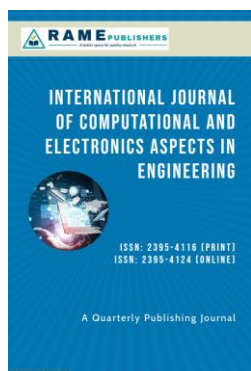


Improving Transient Performance of Hall-Commutated BLDC Drives via a Compact ANN Controller Trained from PID-Tuned Data

Omar Gheyath Ibrahim* 

Department of Medical Instrumentation Techniques Engineering, Polytechnic College Mosul, Northern Technical University, Mosul, Iraq.

*Correspondence: mti.lec17.omar@ntu.edu.iq



Abstract: Speed requirement are high speed tracking and well damped torque transients of brushless DC (BLDC) drives during starting and load disturbances. This paper introduces a Model of Simulink for Hall-effect sensor BLDC drive with six-step commutation, and compares the performance of a classical discrete-time PID speed controller to that of an ANN controller as a plug-and-play alternative. Where the PID gains are obtained through MATLAB PID Tuner (based on frequency response), supervised data to train a feedforward regression ANN with one input, and one hidden layer containing 20 neurons and an accelerated output. At the rated condition (750 rpm, 50 N·m), compared to PID, ANN exhibits faster speed transient response and less electromagnetic torque overshoot. Validation at 500, 750, and 1000 rpm confirms the improvement with a significant reduction in rise time and settling time together with suppression of torque peaks. The disturbance rejection is evaluated by a step load test showing that the ANN yields smaller speed deviations and smoother torque transitions. These results demonstrate that an appropriately trained ANN can provide better transient performance than a tuned PID controller in BLDC drives without altering the inverter or commutation scheme.

Keywords: BLDC motor drive; Hall-effect sensors; neural network controller; feedforward regression ANN; torque overshoot reduction.

Article – Peer Reviewed

Received: July 30, 2026

Accepted: September 16, 2026

Published: September 27, 2026

Copyright: © 2026 RAME Publishers

This is an open access article under the CC BY 4.0 International License.



<https://creativecommons.org/licenses/by/4.0/>

Cite this article: Omar Gheyath Ibrahim, “Improving Transient Performance of Hall-Commutated BLDC Drives via a Compact ANN Controller Trained from PID-Tuned Data”, *International Journal of Computational and Electronic Aspects in Engineering*, RAME Publishers, vol. 7, issue 4, pp. 466-478, 2026.

<https://doi.org/10.26706/ijceae.7.4.20260813>

1. Introduction

BLDC have about 85% to 90% efficiency compared to brushed machines which makes these motor sheets the most preferred current types of motors used in modern electromechanical systems [1]. These properties make BLDC drives interesting for a wide field of applications, comprising, robotics, electric vehicles and industrial automation as well as aerospace actuators [2]. In parallel, recent developments in electric-vehicle energy systems have highlighted the importance of high-efficiency power-transfer technologies, including photovoltaic-integrated wireless power transfer for EV charging applications [3]. Nevertheless, the control strategy used will have a significant influence on the performance of a BLDC drive, more intensively when the transients are present (start-up, load disturbances and rapid changes in speed reference) [5]. Fast speed tracking with smooth torque production and limited overshoot is still a main challenge for these operating scenarios [4]. The conventional (PID) control is still one of the most commonly adopted methods in BLDC drive systems owing to its structural simplicity, easy implementation, and intuitive parameter interpretation [5]. However, PID performance tends to be sensitive with respect to parameter tuning and operating point variations [6]. In practice, BLDC drive shows nonlinearities due to inverter switching, trapezoidal back-EMF characteristics, commutation and mechanical load dynamics [7][23]. They result in nonlinearities that may exhibit undesirable dynamic behavior, e.g., increased settling time for speed and overshoot / ripple on electromagnetic torque, particularly when the controller is tuned only for narrow operating confines [8]. To overcome those limitations, many researchers in motor drive field turn their eyes into data-driven and intelligent control methods. Artificial neural networks (ANNs) are especially

promising in that they provide a flexible framework to approximate nonlinear input–output mappings without requiring an explicit analytical model of the full system dynamics. With adequate training, an ANN-based controller can imitate or improve the performance of a baseline controller while adding better adaptability and robustness for uncertainties and nonlinearities [9]–[12]. That’s why neural controllers are strong candidates for high-performance BLDCs drives since transient response and torque quality is vital. In this work, a comparative control architecture is elaborated to model and control the BLDC motor drive, in two different approaches. As a first step, classical PID controller is designed and tuned by means of the impressive automatic tuning function in MATLAB which provides optimal set of gains for treatment according to operating conditions. Using this tuned PID controller, we can now create representative input–output data that reflects how the drive will behave in closed-loop operation. A two-layer neural network controller is trained to substitute the conventional PID controller using these data. The ANN-based controller that results is tested under analogous tests and demonstrated to increase speed dynamic response (lower response/settling time) while lowering torque overshoot during transients. The achieved results confirm that the proposed neural controller is able to attain the baseline regulation objective while significantly improving the performance indices of primary importance for BLDC applications. In RNN-trained methods PID-generated data are only used for the initial supervised training, during evaluation the ANN takes its place in the controller loop and its improved performance is not linked with direct limitations of PID. The remainder of this paper is organized as follows: Section II presents the BLDC motor model and the drive system formulation. Section III details the baseline PID control design and MATLAB-based tuning procedure. Section IV introduces the neural network architecture, training data generation, and learning procedure. Section V provides simulation results and performance comparisons in terms of speed tracking dynamics and torque transient characteristics. Finally, Section VI concludes the paper and outlines future research directions [13]–[16].

2. Literature review

Classical PID/PI regulators are still widely used in BLDC drives due to their simplicity and ease of implementation, but in the presence of switching effects, trapezoidal back-EMF shape, commutation nonlinearities or variations in operating point, transition they may suffer a reduced performance that usually translates into conservative tuning for low torque transients [19][24]. Thus, there are many studies focusing on smart and data-based controllers (e.g. neural networks, fuzzy logic and hybrid schemes) that can characterize nonlinear control laws as well as provide improved performance in robustness and transience quality without the necessity of an entire analytical model [18][20][22].

