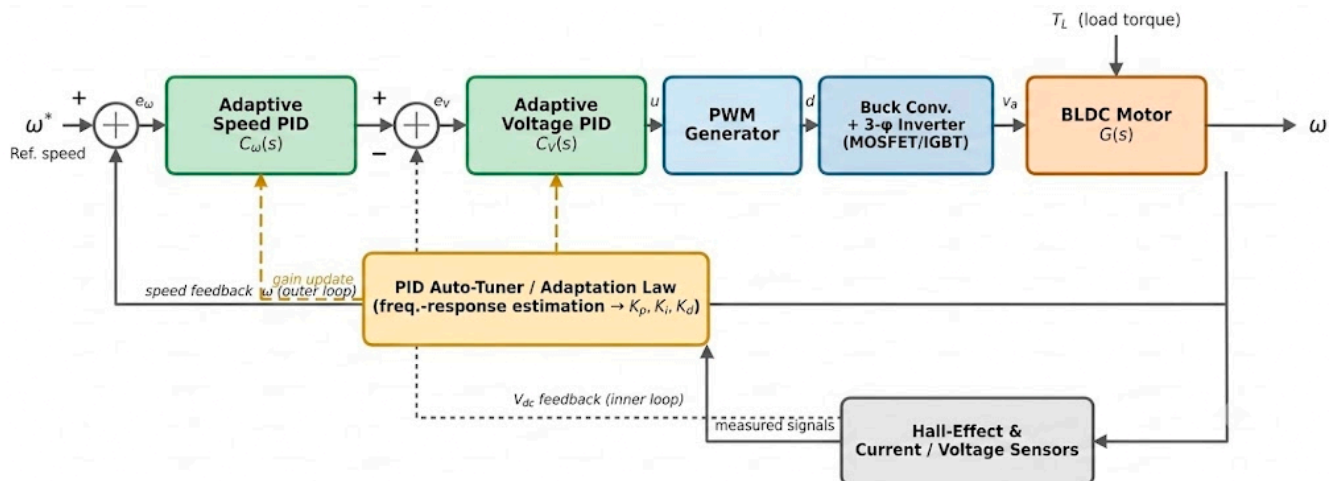


Figure 2. Block diagram of the proposed adaptive PID control architecture for the BLDC EV drive. The outer loop regulates speed, and the inner loop regulates the DC-link/PWM command; the PID auto-tuner updates the gains of both PID blocks online from the Hall effect and current/voltage sensor measurements.

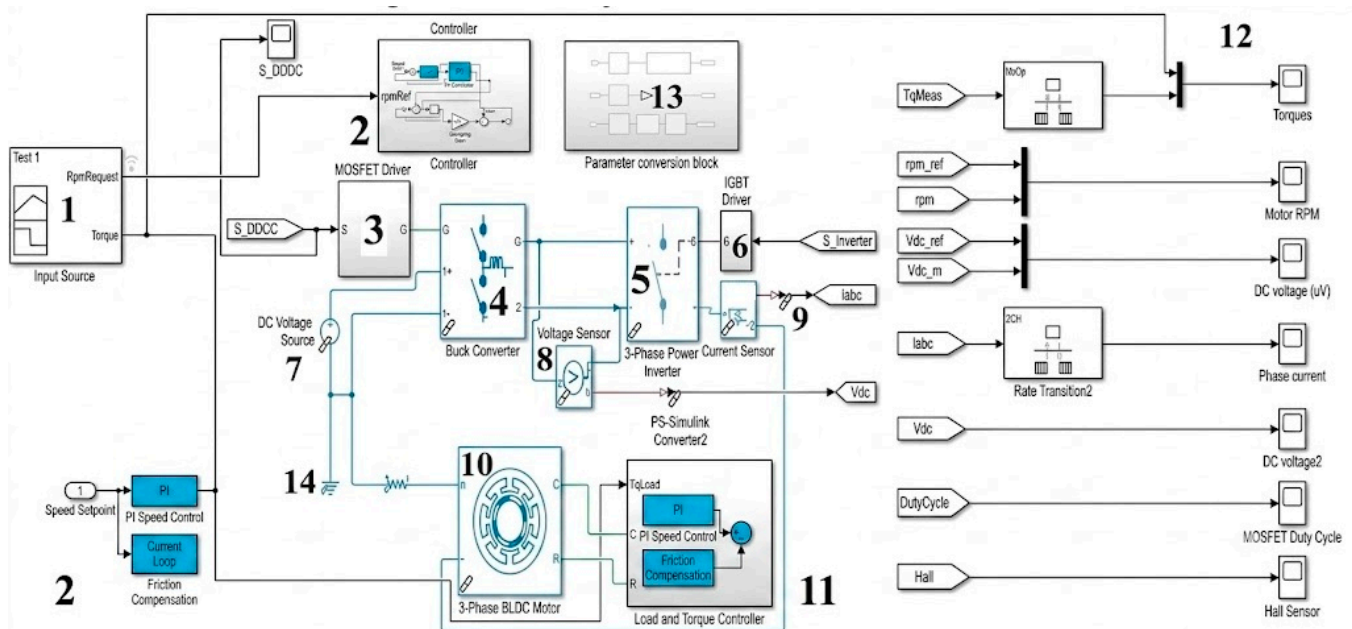


Figure 3. Dynamic simulation framework for the controller: (1) Input Source/Test Signal Request; (2) Controller Block; (3) MOSFET Driver; (4) Buck Converter; (5) 3-Phase Power Inverter; (6) IGBT Driver; (7) DC Voltage Source; (8) Voltage Sensor; (9) Current Sensor; (10) 3-Phase BLDC Motor; (11) Mechanical Load Subsystem; (12) Scope/Signal Measurement Outputs; (13) Parameter Conversion Block; (14) Electrical Reference Ground.

