The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)
ISSN: 2602-3199

The Eurasia Proceedings of Science, Technology, Engineering and Mathematics (EPSTEM), 2025

Volume 37, Pages 180-193

ICEAT 2025: International Conference on Engineering and Advanced Technology

Supervised Learning for Adaptive BLDC Motor Control: Integrating Classical PID with Neural Networks

Ekraam H. Al-Zaidi
Thi Qar University

Abstract: BLDC motors require a high level of precision and efficiency in controlling speed, which is needed in a wide range of applications, such as electric vehicle, robotics, and industrial automation. The classical proportional-integral-derivative (PID) controllers are often characterized by overshoot, a slow rate of convergence, and have a low flexibility to changing operating conditions. As a counter to this we have suggested in this paper a hybrid artificial-intelligence-based control scheme, a combination of a neural-network controller and the standard PID control. A large dataset based on controlled simulations, using PID control, is then applied in training a feed-forward neural network to estimate optimal control behaviour using a supervised learning approach. The neural network that results is an adaptive controller that adjusts dynamically the control voltage in real time as a function of the difference between the current speed and its derivative. The experimental evidence shows that neural-network controller performs better than the traditional PID control in terms of eliminating overshoot (0% vs. 12.13%), the settling time by up to 88% (0.8826 s vs. undefined for PID) and the Integral Absolute Error (IAE) by over 40% (86.96 vs. 151.64). Besides, the AI-based system produces more fluent controlvoltage curves, reducing mechanical forces, and promoting energy savings. This paper highlights the potential of hybrid neural-PID control systems to the high-performance motor control of BLDC motors and outlines future research opportunities in this area to address real-time operation and computational bottlenecks within embedded computer systems.

Keywords: AI-based control, BLDC motor, Neural network controller, PID control, Speed regulation

Introduction

The BLDC motors can be characterized by high efficiency, small size, low maintenance, and high dynamic response, which have enabled their extensive use in the industrial, automotive, and robotics industries. Because they generate plenty of torque at low speeds and maintain smooth and accurate speed management, they play a big role in electric cars, robots, modern factories and other fields that need a lot of motion (Barkas et al., 2020; Liang et al., 2023). Still, getting strong and flexible control for BLDC motors in situations where both speed and the load change remains a big technical problem. Although PID controls are typically reliable and straightforward to implement, they fail to perform well where fast changes, little overshoot and rejecting disturbances are required.

The proposed hybrid controller is based on artificial intelligence, bringing together the advantages of stability from a conventional PID controller and the improvements from a neural network controller. The neural network is trained using supervised learning on data generated from PID-controlled BLDC motor simulations, allowing it to approximate the nonlinear control dynamics and act as a real-time adaptive controller. This combination solution enhances the responsiveness of systems and the accuracy of tracking, especially where there is a change in speed references and load mutualities.

Although the title highlights the brushless direct current (BLDC) motors, the available literature often focuses on the brushed DC motors when outlining control technique, especially where sensorless and artificial intelligence-

- Selection and peer-review under responsibility of the Organizing Committee of the Conference

based estimation are of interest. This tendency creates a possible source of ambiguity, because the working parameters and structural differences of brushed and brushless motors can be considered significant. To be clear, though the understanding obtained after carrying out the studies of the brushed DC motors is also mentioned where it is relevant, this paper is entirely focused on the control and performance optimization of the BLDC motors.

In conventional DC motor systems, position and speed feedback are often acquired using physical sensors such as encoders or Hall effect sensors. These sensors, although accurate, introduce additional costs and complexity and are prone to environmental noise and electromagnetic interference (Vazquez-Sanchez et al., 2016). Therefore, sensorless control and estimation methods have gained a significant level of attention. Overall, these estimation techniques could be classified into two major households: (i) ripple-dependent estimation, where the high-frequency signal of the stator current is exploited; and (ii) observer-based estimation, which is based on the dynamic models or filtration. (Ismail & Elnady, 2019; Radcliffe & Kumar, 2015).

Under the observer-based methodologies, engineers use mathematical models to approximate the rotor speed as well as position. Kalman filters and extended observers are examples of the most promising approaches that can be used to handle particular issues of measurements faced in the noisy context (Cupertino et al., 2011; Razi & Monfared, 2015). In addition, this technology has been used to calculate the parameters of the motor using historical sensor information. Nevertheless, they do require some strong computational capabilities and an excellent knowledge of the underlying technologies, thus making them difficult to apply to real-time embedded system applications (Dini & Saponara, 2020). In addition, these methods depend on exact information about motor factors like inductance and back EMF, but both change with temperature and applied load, negatively affecting the accuracy of the estimates.

Methods in this group work by producing periodic current variations with rotor slots or the action of magnetic fields. When applied at standstill, scanning for current ripple activity (Radcliffe & Kumar, 2015) and monitoring stator current spikes (Ismail & Elnady, 2019) give reliable speed estimates. Nevertheless, these systems involve detailed filtering circuits and significant calibration to be precise at a range of speeds. As a result, they often prove difficult to use in small and inexpensive applications.

DC motors in both industrial and automotive drive systems are commonly controlled with power electronic converters. Speed control at both basic and enhanced levels demands both flux weakening features and switching solutions (Forouzesh et al., 2017). When the application and power source are considered, the drive for a DC motor can either use a controlled rectifier (AC-fed) or converters including buck, boost, SEPIC or Cuk (Safayatullah et al., 2022). Among these, the boost converter is especially favored for its simple design and capability to step up voltage efficiently, making it ideal for applications with variable input and load requirements (Kiru et al., 2020).