In parallel, accurate modeling and advanced control of power-electronic conversion systems remain important for improving transient behavior, stability, ripple characteristics, and overall system performance [17].

Specifically, in the BLDC subjective boundary, ANN based controllers have either been used for online tuning of PID parameters or to enable replacement of conventional controllers by learning an input–output mapping using simulation or experimental data [20][21]. In the current paper further extending upon the controller-replacement paradigm by training a compact feedforward ANN to replace a tuned PID teacher baseline, deploying an ANN as a replacement block in Simulink model for Hall-commutated BLDCs and tracking discrete application-specific metrics that reflect speed transient indices to ultimately quantify improvements during torque overshoot against fixed versus time-varying load conditions [18][19].

3. System Modeling and Simulink Implementation of the BLDC Drive

3.1. Drive system overview

MATLAB/Simulink is used to build and test the proposed BLDC drive. This system consists of a DC link supply, a three-phase two-level VSI, trapezoidal-back-EMF permanent-magnet machine model BLDC mode with Hall-effect position sensing and an outer-loop speed controller [26][27][28]. which is used as a control objective of the speed, at rated conditions, and making improvements to transient performance during reference changes by reducing torque overshoot and speeding up the speed response. A Hall-based six-step commutation mechanism is adopted. The three Hall signals (h_a, h_b, h_c) are decoded by a commutation logic subsystem to generate the discrete phase excitation sequence. The speed controller generates a scalar command that scales this commutation pattern; the resulting three-phase command is then converted into six inverter gate pulses that drive the MOSFET-based VSI, as shown in Fig. 1.

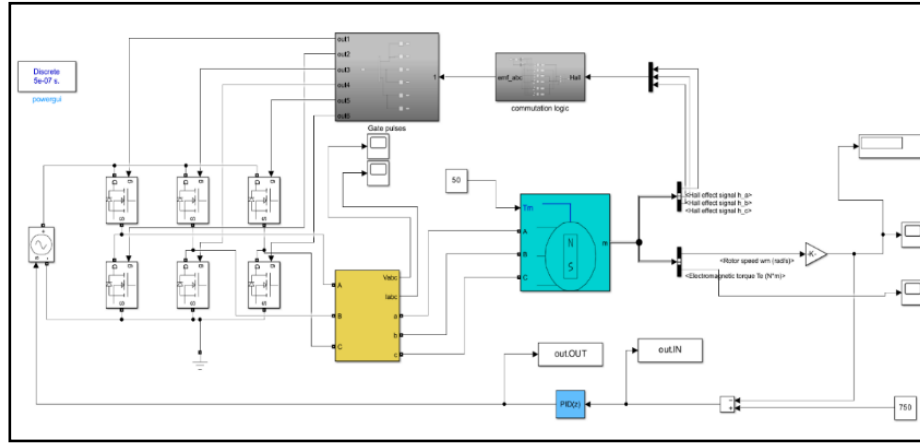


Figure 1. Overall Simulink model of the BLDC drive

3.2. Electrical model of the trapezoidal back-EMF PM machine

The electrical dynamics of the three-phase machine in phase variables can be expressed as Equation (1):

$$v_a = R_s i_a + L_s \frac{di_a}{dt} + e_a, v_b = R_s i_b + L_s \frac{di_b}{dt} + e_b, v_c = R_s i_c + L_s \frac{di_c}{dt} + e_c \quad (1)$$

where $v_{a,b,c}$ and $i_{a,b,c}$ are phase voltages and currents, respectively, and R_s and L_s are the per-phase stator resistance and inductance [25].

For BLDC operation, the back-EMF is trapezoidal and depends on mechanical speed ω_m and electrical rotor position θ_e . It is modeled as Equation (2):

$$e_a = K_e \omega_m f_a(\theta_e), e_b = K_e \omega_m f_b(\theta_e), e_c = K_e \omega_m f_c(\theta_e) \quad (2)$$

where K_e is the back-EMF constant and $f_{a,b,c}(\theta_e)$ are normalized trapezoidal functions phase-shifted by 120° electrical. In the adopted machine model, the back-EMF flat region is set to 120° , which is consistent with six-step commutation and 120° conduction [25][26].

3.3. Electromagnetic torque and mechanical dynamics

The electromagnetic torque can be written using the power conversion relation as Equation (3) [25]:

$$T_e = \frac{e_a i_a + e_b i_b + e_c i_c}{\omega_m} \quad (3)$$

The mechanical subsystem is governed by Equation (4) [19]:

$$J \frac{d\omega_m}{dt} + B \omega_m = T_e - T_L - T_f \text{sgn}(\omega_m) \quad (4)$$

where J is the rotor inertia, B is the viscous damping coefficient, T_L is the load torque, and T_f represents static friction. In this work, the static friction is set to zero in the machine block.

3.4. Hall-based six-step commutation logic

Rotor position information is provided by three Hall-effect signals. The commutation logic subsystem maps the Hall state $\mathbf{h} = [h_a \ h_b \ h_c]^T$ into a three-phase discrete excitation pattern as shown in Equation (5):

$$\mathbf{s}(\theta_e) = [s_a(\theta_e) \ s_b(\theta_e) \ s_c(\theta_e)]^T, s_x \in \{-1, 0, +1\} \quad (5)$$

One commutation sector drives two phases, with one floating to achieve 120° conduction. The output of the commutation logic block (shown as in Simulink) thus corresponds to the desired phase polarity sequence for forming the inverter switching commands [26].

To decouple commutation sequencing from amplitude regulation, the scalar controller output $u(t)$ scales the commutation vector as shown in Equation (6):

$$\mathbf{v}^*(t) = u(t) \mathbf{s}(\theta_e) \quad (6)$$

This structure enables a fair controller replacement: the baseline PID(z) block and the proposed neural-network controller both generate the same scalar command $u(t)$, while the commutation and inverter subsystems remain identical as shown in Fig. 2.