2.2. BLDC Motor Model

The BLDC machine is a three-phase, permanent-magnet synchronous machine with a trapezoidal back-EMF. With the usual assumptions of a symmetric, non-saturated machine, negligible mutual inductance variation, and equal phase resistances and inductances, the back-EMF of the three phases is proportional to speed and to the rotor position shape functions f_a , f_b , and f_c :

$$e_a = K_e \cdot \omega \cdot f_a(\theta_e), e_b = K_e \cdot \omega \cdot f_b(\theta_e), e_c = K_e \cdot \omega \cdot f_c(\theta_e) \quad (1)$$

where K_e is the back-EMF constant, ω is the rotor mechanical speed, and θ_e is the electrical angle. Each stator phase obeys a voltage balance equation with phase resistance R_s , phase inductance L , and the corresponding back-EMF e_x :

$$v_x = R_s \cdot i_x + L \cdot \frac{di_x}{dt} + e_x, x \in \{a, b, c\} \quad (2)$$

During two-phase conduction, the electromagnetic torque produced by the machine is the sum of the three electrical powers divided by speed, which for a trapezoidal machine reduces to a torque constant K_t multiplied by the conducting current i :

$$T_e = \frac{e_a i_a + e_b i_b + e_c i_c}{\omega} = K_t \cdot i \quad (3)$$

The mechanical dynamics relate this electromagnetic torque T_e to the load torque T_L , the combined rotor-and-load inertia J , and the viscous friction coefficient B :

$$J \cdot \frac{d\omega}{dt} + B \cdot \omega = T_e - T_L \quad (4)$$

Lumping the conducting pair into an equivalent single-phase representation, the electrical balance seen by the controller is

$$V = R_s \cdot i + L \cdot \frac{di}{dt} + K_e \cdot \omega \quad (5)$$

where V is the effective armature voltage applied by the inverter. Equations (3)–(5) are nonlinear because of the commutation-dependent shape functions and the speed-dependent back-EMF; for the controller design, they are linearised about an operating point. Taking Laplace transforms of (4) and (5) at zero initial conditions and eliminating the current gives the speed-to-voltage transfer function of the machine, which is justified as a small-signal model valid about each operating point:

$$G(s) = \frac{\omega(s)}{V(s)} = \frac{K_t}{(Js + B)(Ls + R_s) + K_t K_e} \quad (6)$$

This second-order model $G(s)$ is the plant for both the analytical design and the auto-tuner's target. Because J , B , R_s , and the effective K_e/K_t vary with temperature, load, and state of charge, the coefficients of $G(s)$ drift during operation, which is precisely the variation the adaptive scheme is intended to track. Representative parameters used in the simulation are listed in Table 1.

Table 1. Benchmark BLDC motor and drive parameters used in the simulation.

Parameter	Symbol	Value
DC-link/source voltage	V	48 V
Phase resistance	R _s	0.5 Ω
Phase inductance	L	1.5 mH
Back-EMF constant	K _e	0.08 V·s/rad
Torque constant	K _t	0.08 N·m/A
Rotor + load inertia	J	1.0 × 10 ^{−3} kg·m ²
Viscous friction	B	1.0 × 10 ^{−3} N·m·s
Number of pole pairs	p	4
Rated load torque	T _L	10 N·m
PWM switching frequency	f _{sw}	20 kHz

2.3. PID Transfer Functions to Be Adapted

Each controller block in Figure 2 is a PID with a filtered derivative term. In continuous time, the control law is

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de(t)}{dt} \quad (7)$$

and the corresponding transfer function, with the derivative filtered by a first-order term of bandwidth N to avoid noise amplification, is

$$C(s) = K_p + \frac{K_i}{s} + \frac{K_d \cdot N \cdot s}{s + N} \quad (8)$$

The speed controller $C\omega(s)$ and the voltage controller $CV(s)$ share this structure but have independent gain sets $\{K_p, K_i, K_d\}$; these are the parameters the auto-tuner adapts. With the plant $G(s)$ from (6), the closed-loop transfer function of each loop is

$$T(s) = \frac{C(s) \cdot G(s)}{1 + C(s) \cdot G(s)} \quad (9)$$

The gains are what the adaptation law adjusts so that $T(s)$ retains the desired phase margin and crossover frequency as $G(s)$ drifts. Initial gains were obtained analytically from (6) by pole placement/phase-margin design at the nominal operating point and then handed to the auto-tuner as a starting condition.