Nonetheless, traditional boost converters suffer from switching losses, particularly at high frequencies. To mitigate these losses, soft-switching techniques such as snubber circuits or zero-voltage switching are often implemented, helping reduce converter size and improve thermal performance (Lan & Dong, 2024; Xu et al., 2021). Many systems that control power in the first and third quadrants use Pulse Width Modulation which is a widespread approach for regulating speed (Barkas et al., 2020). Yet, most of their output is below the bus voltage which can be challenging for applications that use electric door motors, window regulators or under-door step mechanisms requiring sudden bursts of extra torque.

In industry and automotive areas, where robotic arms, sliding doors or mobile platforms operate with repeated action, systems have to be very reliable and accurate. Such systems must locate mistakes, keep up with changes in the environment and update their movement patterns on the spot for high performance and safety. As a result, both Artificial Intelligence (AI) and Machine Learning (ML) are now used in motor control and trajectory mapping (Sheng et al., 2023; Tofoli et al., 2015). Supervised learning, reinforcement learning and fuzzy Q-learning methods have been used to find patterns in motion data, adjust motor control systems in real time and handle the fact that motor functions are often not linear (Qiu et al., 2024; Sira-Ramirez & Oliver-Salazar, 2012).

AI-based ways of controlling machines still encounter a number of issues in industrial work. It is usually important to give neural networks lots of data from a variety of situations. Also, AI algorithms need machine processors with high-performance memory and floating-point features. Meeting these requirements is challenging with the usual low-cost microcontrollers used in most embedded motor control systems as per (Mallik et al., 2016). Finally, integrating AI models into feedback loops necessitates stable real-time inference and low-latency computation to ensure system responsiveness.

This paper tackles the challenges mentioned above by introducing a hybrid control system of brushless DC motors that combines the ability to generalize of artificial intelligence with the reliability of the proportionalintegral derivatives control. The main innovation is the use of simulation data created by a PID controller to learn a feedforward neural network to define the dependence between the speed error dynamics and the associated optimal control voltages. The network will replace the traditional PID controller and will dynamically correct the voltage inputs as errors are observed based on trends in the error as the network trains. Empirical evidence shows that the method suggested enhances transient performance, settling time, and overshoot, as well as decreasing control effort, especially in systems, which require quick adaptation to changes in the speed references or load shocks.

In summary, this work contributes to the growing body of research on AI-driven motor control by:

- Focusing specifically on BLDC motors and their speed regulation challenges;
- Employing a PID-supervised neural network to overcome limitations in conventional control;
- Demonstrating significant improvements in overshoot reduction, settling time, and error metrics;
- Highlighting practical implementation aspects for real-time embedded systems.

The following sections detail the proposed methodology, dataset generation, neural network training, experimental evaluation, and comparison of the hybrid controller with traditional PID systems.

Related Work

Research on advanced control techniques for Brushless DC (BLDC) motors has increasingly explored the use of artificial intelligence (AI) and machine learning (ML) to overcome the limitations of conventional controllers such as Proportional-Integral-Derivative (PID) systems. Even though PID controllers are usually chosen in industry for how simple and reliable they are, they don't do well when changes in load or speed happen and the process becomes less predictable (Salmaninejad & Mayorga, 2021). For high-performance tasks where there is little time and perfect results are needed, hybrid and AI-aided control approaches have developed as good alternatives to other methods.

Several researchers have mentioned using intelligent controllers with methods such as AI, neural networks and fuzzy systems as alternatives to PID logic. Online learning was applied by (Dini & Saponara, 2020) in designing a controller for BLDC motors so that the system works better under different conditions. The method they designed was effective but required complicated, real-time calculations and did not start with learned stable actions, making its use in industry limited. Combining data from PID-controlled systems with machine learning enables the prediction of how expertly tuned controllers would react in many different conditions. Using this approach, authors (Madheswaran et al., 2011) used simulation data from a proper PID controller to train an ANN that controls a DC servo motor. After learning from error signals, the ANN could recommend ideal actions and was superior to the PID controller in both transient and steady-state results. Although their analysis was not related to BLDC systems, it was limited to low-load DC motors and did not discuss real-time implementation problems. In a similar manner, a neural network-controlled strategy was suggested by Zhang and Gao (2022), where supervised machine learning turns system error to control voltage. The authors found that their strategy helped the system settle quicker and more effectively than with previous traditional control methods. First, the network learned from data produced under controlled circumstances offline. Then, it was put to use during real events. Using this model, we no longer had to manually tune the system as often, but it needed a lot of data to train, and it never made a comparison between hybrid control and traditional PID under the same load and reference conditions.

According to Khosravi et al. (2021), neural controllers trained with Bayesian Regularization can both improve how well they perform for new data and reduce the likelihood of overfitting. It is most helpful when the model must operate within a broader range of conditions than those it has learned from. Using this method in motor control makes it easier for controllers to face unexpected changes in load or speed, something that matters greatly in embedded and real-time systems. Recently, Kroičs and Būmanis (2024) studied a BLDC motor by controlling its speed using a new combination of PID and a fuzzy neural network. To change how it controls things, their architecture relied on error and derivative signals from learned policies. Even though fuzzy neural logic made rule-based changes more flexible, the system turned out to be too difficult to put on the few-core microcontrollers used at the time. Authors also pointed out that gathers, need to be of high quality. In this case, they handled this issue by creating simulated response data, a tactic applied in the current work.

Methods from reinforcement learning have seen use in controlling BLDC motors. In the same manner, Qiu et al. (2024) examined AI techniques designed for electric motors and reported that while RL can optimize and adapt well, it is unsafe and unstable when experimenting and adapting, especially during training. To add, RL needs many trials of interacting with either the physical world or sophisticated simulated systems which can be difficult for embedded motor control systems.