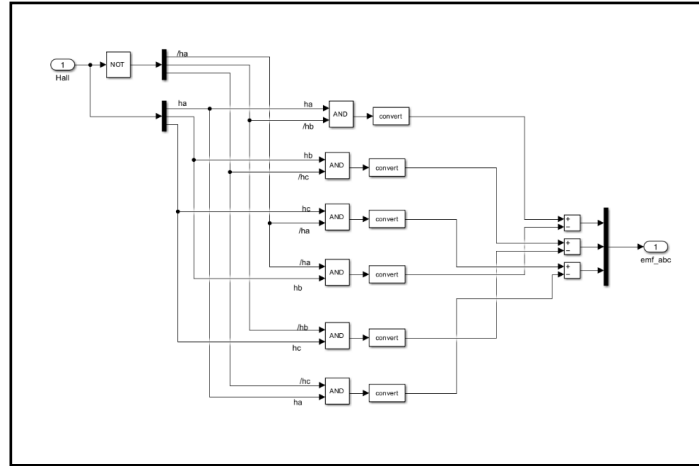


Figure 2. Hall-based commutation logic subsystem used to generate the commutation pattern from Hall signals.

3.5. Gate pulse generation and inverter switching

The drive employs a three-phase two-level VSI implemented using MOSFET devices with antiparallel diodes. Gate signals are derived from the commanded phase pattern $\mathbf{v}^* = [v_a^*, v_b^*, v_c^*]^T$ via sign-based logic implemented using comparators [19][26] as shown in Equation (7):

$$g_{\text{upper},x} = 1\{v_x^* > 0\}, g_{\text{lower},x} = 1\{v_x^* < 0\}, x \in \{a, b, c\} \quad (7)$$

As a result, a positive phase command turns on the corresponding upper switch, while a negative command turns on the corresponding lower switch. This logic is consistent with six-step commutation and the trapezoidal back-EMF machine model used in the simulation as shown in Fig. 3.

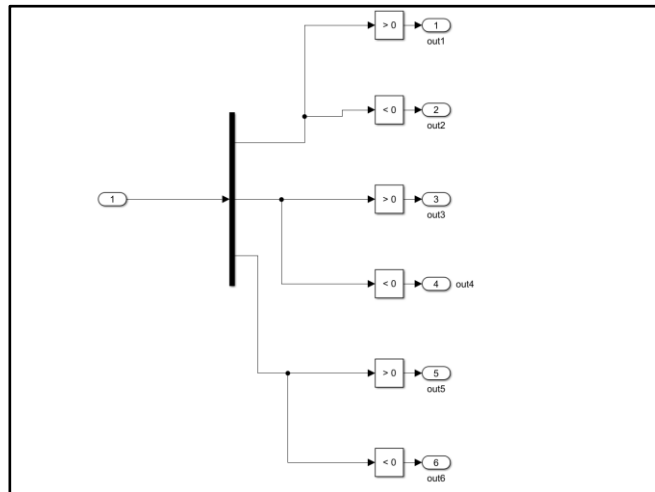


Figure 3. Gate pulse generation subsystem

3.6. Simulation parameters and rated operating point

Tables 1 and 2 show the machine and converter parameters used in the developed Simulink model. Rated reference speed for closed-loop tests set to rpm. If needed for an analysis, this can be put in SI angular speed form as Equation (8):

$$\omega^* = 750 \text{ rpm} = 750 \left(\frac{2\pi}{60} \right) = 78.54 \text{ rad/s} \quad (8)$$

Table 1. Motor parameters used in the Simulink model

Parameter	Symbol	Value	Unit
Stator phase resistance	R_s	0.7	Ω
Stator phase inductance	L_s	$2.7\text{e} - 3$	H
Flux linkage (PM-established)	λ_f	0.1194	V·s
Back-EMF flat area	—	120	deg
Rotor inertia	J	0.0027	kg·m ²
Viscous damping (Simulink field: F)	B	0.00049	N·m·s
Pole pairs	p	4	—
Static friction	T_f	0	N·m
Initial conditions	$[\omega_m, \theta_m, i_a, i_b]$	$[0, 0, 0, 0]$	(rad/s, deg, A, A)
Rotor flux position at $\theta = 0$	—	90° behind phase-A axis	—

Table 2. Inverter and supply parameters used in the Simulink model

Item	Parameter	Value	Unit
DC link source	Type	DC	—
DC link voltage	V_{dc}	160	V
Power switch model	Device	MOSFET + antiparallel diode	—
MOSFET on-resistance	R_{on}	0.1	Ω
Diode resistance	R_d	0.01	Ω
Diode forward voltage	V_f	0	V
Diode inductance	L_{on}	0	H
Snubber resistance	R_s	1×10^5	Ω
Snubber capacitance	C_s	∞ (disabled)	F
Rated reference speed	ω^*	750	rpm

3.7. Rated Operating Point and Simulation Settings

The rated operating point is defined by a reference speed 750 rpm and a load torque equal to 50 N·m (conversion of rev/min into rad/s made applying in Equation (9):

$$\omega^* = \frac{2\pi}{60} n^*, \quad n^* = 750 \text{ rpm} \quad (9)$$

The load torque is applied as in Equation (10):

$$T_m = 50 \text{ N}\cdot\text{m}. \quad (10)$$

The simulation is executed in discrete mode using powergui with sample time as shown in Equation (11):

$$T_s = 5 \times 10^{-7} \text{ s}. \quad (11)$$

The phase currents exhibit a typical BLDC 120° conduction pattern with approximate peaks around $\pm 50\text{A}$.

4. PID(z) Speed Controller Design and MATLAB Auto-Tuning

In the outer loop of the proposed BLDC drive, a discrete time PID controller is used as baseline speed regulator as shown in Equation (12). The controller input is the speed tracking error [24]

$$e(k) = \omega^*(k) - \omega(k) \quad (12)$$

where $\omega^*(k)$ is the reference speed (set to the rated value of 750 rpm in the reported tests) and $\omega(k)$ is the measured rotor speed. The controller output $u(k)$ provides a scalar control command that modulates the inverter excitation by scaling the six-step commutation pattern, hence shaping the developed electromagnetic torque and the resulting speed transient.