2.4. Adaptive Tuning Algorithm

The adaptive layer combines a slow, model-free auto-tuner with a fast fixed PID. The auto-tuner periodically estimates the plant frequency response and recomputes the PID gains to meet a target phase margin and crossover frequency; the fast PID provides regulation between updates. The procedure runs continuously and consists of the following steps.

Step 1—Excitation and frequency response estimation. A small probing perturbation (or the natural command and load activity) is applied, and the input u and output y are recorded. The plant frequency response is estimated at a set of probing frequencies ω_k from the ratio of the measured output and input spectra:

$$\hat{G}(j\omega_k) = \frac{Y(j\omega_k)}{U(j\omega_k)}, \quad k = 1, \dots, m \quad (10)$$

This non-parametric estimate $\hat{G}(j\omega_k)$ captures the current dynamics of $G(s)$ without requiring its analytical coefficients, which is what makes the scheme model-free with respect to the feedback path.

Step 2—Gain computation for a target margin. From the estimated response, the gains $\{K_p, K_i, K_d\}$ of the PID in (8) are computed so that the loop gain has unity magnitude at the desired crossover frequency ω_c and the required phase margin φ_m :

$$|C(j\omega_c) \cdot G(j\omega_c)| = 1, \angle[C(j\omega_c) \cdot G(j\omega_c)] = -180^\circ + \varphi_m \quad (11)$$

Solving (11) for the three gains (with the integral and derivative actions distributed to set the low-frequency tracking and the phase lead) yields the auto-tuned gain set $\theta_{tune} = [K_p, K_i, K_d]$. This is the standard frequency response auto-tuning principle, here evaluated online rather than once at commissioning.

Step 3—Blending the slow and fast controllers. Because the estimation in Steps 1–2 takes a finite interval, a fast fixed-gain PID θ_{fast} is kept active. The gains actually applied to the loop are a convex blend of the auto-tuned and fast gain sets, governed by a confidence factor $\beta(t) \in [0, 1]$ that rises as the frequency response estimate converges:

$$\theta_{applied}(t) = \beta(t) \cdot \theta_{tune} + [1 - \beta(t)] \cdot \theta_{fast}, \theta = [k_p, k_i, k_d]T \quad (12)$$

When the estimate is fresh and uncertain ($\beta \rightarrow 0$), the fast PID dominates and guarantees prompt disturbance rejection; as the estimate converges ($\beta \rightarrow 1$), the auto-tuned gains take over and optimise steady-state accuracy and margin. This blending is the mechanism by which the fast PID compensates for the slow auto-tuning phase, and it is applied independently to the speed loop $C\omega(s)$ and the voltage loop $CV(s)$. Table 2. Hybrid adaptive PID tuning.

Table 2. Hybrid adaptive PID tuning (per control loop).

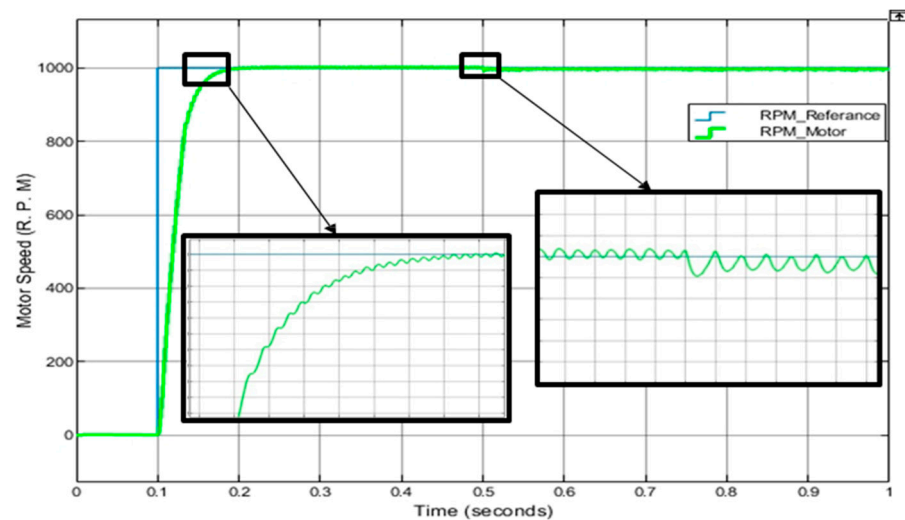
Input: measured $u(t)$, $y(t)$; target margin φ_m , crossover ω_c ; fast gains θ_{fast}	
Output: applied PID gains $\theta(t) = [K_p, K_i, K_d]$	
1:	initialise $\theta \leftarrow \theta_{fast}$, $\beta \leftarrow 0$
2:	repeat every adaptation period:
3:	collect u , y ; estimate $\hat{G}(j\omega_k) = Y(j\omega_k)/U(j\omega_k)$ // Equation (10)
4:	compute θ_{tune} so $ C \cdot \hat{G} = 1$ at ω_c with margin φ_m // Equation (11)
5:	update confidence β from estimate quality (variance)
6:	$\theta \leftarrow \theta_{tune}(1 - \beta) + \theta_{fast} \cdot \beta$ // Equation (12)
7:	apply θ to $C(s)$ // Equation (8)
8:	until stopped