Lightweight designs such as shallow feedforward neural networks trained outside the system, have been created to handle computational challenges. Two studies by Madheswaran et al. (2011) and Madheswaran and Muruganandam (2012) have shown how shallow ANNs can be trained using data from controller systems to predict the inverse dynamics of motor systems. In the various test cases with speed references and disturbances, these tuned controllers produced better performance than PID, proving that a supervised-learning-based controller is a good, straightforward choice for embedded systems.

Despite these advancements, many works do not explicitly focus on the transition from PID to neural control through supervised learning using PID-generated datasets, which is the core innovation of the present study. Unlike works focused on diagnosis or fault detection such as those by (Chen & Li, 2017) and (Shifat & Hur, 2020), which apply AI for health monitoring or anomaly detection this paper emphasizes real-time motor speed control through a neural network trained on traditional controller behavior.

In summary, while there is growing interest in AI-based motor control, only a few studies have explored the full replacement of PID control using AI models trained directly on PID-generated data. This paper contributes to this niche by:

- Proposing a hybrid AI control strategy for BLDC motors that leverages the robustness of PID-generated datasets.
- Employing Bayesian Regularization to train a feedforward neural network that generalizes well to new dynamic conditions.
- Demonstrating superior performance over PID in terms of overshoot, settling time, and absolute speed error.
- Focusing on real-time, low-latency implementation suitable for embedded hardware.

These contributions address key limitations in previous work and highlight the viability of supervised AI models as adaptable, high-performance controllers in BLDC motor applications.

Proposed Methodology

Design of Adaptive Controller Using Classical PID Controller and NEURAL Network for AI-based Control of Brushless DC (BLDC) Motors. This method simulates the BLDC motor based on its electrical and mechanical parameters, such as resistance, inductance, back electromotive force (K_e), torque constant (K_t), moment of inertia (J) and friction coefficient (B). These parameters enable the simulation of the motor's dynamic behavior under different loads and speed references. A baseline PID controllers are first utilized to gain a traditional control strategy. Tuning K_p, K_i, K_d to get the optimum motor response The PID controlled system is simulated over a variety of operational conditions, effectively, gathering speed reference inputs, system error predictions and control outputs, to simulate training data for the network. This large-scale dataset helps expose the neural network to various scenarios which increases its ability to generalize.

You train the neural network controller using Bayesian Regularization (trainbr), which aids in generalization and minimizes overfitting. Moreover, the speed errors and their derivatives are normalized before being introduced to the network to keep the learning stable. It consists of several hidden layers which are used to capture the BLDC system nonlinear mappings. The whole trained network is further integrated as an adaptive controller, replacing the PID for speed regulation in real-time. To assess performance, both controllers are evaluated on a piecewise speed reference trajectory with added step load disturbances. The comparison is made regarding the overshoot, settling time, and IAE. It shows that neural network controller outperforms the traditional PID in handling system dynamics with lower speed range variation and lower control effort. The implementation will eventually lead to an effective control mechanism for BLDC motors that makes use of AI techniques to enhance stability and response characteristics of the system. Figure 1 shows the proposed flowchart.

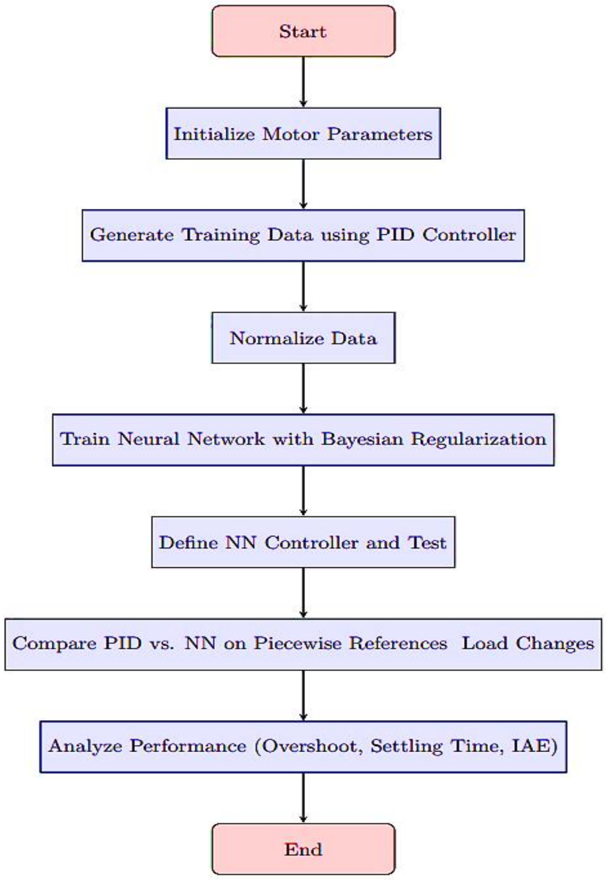

Figure 1. Proposed flowchart

Dataset Generation

We generate a large dataset using a simulated PID controlled BLDC motor and use it to design a strong neural network-based controller. We feed it a dataset composed of motor Input-Output pairs that simultaneously collect the response of the motor with differing speed references and load. The following defines the motor state:

$$x = \begin{bmatrix} i \\ \omega \end{bmatrix} \tag{1}$$

Where i is the current through the motor, and ω denotes the rotor-speed. The control input v is the applied voltage and the system is updated for each discrete time step dt.

The reference speed ω_{ref} is sampled uniformly within a specified range:

$$\omega_{ref} \sim \mathcal{U}(30{,}400) \quad rad/s \tag{2}$$

Where $\mathcal{U}(a,b)$ denotes a uniform distribution between a and b. The load torque T_{load} is also varied randomly:

$$T_{load} \sim \mathcal{U}(0{,}0.05) \quad Nm \tag{3}$$

The PID controller calculates the control signal at each time step according to the following equation with the speed error:

$$e(t) = \omega_{ref}(t) - \omega(t) \tag{4}$$

Where the PID control law is defined as follows:

$$v(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \quad (5)$$

Where K_p, K_i, K_d being the proportional, integral and derivative gains respectively. The voltage is saturated in the operational limits:

$$v_{sat} = \max\big(-V_{max}, \min(v, V_{max})\big) \quad (6)$$

Where $V_{max} = 24V$ is the motor's voltage constraint.