The PID block is implemented in parallel form and executed in discrete time. The drive model is simulated using the powergui discrete environment with a global sampling period of as shown in Equation (13).

$$T_s = 5 \times 10^{-7} \text{ s} \quad (13)$$

and its block sample time set to -1 (inherited) so that the controller shares a sample time with the overall discrete-time simulation. The PID gains are obtained using MATLAB's built-in automated tuning capability [24] (PID Tuner) through the Frequency Response Based method. Equation (14) illustrate the gain values used in this study are:

$$K_p = 105, K_i = 0.1, K_d = 0.1 \quad (14)$$

The derivative filter coefficient is set to $N = 0$ in the selected block configuration. This tuned discrete PID controller serves as the conventional benchmark for performance comparison under the same commutation logic, inverter configuration, and rated operating conditions.

5. ANN Controller

5.1. Motivation and Implementation Concept

The PID controller is replaced by an Artificial Neural Network (ANN) controller as a drop-in replacement of the PID controller without changing inverter, commutation and machine model for enhanced performance in transients. The ANN occupies the position of the PID block, and has the same input (speed error $e(k)$), and output (output $u(k)$) as that of a PID controller. Hence the improvement developed should be due to controller mapping, and no structure change in the drive system as illustrated in Fig. 4.

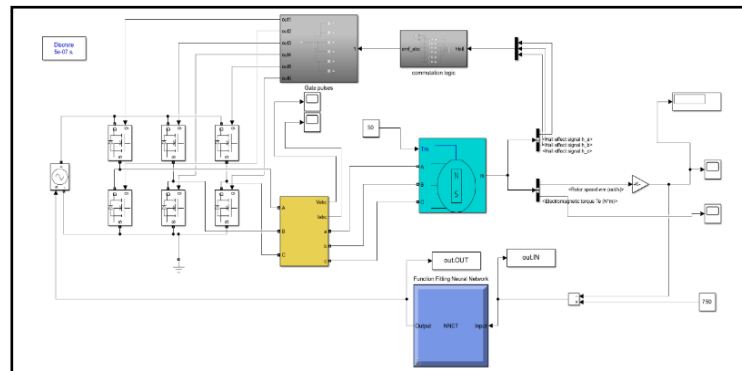


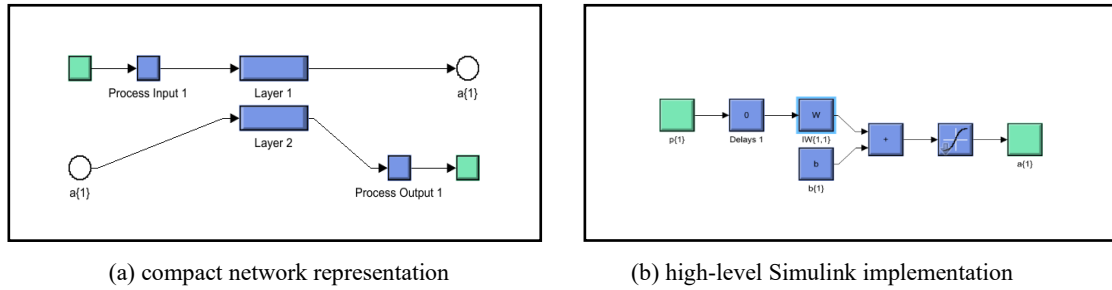
Figure 4. Overall Simulink model of the Hall-sensor-based BLDC drive with six-step commutation and ANN speed controller

5.2. Network Structure

A feedforward regression network is adopted. The ANN consists of: one input (speed error), one hidden layer with 20 neurons and a nonlinear activation function, one linear output neuron. A compact representation as in Equation (15):

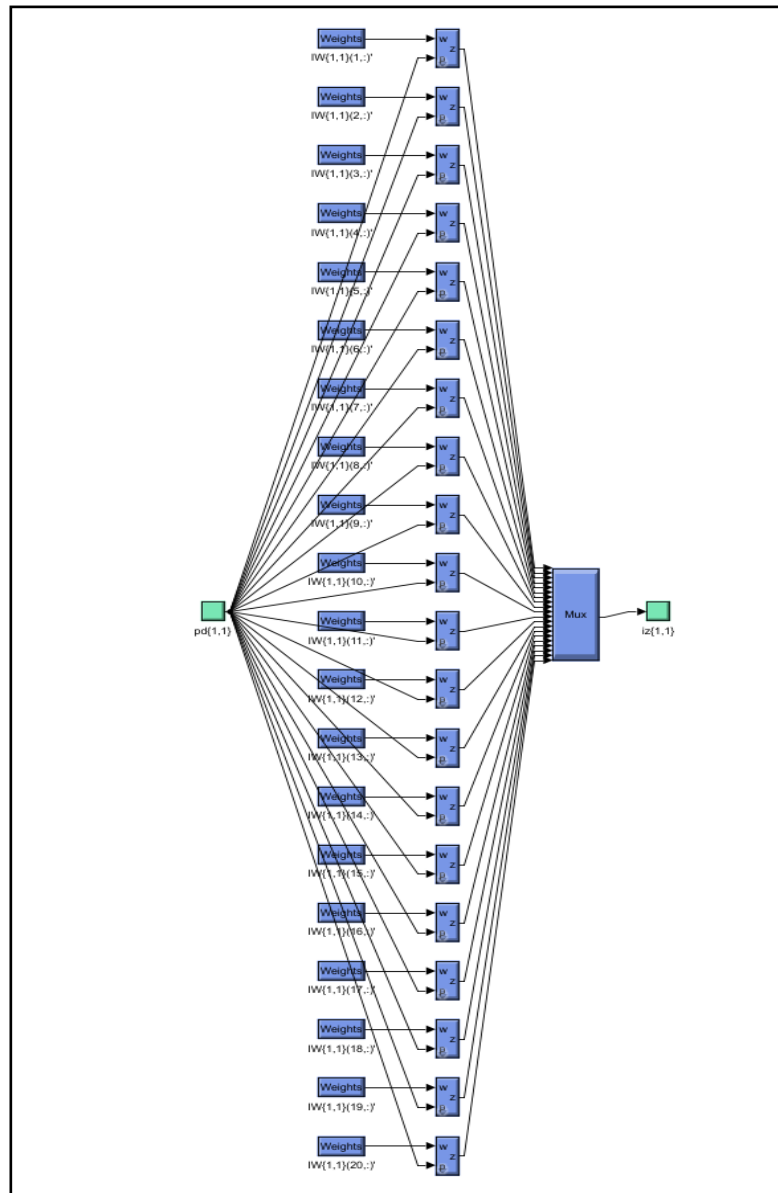
$$u(k) = W_2 f(W_1 e(k) + b_1) + b_2 \quad (15)$$

As shown in Fig. 5. Where W_1, W_2 are weight matrices, b_1, b_2 are bias terms, and $f(\cdot)$ is a nonlinear activation.