2.5. Simulation Platform

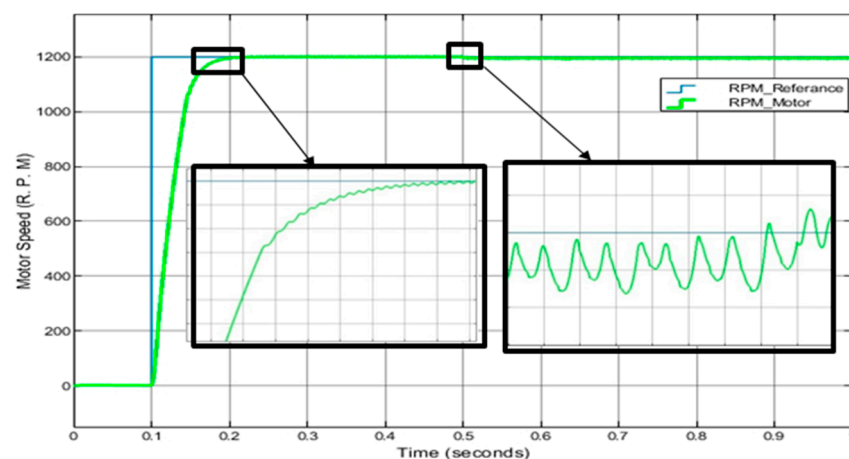
The architecture of Figure 2 was implemented in MATLAB/Simulink-2020(b) as the high-fidelity model shown in Figures 3 and 4, which serves as a digital twin of the physical drive. A 48 V source feeds a buck converter and a three-phase inverter modulated through MOSFET and IGBT gate drivers. Two adaptive PID blocks—one for speed (rpm) and one for the DC-link voltage—implement the cascaded loops of Figure 2, with rate transition blocks interfacing the discrete controller to the continuous power stage. The Hall effect, current, and voltage sensors close the loops and feed the auto-tuner. The model exposes the rpm reference, the load torque input, and the measured speed, voltage, and duty cycle so that overshoot, undershoot, rise time, and ripple can be measured directly. All numerical results in Section 3 were produced with the parameters in Table 1.

Table 3. Simulated measurements for 1000 rpm, 1200 rpm, 1500 rpm, and 1800 rpm.

Measurements	Time for 1000 rpm	Time for 1200 rpm	Time for 1500 rpm	Time for 1800 rpm
Rising time (no load/with load)	0.03638 s/0.05154 s	0.04343 s/0.0739 s	0.06006 s/0.13549 s	0.08526 s/0.13549 s
Max/Min high	1004.52/998.939 rpm	1203/1196.388 rpm	1502/1488.287 rpm	1801/1550.388 rpm
With load, maximum high	1002.97 rpm	1204 rpm	1497 rpm	1576 rpm
Overshoot	0.452%	0.490%	0.502%	0.505%
With load overshoot	0.397%	0.499%	0.305%	0%
With load undershoot	0.963%	1.993%	1.998%	12.444%

**Figure 5.** Speed response at a 1000 rpm reference with a 10 N·m load step and 48 V source and reference (blue) versus measured motor speed (green).

At 1200 rpm, the controller again tracks the reference with only a marginal increase in rise time, attributable to the higher inertia and voltage demand, and holds overshoot at 0.49% (no load) and 0.499% (loaded). Figure 6 shows the corresponding transient. At 1500 rpm, the settling time extends in proportion to the higher speed, consistent with the greater kinetic energy that the adaptive units must manage, and both overshoot and undershoot rise slightly during the transient while the controller still tracks the reference. Figure 7 highlights the instant the output reaches the reference and the response to the applied load.

**Figure 6.** 1200 rpm with preselected torque of -10 N·m and source voltage of 48 V.

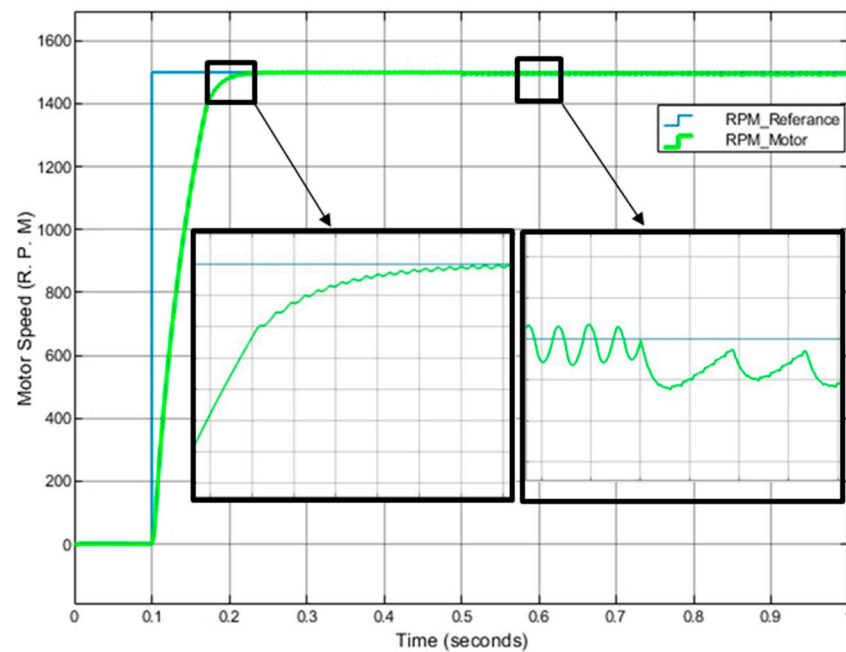


Figure 7. Speed response at a 1500 rpm reference under a 10 N·m load and 48 V source voltage; the marked regions show reference capture and the response to the load step.