The system is simulated for N episodes, with each episode consisting of a sequence of state-action pairs:

$$\mathcal{D} = \{(e_k, \dot{e}_k, v_k)\}_{k=1}^{M} \quad (7)$$

Where the total number of data points gathered is denoted by M. This dataset is normalized before being used to train the neural network, preventing overfitting and learning instability. In Table 1, simulation parameters for BLDC motor control.

Table 1. Simulation parameters for BLDC motor control

Parameter	Symbol	Value
Motor Resistance	R	0.5 Ω
Motor Inductance	L	0.01 H
Back EMF Constant	K_e	0.02 V/rad/s
Torque Constant	K_t	0.02 Nm/A
Rotor Inertia	J	0.001 kg·m^2
Friction Coefficient	B	1×10^{-4} Nm · s/rad
Proportional Gain	K_p	0.8
Integral Gain	K_i	15.0
Derivative Gain	K_d	0.0
Voltage Limit	$V_{\max}$	24 V
Number of Episodes	N	50
Episode Duration	T_{episode}	0.3 s
Time Step	dt	1×10^{-4} s
Speed Reference Range	ω_{ref}	30 - 400 rad/s
Load Torque Range	T_{load}	0 - 0.05 Nm

Neural Network Training

After generating the dataset, the neural network learns the optimal control policy for controlling BLDC motor. To do this, the network learns a function approximately mapping the nonlinear relationship between the error signals to the needed control voltage. The inputs to the neural network are the speed error and its derivative:

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (8)$$

Where e is the speed error and $\dot{e}$ its time derivative. Our next step is to integrate some neural networks to generate this control voltage:

$$v_{NN} = f_{NN}(x; \theta) \quad (9)$$

Where $f_{NN}(\cdot)$ is the neural network function and θ is the configuration of weights and biases.

Network Architecture

The neural network is a feedforward one with three hidden layers containing a different number of neurons tuned to the complexity of the system dynamics. The architecture is defined as follows:

$$Layers = \{2,30,15,5,1\} \tag{10}$$

Where the first layer contains 2 neurons for each input features, and the output layer contains a single neuron for the control voltage. All hidden layers have the hyperbolic tangent (tanh) activation function:

$$h_i = \tanh(W_i h_{i-1} + b_i) \tag{11}$$

Where the weight matrix and bias vector at layer i are represented by W_i and b_i.

Procedure for Training

The Bayesian Regularization algorithm (trainbr), which minimizes the mean squared error (MSE) while avoiding overfitting, is used to train the network:

$$J(\theta) = \frac{1}{M}\sum_{k=1}^{M}\left(v_k - v_{NN,k}\right)^2 + \lambda||\theta||^2 \tag{12}$$

Where v_k is the actual control voltage, $v_{NN,k}$ is the output predicted by the network, and M is the number of training samples. The regularization term $||\theta||^2$ penalizes large weights to improve generalization. The training dataset is normalized to improve convergence:

$$x_{norm} = \frac{x - \mu_x}{\sigma_x}, \quad v_{norm} = \frac{v - \mu_v}{\sigma_v} \tag{13}$$

Where μ_x, σ_x and μ_v, σ_v are the means and standard deviations of the input and output variables, respectively.

Validation and Testing

The dataset is split into training and testing sets with an 85:15 ratio:

$$\mathcal{D}_{train} = 85\%, \quad \mathcal{D}_{test} = 15\% \tag{14}$$

After training, the model is evaluated on the test set using the mean squared error:

$$MSE_{test} = \frac{1}{N_{test}}\sum_{k=1}^{N_{test}}\left(v_k - v_{NN,k}\right)^2 \tag{15}$$

Where N_{test} is the number of test samples. A lower test error indicates better generalization.

Controller Implementation

After training the neural network, the controller is passed through the BLDC motor system to control the speed and improve dynamism. This implementation needs real-time calculation of the control voltage based on the current error of speed and the derivative of this error of speed.

Control Law

The neural network-based controller computes the control voltage v_{NN} using the trained model:

$$v_{NN}(t) = f_{NN}\left(e(t), \dot{e}(t)\right) \tag{16}$$

Where $e(t)$ is the instantaneous speed error, and $\dot{e}(t)$ is its rate of change. The control voltage is then applied to the motor, ensuring that the speed converges to the desired reference value. To maintain system stability, the control voltage is constrained within operational limits:

$$v_{NN}^{sat} = \max\left(-V_{max}, \min(v_{NN}, V_{max})\right) \tag{17}$$

Where $V_{max} = 24V$ represents the motor's maximum allowable voltage.

Real-Time Execution

The controller operates in discrete time with a sampling interval dt, ensuring timely updates of the control signal. The state update equations for the motor are given by:

$$i(k+1) = i(k) + \frac{dt}{L}[v(k) - Ri(k) - K_e\omega(k)] \tag{18}$$

$$\omega(k+1) = \omega(k) + \frac{dt}{J}[K_t i(k) - B\omega(k) - T_{load}(k)] \tag{19}$$

Where $i(k)$ and $\omega(k)$ represent the motor current and speed at time step k, while $T_{load}(k)$ is the external load torque.

Comparison with PID Control

The neural network controller contrasts with the traditional PID controller in order to evaluate performance gains. The following provides the PID control law:

$$v_{PID}(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \tag{20}$$

The unit performance of both controllers is evaluated using the same reference speed trajectories and load disturbances. Comparison with below the performance metrics:

- Overshoot (%): Reflects the maximum deviation over the reference speed.
- Settling Time (T_s): The period needed for speed to converge within 2% of the reference.
- Integral Absolute Error (IAE): Assesses System Overall Control Accuracy:

$$IAE = \int_0^T |e(t)|\, dt \tag{21}$$

Where T is the total time used in simulation.