(a) compact network representation

(b) high-level Simulink implementation



(c) expanded connectivity of the 20-neuron hidden layer

Figure 5. ANN controller structure

5.2.1. Training data generation

The training data are generated from closed-loop simulations with the tuned PID controller. The ANN input signal(s) the same as the PID input and the corresponding PID output are saved during these runs. The dataset generated hence captures the control relationship of a six-step commutated BLDC drive under that nonlinear, switched dynamics. The ANN is then

trained in a supervised manner to learn the mapping implemented by the tuned PID. In short, the PID is not a control law to be copied exactly but rather a baseline data generator for initial supervised training.

5.2.2. Deployment and evaluation protocol

After training, the PID block is replaced by the ANN block. Both controllers (PID and ANN) are evaluated under identical operating conditions, including rated reference speed and rated load torque. This protocol provides a fair and direct comparison between the conventional tuned PID(z) baseline and the proposed ANN controller in terms of speed transient quality and electromagnetic torque behavior. During testing, the PID block is not used online; the ANN is the sole active controller in the closed loop.

5.2.3. Implementations Requirements and Computational Overhead

The proposed ANN controller has a compact computational structure consisting of one scalar input. One hidden layer with 20 neuros and one scalar output. Based on Eq. 15 the network contains 61 trainable parameters including 20 input to hidden weights 20 hidden layer biases 20 hidden to output weights and one output bias. A direct forward pass implementation requires approximately 40 multiplication 40 additions and 20 evaluations of the hidden layer nonlinear activation function per control update. Therefore, the computational overhead increases linearity with the number of hidden neuros, where $N_h=20$ in the present implementation. When single precision floating point representation is used, the ANN parameters require approximately 244 bytes of memory excluding temporary variables and activation function implementation overhead. The Simulink model uses a global discrete simulation step of 5×10^{-7} s, however this numerical simulation step shouldn't be interpreted as an expediently verified microcontroller execution time. Processor specific execution latency depends on the target hardware numerical precision code generation method and implementation of the nonlinear activation function. Consequently, processor in the loop and hardware timing measurements are considered an important direction for future experimental validation.

6. Results and Discussion

All the results are taken at rated reference speed of 750 rpm and rated load torque of 50 N.m; identical motor and inverter parameters along with same six-step commutation logic is used. The blue traces are the ANN controller, and the red traces are the tuned PID controller. Even though the ANN was originally trained and initialized from a dataset taken from a teacher-tuned PID controller, results shown are from independent closed-loop control using the ANN.

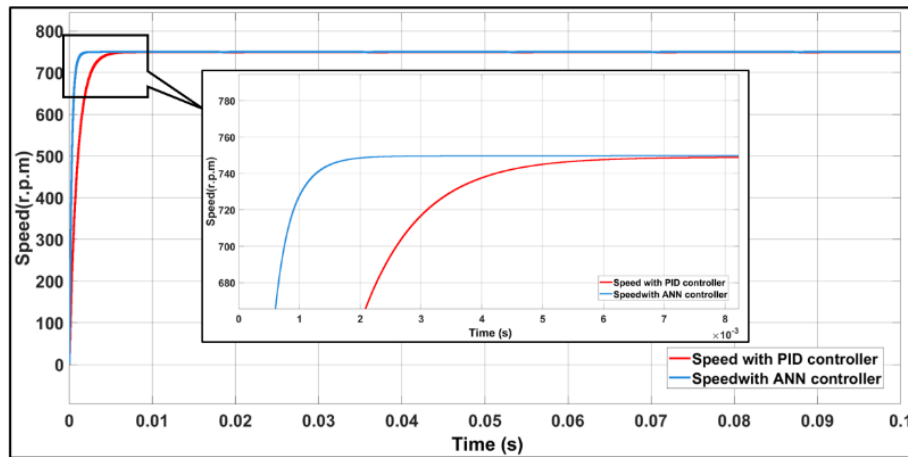


Figure 6. Speed response comparison (PID vs ANN) at 750 rpm reference and 50 N.m load.

Fig. 6 shows the speed response of the PID and ANN controllers, Quantitative indices are computed using standard time-domain step-response metrics. The ANN controller substantially accelerates the transient compared to the tuned PID: the rise time (10–90%) is reduced from 2.1 ms (PID) to 0.597 ms (ANN), and the settling time ($\pm 2\%$) is reduced from 3.7 ms (PID) to 1.1 ms (ANN). Both controllers exhibit negligible speed overshoot, with the PID overshoot at 0.0021% and the ANN overshoot essentially 0%, and both converge to the same steady-state speed.

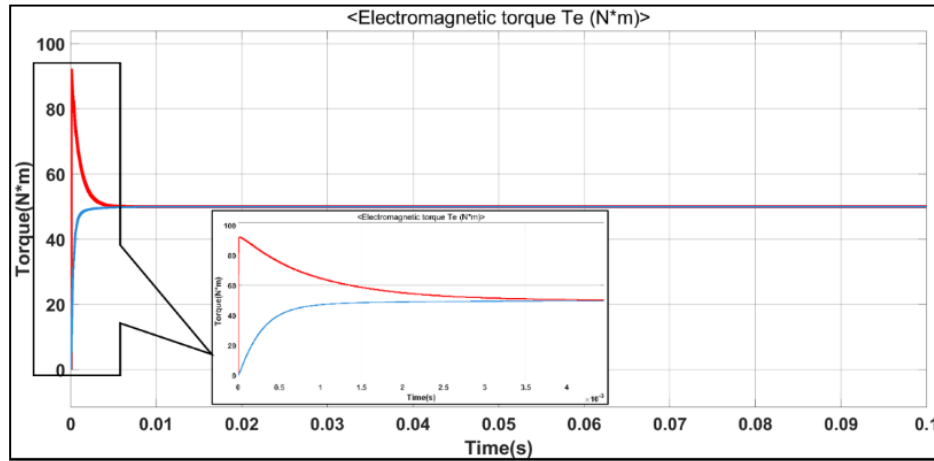


Figure 7. Electromagnetic torque transient comparison (PID vs ANN) at 50 N.m rated load.