At 1800 rpm, the controller is stable under no load, but applying the 10 N·m load at $t = 0.5$ s reveals a clear performance limit: the speed drops to 1576 rpm, which is a 12.444% undershoot, and the magnified response shows a triangular waveform, indicating that at this speed the controller prioritises torque production over speed regulation. This identifies the present high-speed operating boundary and motivates the future work discussed in Section 4. Figure 8 shows the response, and Table 3 presents the numerical results for all four speeds.

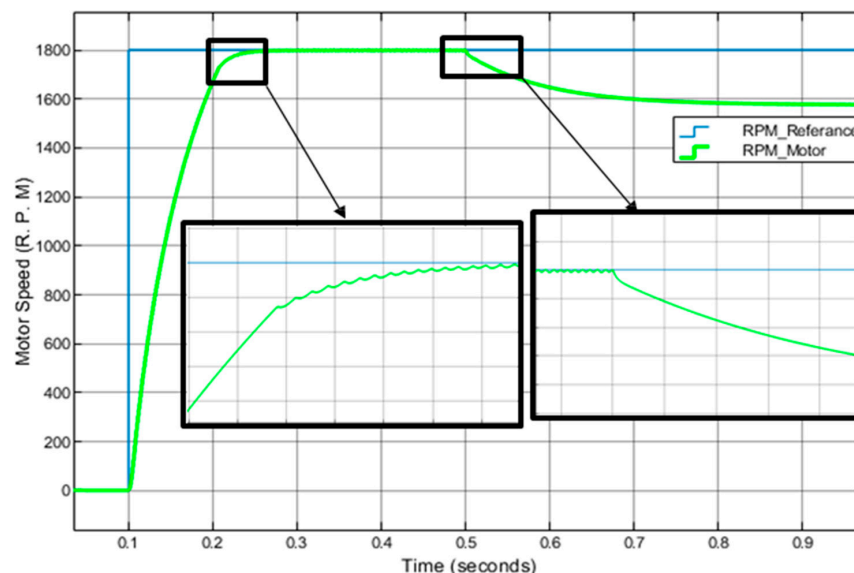


Figure 8. 1800 rpm with preselected torque of -10 N·m and source voltage of 48 V.

3.2. Variable-Load Robustness

Robustness to load was tested at a constant 1000 rpm and 48 V with the torque increasing in steps through a range up to and beyond the 10 N·m design point. As Figures 9 and 10 show, the peak speed shifts only from 1005 rpm to 999.3 rpm as the load

increases, and the lowest speed improves as the torque approaches the 10 N·m design target before diverging under heavier loads—confirming that the adaptive logic is optimised around this operating point. Overshoot decreases and undershoot increases with increasing load, but the RMS speed remains essentially constant, indicating stable energy delivery across the load range.

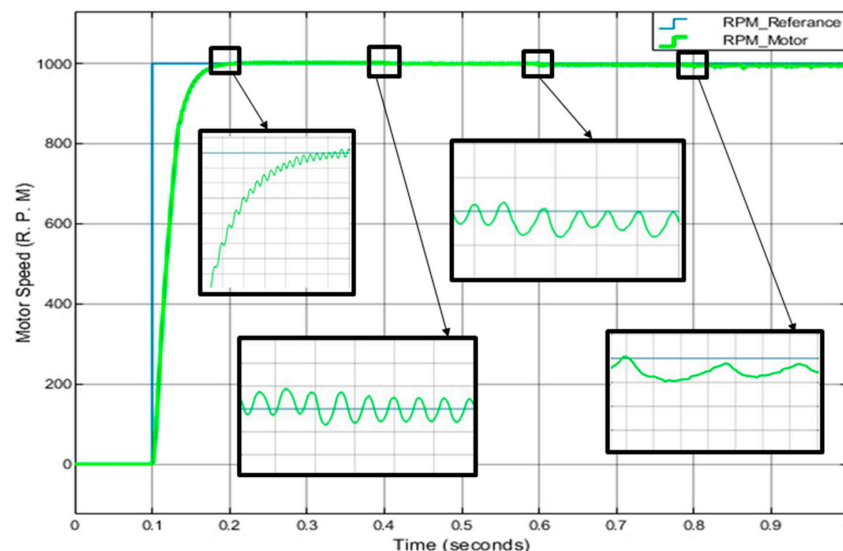


Figure 9. Controller performance in a single simulation for variable torque.