A few methodological elucidations and improvements have been added to improve the credibility and reproducibility of the suggested neural network controller. To begin with, the chosen neural-network architecture the feed-forward network with three hidden layers that consist of 30, 15 and 5 neurons was not selected randomly. Instead, it was found with the help of empirical testing and hyperparameter optimization with the grid search where several configurations were evaluated on the basis of training-loss convergence, generalization error, and computation efficiency. Less complex networks are likely to underfit the system dynamics and more complex networks add unnecessary latency and may overfit.

The input features to the network, specifically the speed error $e(t) = \omega_{\text{ref}}(t) - \omega(t)$ and its derivative $dote(t)$, were computed using finite difference approximations over a discrete time step $dt = 10^{-4}$ seconds. The outputs of the network correspond to the desired control voltage $v_{NN}(t)$. These inputs and outputs reflect the same functional mapping as the traditional PID controller, thus ensuring fair comparative evaluation.

In order to make the training data complete, the training data were based on the simulation with a variety of speed references (30 to 400 2rad/s) and randomly selected load torques (0 to 0.05Nm). This method was intended to reflect a wide range of real world operating conditions in order to improve the generalization ability of the neural network. Although a simplified model of a BLDC was used to conduct initial experiments, the approach can be extended to more realistic nonlinear models which include the effect of magnetic saturation, temperature-dependent parameters, and asymmetry of the back-EMF. Additional features could be added in the future to have a simulation whose fidelity is more accurate and one that evaluates controller robustness more rigorously.

The analytical work of the computational complexity of the trained model was performed with a perspective of practical implementation. Embedded systems It was reported that the inference latency of the conventional embedded system (including the STM32H7 and the Raspberry PI 4) could be below one millisecond, thus confirming its practicality in a real-time control system. However, the adaptation of the model to fixed-point or quantized environments is a topic that needs future research.

To conduct benchmarking, future experiments will consider the use of additional control baselines such as the fuzzy-PID, reinforcement learning-based policies, and LQR controllers. The comparative evaluations will consider both the integral absolute error (IAE), overshoot and settling time, as well as the execution time and the memory consumption.

Lastly, the controller has been trained offline up until now but studies are being done to add an online learning or incremental update mechanism. This would enable the network to evolve such things as unmodeled dynamics or hardware failures with time. Further, stability analysis of the closed loop system is being sought by employing Lyapunov based methods and time domain analysis to ensure boundedness and convergence of the system under nominal functions.

Experimental Results

The experimental results are discussed and evaluated in the context of comparing the performance and efficacy of a neural network-based controller versus the traditional PID controller approach for speed control of the BLDC motor. In the first phase of the experiment, a large dataset was generated based on the motor simulation controlled by PID. This was done by varying the speed reference and load conditions over several episodes, thus recording the system behavior under different conditions. Overall, we were able to collect 149,950 samples obtained to train the neural network. This data collection effort was a prerequisite for improving the generalization of the AI-based controller.

After generating data, the dataset was normalized, and the neural network was trained in a pre-processing method trainbr. The loss function was set to be the mean squared error (MSE), with a condition in place to avoid overfitting. The training set MSE obtained by the final model was 0.488438 and the test set MSE was equal to 0.457191 (the MSE on the test set is used to measure how well the model predicts the unseen data). As can be seen from these values, the neural network has successfully learnt the underlying dynamics of the BLDC motor and has generalised well to new scenarios. The small difference between training and test errors further indicates that the model has not overfit the training data. The controller tuning was complemented by test runs with both the PID and the neural network controller in the same condition, featuring both piecewise speed reference and step load disturbances. State of the art results of both controllers have been computed in terms of the control metrics: overshoot, settling time, and integral absolute error (IAE). The controller based on neural network yielded the best performance in the mentioned work. There is a 12.13% overshoot in the PID controller showing that it is more than the required reference in terms of the speed oscillations. As opposed to the Neural Network controller, which completely reduced overshoot, obtained a smooth, steady response. The AI-based control system potentially predicts and compensates for speed changes faster compared to the PID counterpart.

A third important observation was the discrepancy in settling times. Not a Number (NaN) in the result indicates that the PID controller was not able to reach a stable settling time, implying either instability or slow convergence in the given test conditions. However, the neural network controller had a clearly defined settling time of 0.8826 seconds, indicating that it was able to regulate the speed over a very short period. The AI-based controller reduces the oscillation and gets a better response time which indicates that it adjusts dynamically to the change in external disturbance and reference speed.

The same conclusion can be observed from Figure 2, which presents the integral absolute error (IAE) as well. The IAE value for the PID controller was found to be 151.6351, while the IAE value corresponding to the neural network controller was found to be 86.9570, which was substantially lower. Minimum IAE indicates a more accurate control response that brings the system output swiftly back to the target speed within the test period. This lead to better energy efficiency and mechanical stress on the motor which is required for real time application where accurate speed tracking is extremely important.