Fig. 7. presents the electromagnetic torque transient. A key difference is observed in the peak torque behavior. The tuned PID produces a pronounced initial torque surge with a peak torque of 92.361 N.m, while the steady-state torque is 50.01 N.m, corresponding to a torque overshoot of Equation (16):

$$M_p = \frac{T_{peak} - T_{ss}}{T_{ss}} \times 100\% = 84.687\% \quad (16)$$

In contrast, the ANN controller reaches essentially the same steady-state torque (49.995 N.m) with an almost overshoot-free torque profile: $T_{peak} \approx 50$ N.m and overshoot of only 0.00893%. This indicates that the ANN achieves faster speed regulation while dramatically reducing excessive transient torque, which is beneficial for limiting mechanical stress and mitigating current surges in BLDC drives.

Table 3. summarizes the main transient indicators extracted from simulation

Metric	PID	ANN
Rise time (10–90%) [ms]	2.10	0.597
Settling time ($\pm 2\%$) [ms]	3.70	1.10
Speed overshoot [%]	0.0021	~0
T_{ss} [N.m]	50.01	49.995
T_{peak} [N.m]	92.361	50.000
Torque overshoot [%]	84.687	0.00893

Overall, the ANN controller improves speed transient speed ($\approx 3.5\times$ faster rise and $\approx 3.4\times$ faster settling) and nearly eliminates torque overshoot, while preserving steady-state tracking.

7. multi-speed validation (500, 750, and 1000 rpm)

Table 4 presents the transient speed parameters (rise time, settling time and speed overshoot) and torque metrics (steady-state torque, peak torque and torque overshoot) of both controllers at three reference speeds (500; 750; 1000 rpm), with a constant load torque applied (50 N.m). Operating at all of these points, the ANN controller achieves significantly better stability in terms of minimizing rise time as well as settling time while having an almost negligible speed overshoot for each value compared to the tuned PID baseline. More importantly, the ANN significantly reduces transient electromagnetic torque spikes with respect to the PID controller: it drives down torque overshoot from ~ 84 – 113% (PID) to near-zero values (ANN), maintaining almost identical steady-state torque level of about 50 N.m. The results show that the ANN-based drop-in replacement keeps its benefits over a wider speed range so it is not limited to any particular operating condition.

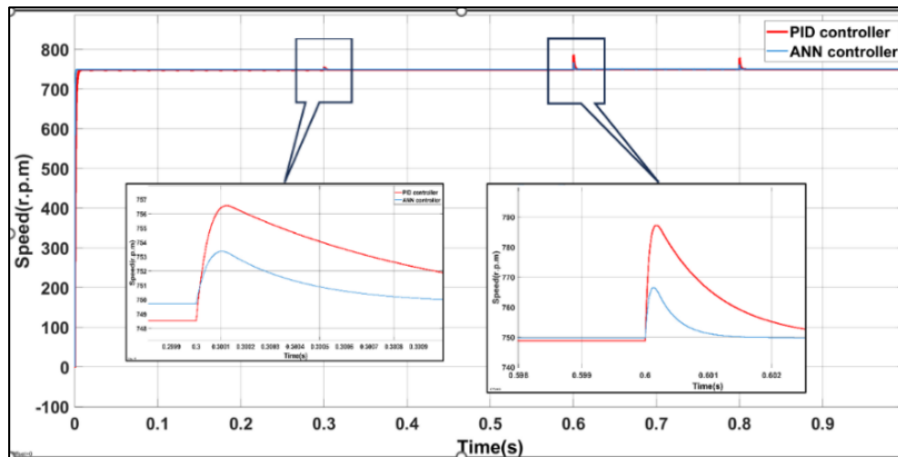
Table 4. multi-speed performance summary (500, 750, 1000 rpm; load torque = 50 N·m)

Speed (rpm)	Controller	Rise time (s)	Settling time (s)	Speed OS (%)	Tss (N·m)	Tpeak (N·m)	Torque OS (%)
500	ANN	5.9986e-04	1.1392e-03	2.4456e-06	50.01	50.011	0.0029319
500	PID	2.0898e-03	3.7505e-03	5.1181e-06	50.005	78.311	56.606
750	ANN	5.9666e-04	1.1000e-03	~0	49.995	50.000	0.0089289
750	PID	2.1000e-03	3.7000e-03	0.0021	50.01	92.361	84.687
1000	ANN	0.00059521	0.001119	0.0019245	50.02	50.022	0.0033467
1000	PID	0.0020746	0.0037189	0.010509	50.01	106.38	112.72

Table 4 summarizes the comparative dynamic performance of the ANN-based controller and the tuned PID controller at three reference speeds (500, 750, and 1000 rpm). The ANN controller consistently achieves faster rise and settling times and significantly lower torque overshoot, while maintaining negligible speed overshoot across all tested operating point.

8. Speed regulation under step-load disturbances

The speed response under the step-load disturbance test is shown in Fig. 8. The reference speed is maintained at approximately 750 rpm while the mechanical load torque is changed in multiple steps. At each load transition (e.g., around $t \approx 0.6$ s and $t \approx 0.8$ s), the tuned PID controller (red) exhibits larger transient speed excursions and a slower recovery. In contrast, the ANN controller (blue) produces smaller speed deviations and returns to the reference more rapidly, indicating improved disturbance rejection and tighter speed regulation under variable loading.

**Figure 8.** Rotor speed response under the step-load disturbance test (PID vs ANN), reference speed $\approx$ 750 rpm

9. Electromagnetic torque response under step-load disturbances

Fig. 9. plots the corresponding electromagnetic torque response for an identical step-load profile. Both tuned PID controller generates excessive torque transients which forms high start-up torque spiking and large torque excursions at the load-step instants. After each load change, the new steady-state torque levels are reached, whereby the ANN controller shows noticeably smoother torque trajectories with lower transient peaks. This decrease in torque peaking corresponds to the improved speed regulation found in figure 9 and is useful for decreasing mechanical stress and limiting current spikes for BLDC drives influenced by time-varying loads.