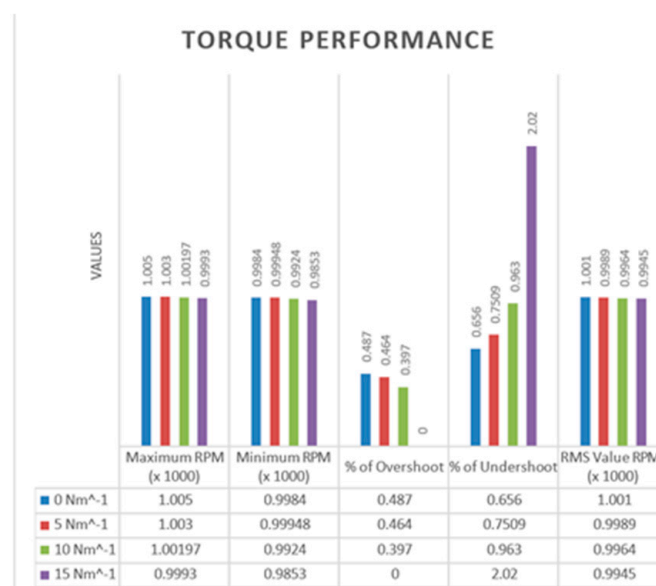


Figure 10. Controller performance at 1000 rpm and 48 V under incrementally applied torque loads; inset windows magnify the transient at each load step.

To assess the robustness of the adaptive PID architecture, we conducted a variable load test at a constant 1000 rpm with a 48 V source. As shown in Figure 9, the controller demonstrated versatile performance across a multi-torque spectrum. Detailed inset windows provide magnified views of the transient response as torque loads were incrementally applied. Data from Figure 10 indicate that while the peak rpm marginally shifted from 1005 rpm to 999.3 rpm as load increased, the system maintained high operational integrity. Interestingly, the lowest rpm values showed nonlinear behaviour, improving as torque approached the 10 Nm⁻¹ design target before diverging under heavier loads. This trend suggests that the adaptive logic is specifically optimised for the 10 Nm⁻¹ benchmark.

3.3. Supply Voltage Robustness and Comparative Evaluation

Resilience to supply variation was assessed over the industry-standard 5–10% band, i.e., a 48 V system operating between 43.2 V and 52.8 V, and at 36 V, 48 V, and 60 V configurations. Across this range the proposed adaptive PID tracked the reference tightly with minimal overshoot or undershoot even under load. Benchmarked against the conventional fixed-gain PID and the FPA-tuned PID under identical conditions, the proposed controller held overshoot below 1% at high speed and produced consistently lower torque ripple than both baselines, as summarised in Table 4. The fixed PID, tuned at the nominal point, showed the largest overshoot and ripple when the operating point moved; the FPA-tuned PID improved on it but, lacking online adaptation, still trailed behind the proposed scheme once the plant drifted. These results support the conclusion that the hybrid auto-tuner/fast-PID combination provides better adaptability and harmonic stability under the variable electrical conditions of an EV drive. Figure 11 shows controllers' output with variable voltage, with a speed of 1000 rpm and a torque of 10 Nm. In Figure 11 the no. (i) figures belong 36 volt, (ii) figures belongs 48 volt and (iii) figures belongs to 60 volt simulation output proposed controller and others controller respectively.

Table 4. Comparative performance at 1000 rpm, 10 N·m, and 48 V

Metric	Fixed PID	FPA-Tuned PID	Proposed Adaptive PID
Overshoot (%)	≈4.8	≈2.1	0.40
Settling time (s)	0.18	0.11	0.052
Steady-state error (rpm)	≈6	≈3	<1
Relative torque ripple	High	Medium	Low
Online adaptation	No	No	Yes