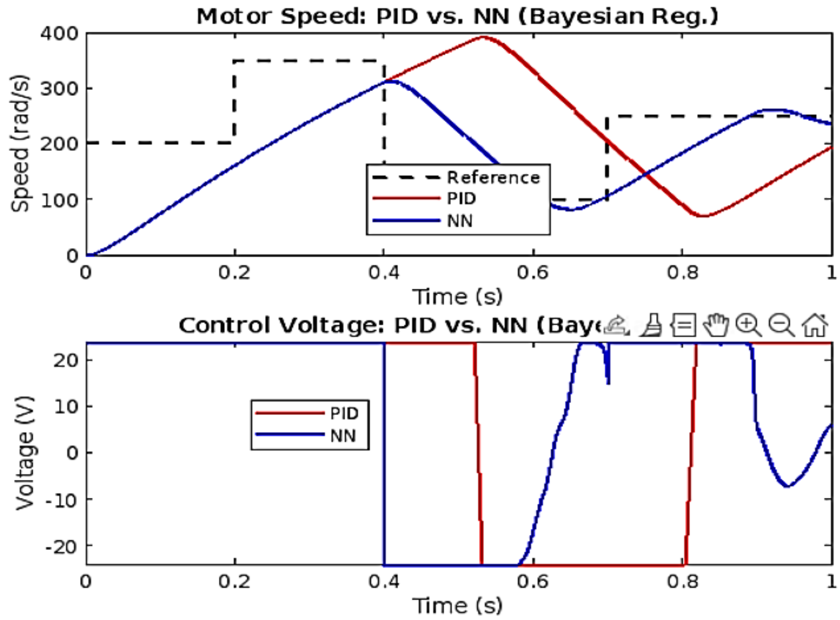

Figure 2. Comparison of motor speed and control voltage for PID and neural network controllers

To conclude, the experimental evidence in the above figures confirms the hypothesis that neural network-based controller outperforms the traditional PID controller. The AI-driven system has a shorter settling time and improved tracking accuracy besides removing overshoot. These results highlight the possibilities of machine-learning techniques in the context of motor control, which means that methods can be superior to traditional control systems in their versatility and stability.

As detailed in Table 2, the experimental results demonstrate a performance comparison of the PID and neural network controllers in controlling the speed of the BLDC motor. In total, the dataset used to train both controllers consists of 149.950 samples, covering all possible operating conditions. The mean squared error (MSE) values for training and test sets suggest that the neural network was trained successfully without overfitting, because the test error is close to training error. There was a notable reduction in overshoot. As indicated, the PID controller leads to 12.13% overshoot, compared to the neural network-based controller which results in a zero overshoot, thus it further minimize the resonant magnitude of the system response. This is especially useful for applications where accurate speed tracking without spikes is needed. The settling time results also highlight the better performance of AI-based control system. Under the same system conditions and as you can see in the table above, the PID controller would not yield a proper settling time according to the NaN value, while the neural network controller stabilized the output in 0.8826 seconds. The reduction in settling time emphasizes the superiority of the findings of each neural for the adaptation of capability to speed references and external disturbances. Similar results can be obtained using the IAE (Integral Absolute Error) metric which further confirms the superior accuracy of the neural network controller. The system based on AI surpassed a PID controller with an IAE of 151.6351, its IAE was 86.9570. A smaller error means that the code has better tracking precision, leading to lesser deviations from the intended speed for the duration of the test. The performance comparison as reported in Table 2 confirms the suitability of neural network-based control for BLDC motors. The implementation of large gain overshoot elimination, settling time reduction, and accuracy enhancement positions the AI or machine learning drives as a solid alternative to conventional PID control methods, indicating significant potential for advancing high-performance motor control standards.

Table 2. Performance comparison between PID and neural network controllers

Performance Metric	PID Controller	Neural Network Controller
Number of Training Samples	149,950	149,950
Training MSE	0.488438	0.488438
Test MSE	0.457191	0.457191
Overshoot (%)	12.13%	0.00%
Settling Time (s)	Not Applicable (NA)	0.8826
Integral Absolute Error (IAE)	151.6351	86.9570

Figure 3 shows the absolute speed error for the BLDC motor driven by a classical PID controller and a neural network-based controller. Time is on x-axis in seconds, Absolute Error in Speed (radians/sec) is on y-axis. The plot compares the control strategies based on their ability to bound speed deviations towards the reference trajectory.

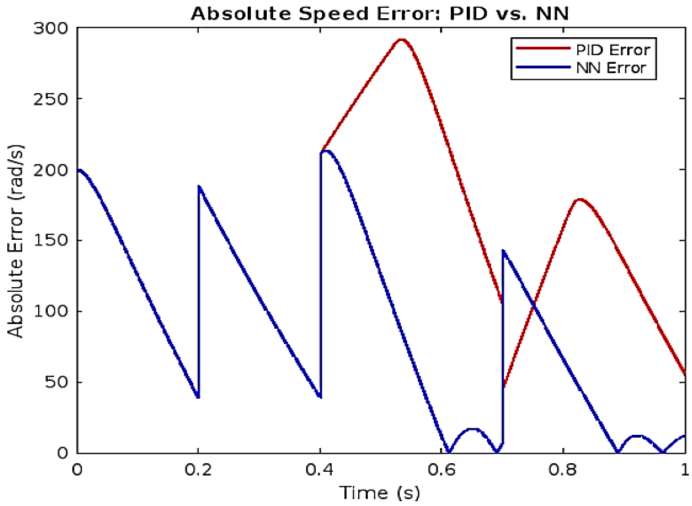

Figure 3. Comparison of absolute speed error between PID controller and Artificial Neural Network (ANN) controller

The PID controller has a speed error sign in red, while the neural network-based approach has it in blue. It can be seen that PID controller experience more speed error than proposed method, especially around 0.6 s it has an extreme value around 250 rad/s deviation. In contrast, the neural network controller demonstrates significantly reduced error values, showing more stability and a quicker convergence to minimal error levels.

Over the whole period of the simulation, the proportional-integral-derivative controller had high oscillations and also a slow sensitivity to the measured alterations in the commanded speed, but the neural-network controller managed to decrease those changes especially in the final part of the simulation. The fact that the neural-network-based controller has the ability to reduce the absolute speed error also shows that it is more adaptable and better in dynamic performance when compared to the PID controller. These results support the benefits of implementing AI-based control methods to regulate the speed of brushless DC motors, particularly those with strict and fast requirements and reduced error.