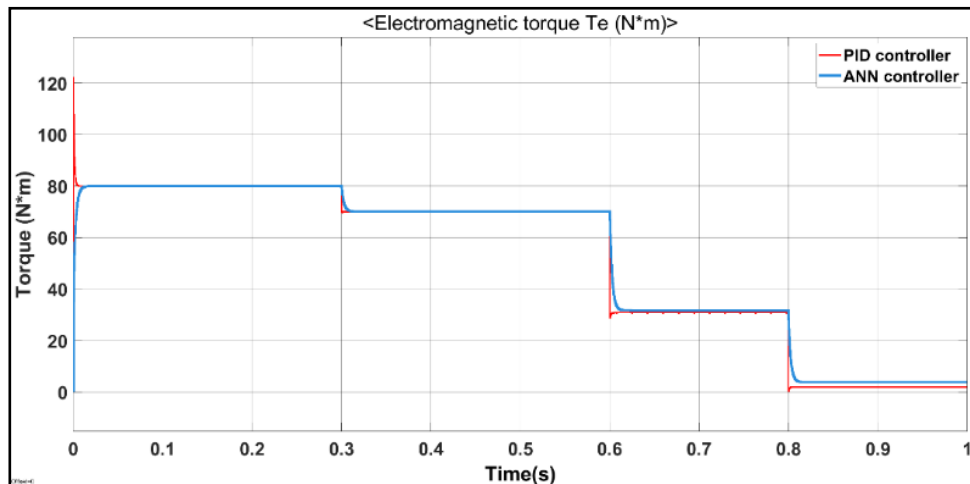


Figure 9. Electromagnetic torque response under the step-load disturbance test

10. Conclusion

An implementation using a Hall-sensor BLDC drive with six-step commutation was modeled in Simulink to compare the tuned discrete-time PID speed controller with a feedforward (FF) ANN deployed in a drop-in replacement manner. The ANN was supervised trained and tested using the same inverter, commutation, and motor setup as used for searching the gains based upon PID Tuner. Throughout the rated case and additional speeds (500/750/1000 rpm at 50 N·m), the ANN produced both faster speed transients, and significantly reduced to near-zero overshoot of heavy electromagnetic torque peaks present with PID while maintaining the necessary steady state torque. The ANN showed lower speed deviations and smoother torque transitions during step-load disturbances, suggesting better disturbance rejection capability and less mechanical stresses. Results show that replacing a tuned PID block with a small ANN improves transient performance of commutated BLDC drives, while avoiding changes in power stage or commutation logic. Thus, the tuned PID controller acted only as a practical source for initial training data, and did not contribute at all to the final control action during ANN tests.

A limitation of this study is that the proposed controller was validated using MATLAB/Simulink simulations rather than a physical testbed for brushless DC motors. Consequently, practical effects—such as sensing, computational delay, inverter dead-time, parameter uncertainty, and physical implementation—have not yet been experimentally evaluated. Future work will focus on Processor-in-the-Loop (PIL) and Hardware-in-the-Loop (HIL) implementations, followed by experimental validation using a testbed for Hall-switched brushless DC motors. These experiments will be used to verify whether the transient performance improvements observed in simulation are maintained under practical operating conditions and to determine the real-time execution response time of the artificial neural network controller.

References

- [1] M. Şen and M. Mutluer, "A review of BLDC motors: Types, application, failure modes and detection," *Energies*, vol. 18, no. 24, Art. no. 6402, 2025, doi: 10.3390/en18246402.
- [2] N. Prabhu, R. Thirumalaivasan, and B. Ashok, "Critical review on torque ripple sources and mitigation control strategies of BLDC motors in electric vehicle applications," *IEEE Access*, vol. 11, pp. 115699–115739, 2023, doi: 10.1109/ACCESS.2023.3324419.
- [3] F. S. Ahmed, O. G. Ibrahim, and M. N. Allawi, "PV-integrated wireless power transfer system for high-efficiency electric vehicle charging applications," *Diagnostyka*, vol. 27, no. 1, pp. 1–8, 2026, doi: 10.29354/diag/218930.
- [4] J. Shi, Q. Mi, W. Cao, and L. Zhou, "Optimizing BLDC motor drive performance using particle swarm algorithm-tuned fuzzy logic controller," *SN Appl. Sci.*, vol. 4, Art. no. 293, 2022, doi: 10.1007/s42452-022-05179-6.
- [5] A. Venkatesh, S. Nalinakshan, S. Phani Kumar, S. S. Kiran, and K. Akash, "Optimized control strategies for BLDC motor in sustainable energy systems: Enhancing stability and performance through intelligent controllers, filtering techniques and internal model control," *Global Energy Interconnection*, early access, Dec. 31, 2025, doi: 10.1016/j.gloi.2025.09.004.
- [6] K. Kroičs and A. Būmanis, "BLDC motor speed control with digital adaptive PID-fuzzy controller and reduced harmonic content," *Energies*, vol. 17, no. 6, Art. no. 1311, 2024, doi: 10.3390/en17061311.