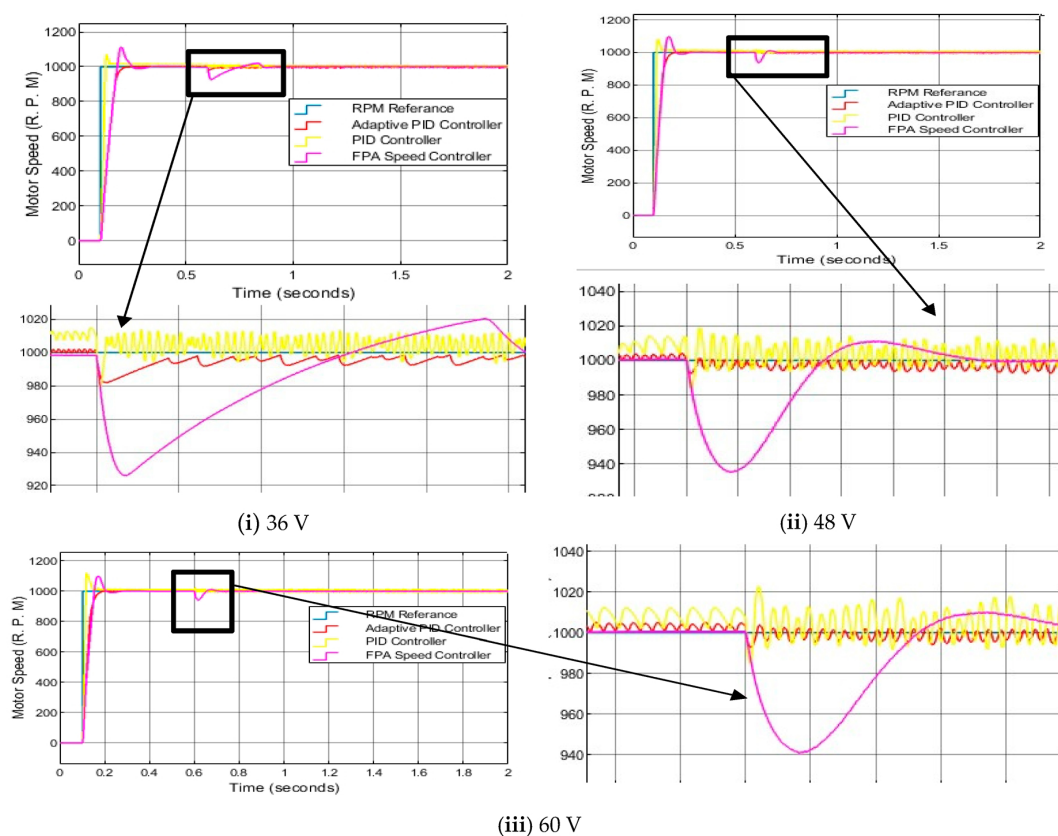


Figure 11. Controllers' output with variable voltage, with speed of 1000 rpm and torque of 10 Nm.

Limitations: The results above were obtained via simulation with the small-signal model of (6); although the model includes the dominant electrical and mechanical dynamics and the commutation-dependent torque, magnetic saturation, cogging, and inverter dead-time are only approximately represented, and the 1800 rpm case shows that the present tuning does not yet handle large load steps at high speed. Experimental validation on a physical dynamometer setup is therefore the essential next step, as discussed in Section 4.

4. Conclusions

This paper presented a hybrid adaptive PID controller for BLDC motor drives in electric vehicles that combines a frequency-response-based PID auto-tuner with a parallel fast PID on a cascaded speed/voltage loop. A complete analytical model of the BLDC machine phase voltage equations, electromagnetic torque, mechanical dynamics, and speed-to-voltage transfer function was derived, the PID transfer functions to be adapted were defined, and the adaptation law was specified as an explicit algorithm. Using Hall sensor feedback, the scheme tracks the reference without requiring an exact analytical feedback model, and the fast PID supplies the rapid corrective action that bridges the auto-tuner's estimation interval. In MATLAB/Simulink, benchmarked against a conventional fixed-gain PID and an FPA-tuned PID, the proposed controller held overshoot below 1% from 1000 to 1500 rpm, maintained low torque ripple under variable load and supply voltage, and outperformed both baselines on overshoot, settling time, and steady-state error. The 1800 rpm case, where a sudden 10 N·m load produced a 12.4% undershoot, marks the present operating limit. Future work will focus on (i) experimental validation using a physical BLDC dynamometer with a dSPACE/FPGA real-time controller, reporting measured rise time, overshoot, ripple, and efficiency against the simulation; (ii) extending the model to include magnetic saturation, cogging, and inverter dead-time; and (iii) improving the high-speed/heavy-load transient.

Author Contributions: Conceptualization, M.M. and S.M.R.I.; methodology, M.M.; software, M.M.; validation, M.M., S.M.R.I. and S.M.A.M.; formal analysis, M.M.; investigation, M.M.; resources, S.M.A.M.; data curation, M.M.; writing—original draft preparation, M.M.; writing—review and editing, S.M.R.I. and S.M.A.M.; visualization, M.M.; supervision, S.M.A.M.; project administration, S.M.A.M.; funding acquisition, S.M.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors Md Mahmud and S. M. Rakibul Islam were employed by the company Resa Power. The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Biswas, P.; Rashid, A.; Habib, A.A.; Mahmud, M.; Motakabber, S.M.A.; Hossain, S.; Lei, T.M. Vehicle to Grid: Technology, Charging Station, Power Transmission, Communication Standards, Techno-Economic Analysis, Challenges, and Recommendations. *World Electr. Veh. J.* **2025**, *16*, 142. [CrossRef]
2. Mahmud, M.; Motakabber, S.M.A.; Alam, A.Z.; Nordin, A.N. Control BLDC motor speed using PID controller. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 477–481. [CrossRef]
3. Mahmud, M.; Motakabber, S.M.A.; Alam, A.Z.; Nordin, A.N. Utilizing of flower pollination algorithm for brushless DC motor speed controller. In *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, Dhaka, Bangladesh, 21–23 December 2020; IEEE: New York, NY, USA, 2020; pp. 1–5.
4. Kuantama, E.; Vesselenyi, T.; Dzitac, S.; Tarca, R. PID and Fuzzy-PID control model for quadcopter attitude with disturbance parameter. *Int. J. Comput. Commun. Control* **2017**, *12*, 519–532. [CrossRef]