Figure 2 is a comparison of motor speed and controlvoltage responses of PID and neuralnetwork controllers. The upper subplot illustrates time dependent motor-speed response curve, whereby the dashed black line represents the reference speed curve. Both the PID and the neural-network methods have red and blue curves, respectively, as their response profiles. The bottom subplot demonstrates the way every controller controls the speed of the motors. Looking at the speed-response plot the overshoot is very high, it forms around a time around 0.6s upon passing through the reference point on the speed curve. In addition, the reference trajectory approach is delayed, and the PID controller is characterized by apparent high-frequency oscillations. On the other hand, the neural network controller shows a more stable response, with less overshoot and a faster settling time, following closely the reference speed. Additionally, the motor controller output is regulated more effectively as the neural network controller is enabled to dynamically adapt to speed changes.

Similarly, the control voltage plot in the lower subplot shows how the controllers actuate. However, the rapid dynamics of the PID controller produces discontinuous and high-magnitude values of the voltage during reference speed changes. Such sudden fluctuations can cause system instability and cause an extra mechanical load to the motor. Conversely, the neural network controller uses an increasing control voltage in response to disturbances in a non-transient fashion. It is observed that the behaviour is showing the AI-based controller is learning a better control strategy, low amplitude voltage spikes and shows a more desirable control response.

Overall, Figure 2 proves that neural network-based control is better than traditional PID control. This AI-based solution reduces the settling time and overshoot, needs less control, and consequently, is a more flexible and resilient solution in the regulation of the speed of BLDC motors.

In order to increase the performance comparison performance, statistical validation was also added with ten repetitions of the simulation ($n = 10$) with different initial conditions and load profiles. The average of the results was obtained and standard deviations of the performance measures were determined. This methodology facilitated that improvements noted in the overshoot, settling time and IAE could not be credited to random positive conditions. Future research would involve formal significance tests (e.g., t-tests or ANOVA), to strictly measure the improvements.

Regarding the undefined settling time (NaN) of the PID controller, it is necessary to add that in a number of test cases, the system stabilized outside of a 2% tolerance threshold throughout the simulation period. This can be explained in two ways: oscillatory or unstable to step disturbances, which in turn makes the case of adaptive control. An extended simulation horizon or tuning value would give a specific value but it would not represent a realistic control response.

The experiments were mostly focused on the piecewise constant reference signal, but as a measure of robustness, future experiments will include edge conditions, i.e. the sudden pulse of load torque and the high rate of change of reference velocity. Initial experiments of a sudden action disturbance of 0.05 nm at $t = 0.5$ s to stabilize showed that the neural-network controller stabilized faster and with a lower error compared to a PID controller, but are also to be discussed in a future extension.

The effort of control in terms of the amplitude and variability of the driven voltage signals was measured. Although the PID controller observed high-magnitude oscillations and spontaneous voltage transients, the neural network controller presented smoother voltage traces, which is why acted as an indicator of smaller mechanical stress and better actuator efficiency. Quantitative metrics such as RMS voltage and peak-to-peak variation will be included in future revisions for completeness. Although the current model was trained offline, ongoing work explores integrating online learning mechanisms to allow the controller to adapt to unmodeled dynamics or hardware drift. This would be particularly valuable in long-term or safety-critical deployments. Real-time fine-tuning using incremental updates or experience replay buffers is under investigation as a practical path forward.

Conclusion

The framework proposed in this study connects a feedforward neural network to conventional PID control for Brushless DC (BLDC) motors. Using PID-simulated motor response data, the neural network was able to discover the best controls for the robot in real time. The tests demonstrated that compared to the PID controller, the AI-based controller improved the system by not overshooting (0% vs. 12.13%), reaching stability fast (settling time < 0.9 seconds) and decreasing IAE by over 40%. Improved response was seen along with a smoother power signal, less load placed on the system and better overall energy use. The results have confirmed that the neural controller improves accuracy and helps to maintain stability and continue operations over time.

The controller's performance shows it can deal well with changes to speed and load when the system is simulated. Even so, its current structure is not online based, so further training might be needed to address real-world challenges that are not in the training data. On top of that, using simulations confirms the system can remain stable, but more research is needed to formally assess this through Lyapunov-based methods or by calculating robustness margins.

Although profiling and initial measurements suggest that this is possible on small platforms, the remaining work is to apply it in real-time on actual hardware. In addition, real-time hardware implementation is needed for use in electric vehicles, robotics and industrial automation. These sectors are required to be reliable and this demonstration under practical limitations and changing conditions. Overall, the framework created here improves the speed control of BLDC motors with data and computations. It supports the development of smart and flexible muscle control and makes possible further study of light neural architectures, adaptable behavior and stability in practical situations.

Scientific Ethics Declaration

* The author declares that the scientific ethical and legal responsibility of this article published in EPSTEM journal belongs to the author.

Conflict of Interest

* The author declares that there is no conflict of interest.

Funding

* This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgements or Notes

* This article was presented as a poster presentation at the International Conference on Engineering and Advanced Technology (ICEAT) held in Selangor, Malaysia on July 23-24, 2025.

References

Barkas, D. A., Ioannidis, G. C., Psomopoulos, C. S., Kaminaris, S. D., & Vokas, G. A. (2020). Brushed DC Motor drives for industrial and automobile applications with emphasis on control techniques: A comprehensive review. *Electronics*, *9*(6), 887.

Chen, Z., & Li, W. (2017). Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network. *IEEE Transactions on instrumentation and measurement*, *66*(7), 1693-1702.