- [7] H. Yin, W. Yi, J. Wu, K. Wang, and J. Guan, "Adaptive fuzzy neural network PID algorithm for BLDCM speed control system," *Mathematics*, vol. 10, no. 1, Art. no. 118, 2022, doi: 10.3390/math10010118.
- [8] M. H. Mahmood, O. F. Ali, M. J. Mnati, B. M. Sabbar, and A. Van Den Bossche, "Intelligent PID parameter tuning for BLDC motors by using genetic algorithms," *J. Européen des Systèmes Automatisés*, vol. 58, no. 7, pp. 1463–1476, 2025, doi: 10.18280/jesa.580714.
- [9] A. A. Hadi, "The Impact of Artificial Neural Network (ANN) on the Solar Energy Cells: A Review," *International Journal of Computational and Electronic Aspects in Engineering*, vol. 5, no. 1, pp. 30–41, 2024, doi: 10.26706/ijceae.5.1.20240505.
- [10] H. A. Mutar, I. J. Mohammed, and I. R. N. ALRubeei, "AI for Solar PV Forecasting, MPPT, and Energy Management: A Comprehensive Review," *International Journal of Computational and Electronic Aspects in Engineering*, vol. 7, no. 1, pp. 100–114, 2026, doi: 10.26706/ijceae.7.1.20260111.
- [11] F. H. Faris, "Physics-Informed Neural Network Framework for Dynamic Thermal Modeling of Power Transformers Under Extreme Climate Conditions," *International Journal of Computational and Electronic Aspects in Engineering*, vol. 7, no. 2, pp. 244–254, 2026, doi: 10.26706/ijceae.7.2.20260313.
- [12] E. H. Al-Zaidi, "Supervised learning for adaptive BLDC motor control: Integrating classical PID with neural networks," *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics*, vol. 37, pp. 180–193, 2025, doi: 10.55549/epstem.1195.
- [13] S. Ghamari and A. Aziz, "A novel deep transfer learning-based adaptive cascade PI controller enhanced by reinforcement learning algorithm and snake optimization for robust speed regulation of brushless DC motors," *IET Power Electronics*, vol. 19, no. 1, Art. no. e70157, 2026, doi: 10.1049/pel2.70157.
- [14] A. Hasni, H. El Fadil, A. Lassoui, M. El Ancary, Y. El Asri, A. Hamed, and S. EL Jeilani, "Optimized speed regulation of BLDC motors: A comparative performance study of artificial neural networks and super-twisting sliding mode controllers," *Int. J. Robot. Control Syst.*, vol. 5, no. 4, pp. 2360–2378, 2025, doi: 10.31763/ijrcs.v5i4.2046.
- [15] M. Megrini, A. Gaga, and Y. Mehdaoui, "Processor in the loop implementation of artificial neural network controller for BLDC motor speed control," *Results Eng.*, vol. 23, p. 102422, Sep. 2024, doi: 10.1016/j.rineng.2024.102422.
- [16] Chandrarupa and M. Silas, "A Concise Review on Modelling and Performance Characteristics of Multistage Buck Converters for Deep Step-Down Applications," *International Journal of Computational and Electronic Aspects in Engineering*, vol. 7, no. 2, pp. 115–121, 2026, doi: 10.26706/ijceae.7.1.20260107.
- [17] M. Kadhim, "Improving Power Quality with Active Filtering and Energy Storage Technologies in Grid-Connected Hybrid Solar-Wind Renewable Energy Systems," *International Journal of Computational and Electronic Aspects in Engineering*, vol. 7, no. 2, pp. 291–304, 2026, doi: 10.26706/ijceae.7.2.20260317.
- [18] R. Shenbagalakshmi, S. K. Mittal, J. Subramanian, V. Vengatesan, D. Manikandan, and K. Ramaswamy, "Adaptive speed control of BLDC motors for enhanced electric vehicle performance using fuzzy logic," *Sci. Rep.*, vol. 15, Art. no. 12579, 2025, doi: 10.1038/s41598-025-90957-6.
- [19] J. Ouakrim, A. Bodian, D. Ouadani, and A. Cardenas, "FPGA implementation and performance evaluation of classic PID, IMC and DTC for BLDC motor control," *Vehicles*, vol. 8, no. 2, Art. no. 42, 2026, doi: 10.3390/vehicles8020042.
- [20] V. Busher, V. Kuznetsov, V. Kovalenko, M. Babyak, V. Druzhinin, V. Tytiuk, A. Rojek, K. Klochko, I. Gurin, and Y. Shramko, "Fractional PI-PI μ D controllers with neural network adaptation in control system of BLDC motor electric drives," *Energies*, vol. 18, no. 23, Art. no. 6132, 2025, doi: 10.3390/en18236132.
- [21] G. Yao, J. Feng, G. Wang, and S. Han, "BLDC motors sensorless control based on MLP topology neural network," *Energies*, vol. 16, no. 10, Art. no. 4027, 2023, doi: 10.3390/en16104027.
- [22] N. Prabhu, T. Rajaram, and B. Ashok, "Design of sliding mode model predictive dual-loop control through self-learning strategy to mitigate the torque ripple in BLDC motor for electric vehicles," *Scientific Reports*, vol. 15, Art. no. 45617, 2025, doi: 10.1038/s41598-025-29445-w.
- [23] R. Inan, "An improved model predictive current control of BLDC motor with a novel adaptive extended Kalman filter-based back EMF estimator and a new commutation duration approach for electrical vehicle," *Int. J. Circuit Theory Appl.*, vol. 53, no. 2, pp. 1135–1150, 2025, doi: 10.1002/cta.4407.
- [24] S. T. Nguyen, T. M. Pham, A. Hoang, and T. T. Cao, "Optimal tuning of digital PID controllers for commercial BLDC motors using the Nelder–Mead method," *Power Electron. Drives*, vol. 10, no. 1, pp. 210–226, 2025, doi: 10.2478/pead-2025-0015.
- [25] M. A. Taleb and G. Husi, "Dynamic modeling of a three-phase BLDC motor using bond graph methodology," *Actuators*, vol. 14, no. 11, Art. no. 523, 2025, doi: 10.3390/act14110523.

- [26] R. Ristiana, S. Kaleg, A. C. Budiman, S. Sudirja, A. Hapid, A. Muharam, and Amin, "Novel hybrid-MPCC approach to generate switching sequence for six-step commutation of BLDC motor using Hall effect sensor," *Alexandria Eng. J.*, vol. 82, pp. 43–54, 2023, doi: 10.1016/j.aej.2023.09.052.
- [27] Y. I. Son, H. Cho, and J. Kang, "Design and experimental validation of an ISMC-based position controller with supervisory RBF neural network and PIO for BLDC motor systems," *Electronics*, vol. 15, no. 15, Art. no. 3373, 2026, doi: 10.3390/electronics15153373.
- [28] S. Solgi, A. Stadler, K. Pourhossein, A. Jahic, M. Plenz, and D. Schulz, "Modeling, simulation, and performance evaluation of a commercial electric scooter," *World Electr. Veh. J.*, vol. 16, no. 9, Art. no. 529, 2025, doi: 10.3390/wevj16090529.