5. Potnuru, D.; Mary, K.A.; Babu, C.S. Experimental implementation of Flower Pollination Algorithm for speed controller of a BLDC motor. *Ain Shams Eng. J.* **2019**, *10*, 287–295. [[CrossRef](#)]
6. Suganthi, S.; Karpagam, R. Dynamic performance improvement of PMSM drive using fuzzy-based adaptive control strategy for EV applications. *J. Power Electron.* **2023**, *23*, 510–521. [[CrossRef](#)]
7. Pradana, A.B.; Haque, M.M.; Krismanto, A.U.; Setiadi, H.; Nadarajah, M. Robust PID control of EV aggregator for C-FCAS using sequential multi-objective optimization. *IEEE Access* **2024**, *12*, 12602–12617. [[CrossRef](#)]
8. Davoudkhani, I.F.; Zare, P.; Abdelaziz, A.Y.; Bajaj, M.; Tuka, M.B. Robust load-frequency control of islanded urban microgrid using 1PD-3DOF-PID controller including mobile EV energy storage. *Sci. Rep.* **2024**, *14*, 13962. [[CrossRef](#)] [[PubMed](#)]
9. Shenbagalakshmi, R.; Mittal, S.K.; Subramanian, J.; Vengatesan, V.; Manikandan, D.; Ramaswamy, K. Adaptive speed control of BLDC motors for enhanced electric vehicle performance using fuzzy logic. *Sci. Rep.* **2025**, *15*, 12579. [[CrossRef](#)] [[PubMed](#)]
10. Saiteja, P.; Ashok, B.; Upadhyay, D. Evaluation of electric vehicle performance characteristics for adaptive supervisory Self-Learning-Based SR motor energy management controller under Real-Time driving conditions. *Vehicles* **2024**, *6*, 509–538. [[CrossRef](#)]
11. Chu, L.; Li, H.; Xu, Y.; Zhao, D.; Sun, C. Research on longitudinal control algorithm of adaptive cruise control system for pure electric vehicles. *World Electr. Veh. J.* **2023**, *14*, 32. [[CrossRef](#)]
12. Vitelli, M. Small-Signal Modeling, Comparative Analysis, and Gain-Scheduled Control of DC–DC Converters in Photovoltaic Applications. *Electronics* **2025**, *14*, 4308. [[CrossRef](#)]
13. Cheong, S.-Y.; Safonov, M.-G. Slow-Fast Controller Decomposition Bumpless Transfer for Adaptive Switching Control. *IEEE Trans. Autom. Control* **2012**, *57*, 721–726. [[CrossRef](#)]
14. Xu, S.; Tian, X.; Wang, C.; Qin, Y.; Lin, X.; Zhu, J.; Sun, X.; Huang, T. A Novel Coordinated Control Strategy for Parallel Hybrid Electric Vehicles during Clutch Slipping Process. *Appl. Sci.* **2022**, *12*, 8317. [[CrossRef](#)]
15. Zhang, P.; Li, X.; Yu, L.; Zhang, Z.; Dunkin, F.; Liu, H.; Li, Z. Boosting Learning Efficiency in Few-Shot Tasks with Layer-Adaptive PID Control. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: New York, NY, USA, 2026; Volume 48, pp. 7079–7096. [[CrossRef](#)] [[PubMed](#)]
16. Almutairi, S.; Anayi, F.; Packianather, M.; Shouran, M. Comparative Assessment of PSO-Tuned Hybrid Fuzzy Controllers for Load Frequency Control in a Two-Area Hybrid Power System Under Nonlinear and Parametric Uncertainty. *Energies* **2026**, *19*, 2677. [[CrossRef](#)]
17. Ding, Z.; Yao, Y.; Gao, S.; Yang, X.; Li, C.; Ren, J.; Dong, J.; Wu, J.; Ma, F.; Liu, X. Research on Variable Universe Fuzzy Adaptive PID Control System for Solar Panel Sun-Tracking. *Sustainability* **2026**, *18*, 1503. [[CrossRef](#)]
18. Wang, J.; Yan, S.; Bao, H.; Chen, C.; Yu, D.; Li, J.; Chen, X.; Dou, R.; Tang, Y.; Li, S. Reinforcement-Learning-Based Adaptive PID Depth Control for Underwater Vehicles Against Buoyancy Variations. *J. Mar. Sci. Eng.* **2026**, *14*, 323. [[CrossRef](#)]
19. Ekinici, S.; Izci, D.; Tümen, V.; Jabari, M.; Çelik, E.; Elrashidi, A. Intelligent Control of Magnetic Ball Suspension Systems via a Novel Hyperbolic Tangent PID Controller Tuned by the Artificial Lemming Algorithm. *Biomimetics* **2026**, *11*, 205. [[CrossRef](#)] [[PubMed](#)]
20. Ekinici, S.; Izci, D.; Jabari, M.; Çelik, E.; Bajaj, M.; Vishnuram, P.; Rubanenko, O. A novel hyperbolic tangent-based PID controller tuned by the artificial lemming algorithm for nonlinear steam condenser pressure control. *Sci. Rep.* **2026**, *16*, 4600. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.