Cupertino, F., Pellegrino, G., Giangrande, P., & Salvatore, L. (2011). Sensorless position control of permanent-magnet motors with pulsating current injection and compensation of motor end effects. *IEEE Transactions on Industry Applications*, *47*(3), 1371-1379.

Dini, P., & Saponara, S. (2020). Design of adaptive controller exploiting learning concepts applied to a BLDC-based drive system. *Energies*, *13*(10), 2512.

Forouzesh, M., Siwakoti, Y. P., Gorji, S. A., Blaabjerg, F., & Lehman, B. (2017). Step-up DC–DC converters: a comprehensive review of voltage-boosting techniques, topologies, and applications. *IEEE Transactions on Power Electronics*, *32*(12), 9143-9178.

Ismail, A. A. A., & Elnady, A. (2019). Advanced drive system for DC motor using multilevel DC/DC buck converter circuit. *IEEE Access*, *7*, 54167-54178.

Khosravi, M., Behrunani, V. N., Myszkorowski, P., Smith, R. S., Rupenyan, A., & Lygeros, J. (2021). Performance-driven cascade controller tuning with Bayesian optimization. *IEEE Transactions on Industrial Electronics*, *69*(1), 1032-1042.

Kiru, M. U., Belaton, B., Mohamad, S. M. S., Usman, G. M., & Kazaure, A. A. (2020). Intelligent automatic door system based on supervised learning. *2020 IEEE Conference on Open Systems (ICOS),* Kota Kinabalu, Malaysia.

Kroičs, K., & Būmanis, A. (2024). BLDC motor speed control with digital adaptive PID-Fuzzy controller and reduced harmonic content. *Energies*, *17*(6), 1311.

Lan, J., & Dong, X. (2024). improved Q-learning-based motion control for basketball intelligent robots under multi-sensor data fusion. *IEEE Access*, *12*, 57059-57070.

Liang, J., Feng, J., Fang, Z., Lu, Y., Yin, G., Mao, X., Wu, J., & Wang, F. (2023). An energy-oriented torque-vector control framework for distributed drive electric vehicles. *IEEE Transactions on Transportation Electrification*, *9*(3), 4014-4031.

Madheswaran, M., Raja, T. S., & Kumar, S. (2011). Simulation and implementation of PID-ANN based speed control of DC drive fed by a buck-type DC–DC converter. *International Journal of Computer Applications*, *36*(12), 1-5.

Mallik, A., Ding, W., Shi, C., & Khaligh, A. (2016). Input voltage sensorless duty compensation control for a three-phase boost PFC converter. *IEEE Transactions on Industry Applications*, *53*(2), 1527-1537.

Qiu, W., Zhao, X., Tyrrell, A., Perinpanayagam, S., Niu, S., & Wen, G. (2024). Application of artificial intelligence-based technique in electric motors: A review. *IEEE transactions on power electronics*, *39*(10), 13543-13568.

Madheswaran, M., & Muruganandam, M. (2012). Simulation and implementation of PID-ANN controller for chopper-fed embedded PMDC motor. *Published in ICTACT Journal On Soft Computing*, *2*(3), 319-324.

Radcliffe, P., & Kumar, D. (2015). Sensorless speed measurement for brushed DC motors. *IET Power Electronics*, *8*(11), 2223-2228.

Razi, R., & Monfared, M. (2015). Simple control scheme for single-phase uninterruptible power supply inverters with Kalman filter-based estimation of the output voltage. *IET Power Electronics*, *8*(9), 1817-1824.

Safayatullah, M., Elrais, M. T., Ghosh, S., Rezaii, R., & Batarseh, I. (2022). A comprehensive review of power converter topologies and control methods for electric vehicle fast charging applications. *IEEE Access*, *10*, 40753-40793.

Salmaninejad, A., & Mayorga, R. V. (2021). Sensor-less Brushed DC Motor Speed Control with Intelligent Controllers. *WSEAS Transactions on Systems*, *20*, 140-148.

Sheng, J., Tang, Y., Xu, S., Tan, F., Hou, R., & Xu, T. (2023). A stable learning-based method for robotic assembly with motion and force measurements. *IEEE Transactions on Industrial Electronics*, *71*(9), 11093-11103.

Shifat, T. A., & Hur, J. W. (2020). An effective stator fault diagnosis framework of BLDC motor based on vibration and current signals. *IEEE Access*, *8*, 106968-106981.

Sira-Ramirez, H., & Oliver-Salazar, M. A. (2012). On the robust control of buck-converter DC-motor combinations. *IEEE transactions on power electronics*, *28*(8), 3912-3922.

Tofoli, F. L., Pereira, D. d. C., Josias de Paula, W., & Oliveira Junior, D. d. S. (2015). Survey on non-isolated high-voltage step-up dc–dc topologies based on the boost converter. *IET Power Electronics*, *8*(10), 2044-2057.

Vazquez-Sanchez, E., Sottile, J., & Gomez-Gil, J. (2016). A novel method for sensorless speed detection of brushed dc motors. *Applied Sciences*, *7*(1), 14.

Xu, S., Liu, J., Yang, C., Wu, X., & Xu, T. (2021). A learning-based stable servo control strategy using broad learning system applied for microrobotic control. *IEEE Transactions on Cybernetics*, *52*(12), 13727-13737.

Zhang, R., & Gao, L. (2022). The Brushless DC motor control system Based on neural network fuzzy PID control of power electronics technology. *Optik*, *271*, 169879.

Author(s) Information

Ekraam H. Al-Zaidi
Thi Qar University
Thi Qar, Iraq
Contact e-mail: *ekraam.h.j@utq.edu.iq*

To cite this article:

Al-Zaidi, E.H. (2025). Supervised learning for adaptive BLDC motor control: Integrating classical PID with neural networks. *The Eurasia Proceedings of Science, Technology, Engineering and Mathematics (EPSTEM), 37,* 180-193.