



Reinforcement Learning-Based Control Strategy for BLDC Motors

Anwar Hasni^{*}, Hassan El Fadil^{}, Abdellah Lassioui^{}, Abdessamad Intidam^{}, Marouane El Ancary^{},
Yassine El Asri^{}, Soukaina Nady^{}

ISA Laboratory, National School of Applied Sciences (ENSA), Ibn Tofail University, Kénitra 14000, Morocco

Corresponding Author Email: hasni.anwar20@gmail.com

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.580603>

ABSTRACT

Received: 5 May 2025

Revised: 13 June 2025

Accepted: 23 June 2025

Available online: 30 June 2025

Keywords:

BLDC motor, speed control, machine learning, reinforcement learning, MATLAB

Machine learning has undergone significant advances in recent years, especially in the field of control systems, enabling the development of fully autonomous solutions. This article presents a simulation-based study for the control of a Brushless DC (BLDC) motor using reinforcement learning (RL). A Twin Delayed Deep Deterministic Policy Gradient (TD3) agent is implemented as the control strategy. The performance of the proposed controller is evaluated through two test scenarios: one focused on reference tracking accuracy, and the other on robustness under variable torque conditions. The simulation results show a stable and accurate behavior, with response times ranging from 1.2 to 1.9 seconds for reference tracking and from 1.6 to 1.8 seconds in the presence of load disturbances. These performances highlight the ability of reinforcement learning to significantly enhance the precision control of BLDC motors, which remains a challenge in many applications, particularly in complex, dynamic, and nonlinear environments.

1. INTRODUCTION

Brushless DC (BLDC) motors run on a direct voltage source. They are becoming more popular and widely used. This is because they are constantly being improved. New materials make them stronger and more reliable [1]. They also use less energy, which makes them more efficient. In addition, the electronic parts that control them are getting better. These parts now use digital circuits instead of brushes, which were used in traditional DC motors [2]. These motors serve as a replacement or alternative solution to conventional motors, as they offer a very good power-to-weight ratio and a higher capacity to operate across wider speed ranges [3]. The weaknesses of brushed motors, which BLDC motors successfully address, include lower efficiency, mechanical wear at the commutation level, the need for regular inspection and maintenance, and the requirement for expensive control systems. Due to their favorable electrical and mechanical characteristics, BLDC motors are frequently used in high-precision control systems such as in automotive, aerospace, medical applications, measuring equipment, industrial automation, and electromechanical actuation systems. Numerous control techniques have been developed to maximize the performance and robustness of BLDC motor drive systems [4].

A large number of control models, such as the classical PID controller, can be optimized using online or offline optimization algorithms, or through methods such as fuzzy logic or artificial neural networks (ANN) to determine their parameters [5]. The non-adaptive fuzzy logic controller (FLC) [6], the adaptive fuzzy logic controller, ANFIS [7], sliding mode control, as well as model predictive control (MPC) [8],

have also been developed for the speed control of brushless DC motors. The majority of industrial applications still rely on classical PID controllers due to their simplicity and robustness. However, classical PID controllers are generally not effective when the processes involved are higher-order systems, time-delay systems, nonlinear systems, complex or imprecise systems, lacking accurate mathematical models, or subject to uncertainties [9].

Today, methods such as neural networks, random forests, and adaptive neuro-fuzzy inference systems (ANFIS) are used to estimate or dynamically adjust the parameters of PID controllers, to model complex systems, or to predict changes in the behavior of electric motors [10].

Machine learning has revolutionized the field of electric motor control by introducing intelligent approaches capable of adapting to dynamic and nonlinear environments [11]. Unlike traditional methods based on strict mathematical models, machine learning enables the extraction of control laws from measured data, even in the presence of uncertainties, disturbances, or variations in system parameters [12, 13].

More recently, reinforcement learning (RL) has attracted significant interest in the control of electric motors, particularly in electric vehicles, drones, and autonomous robots. This approach is based on the interaction between an agent and its environment, learning to optimize its actions by maximizing a cumulative reward, which is particularly well-suited for speed or torque control. Algorithms such as Q-learning [14], Deep Q-Networks (DQN) [15], and Proximal Policy Optimization (PPO) enable the implementation of adaptive and optimal control strategies without the need for an accurate model of the system [16, 17].

Given the limitations of conventional control methods

particularly their reliance on accurate models and lack of robustness in the face of uncertainty Reinforcement Learning is attracting growing interest as a model free, data driven control strategy. Unlike traditional approaches, RL enables the design of optimal control policies through interaction with the environment, learning from system states, performed actions, and received rewards [18]. This approach allows RL agents to identify effective strategies without requiring complex mathematical models, making it especially suitable for systems that are difficult to model. Among its main advantages is its dynamic adaptability, which ensures good performance despite variations in temperature, wear, or external disturbances. RL also stands out for its ability to operate without an explicit model, in contrast to PI or MPC controllers that require precise motor modeling [19]. Finally, it excels at handling the nonlinearities inherent in electric motors and power electronics, by learning complex relationships directly from data. These features make RL a promising approach for intelligent motor control in real world, uncertain environments [20].

Highly precise speed control of electric motors, especially brushless motors, is a major challenge in most industrial and automotive applications, due to the high demands placed on performance, stability and robustness. Conventional control techniques such as PID controllers or fuzzy logic sometimes demonstrate their limits, especially under non-linear, uncertain or disturbed conditions. In this context, reinforcement learning is an interesting alternative. It enables the development of intelligent controllers capable of learning and conforming to system dynamics in real time, without the need for a precise model. This is the motivation behind this article, which aims to:

- Exploit the application of artificial intelligence methods in electric motor control systems, in particular BLDC motors;
- Software design of an intelligent RL-based controller for high-performance, precise control of BLDC motors;
- Overcome the difficulties of standard methods by applying a more flexible, higher-performance approach to BLDC engine control;
- Provide a strong solution to the uncertainties and non-linearities present in real-action scenarios.

This article, which focuses on the speed control of a brushless motor using reinforcement learning, is structured to cover all the essential aspects of this innovative approach. It begins with an introduction that presents the general context, the main control techniques developed in the literature, as well as the motivation behind this study. The second part is devoted to the mathematical modeling of the BLDC motor, a crucial step to accurately represent the system to be controlled. The third part introduces the reinforcement learning algorithm used, detailing the role of the agent and the training process implemented. The fourth part is dedicated to the presentation and analysis of the obtained results, allowing for an evaluation of the effectiveness of the proposed method. Finally, a conclusion summarizes the main contributions of the study and suggests future research directions to improve and extend this approach.

2. MATHEMATICAL MODELING OF BLDC MOTOR

The mathematical representation of the BLDC three-phase motor model with two pairs of poles is detailed in this first

subsection, based on its electrical and mechanical equations [21]. The stator consists of a full-pitch winding connected in a Y (star) configuration, while the rotor features a smooth-surfaced polar structure. Three Hall-effect sensors are symmetrically positioned with a 120-degree electrical offset, ensuring precise and balanced commutation [22, 23]. Furthermore, the mathematical equations describing the behavior of the BLDC motor are derived based on a set of simplifying assumptions, which are detailed below.

- Disregarding cardiac saturation, eddy currents and deceleration.
- Observe the magnetic field expanding from the air gap as a trapezoidal wave with a smooth top covering 120 electrical angles after removing the armature feedback.
- Assume that electrically conducting components are dispersed evenly and continuously across the armature's surface, ignoring the impact of feedback.
- Flywheel diodes and switches in the power converter control circuit provide perfect switching performance.

The BLDC motor equivalent diagram is shown in Figure 1:

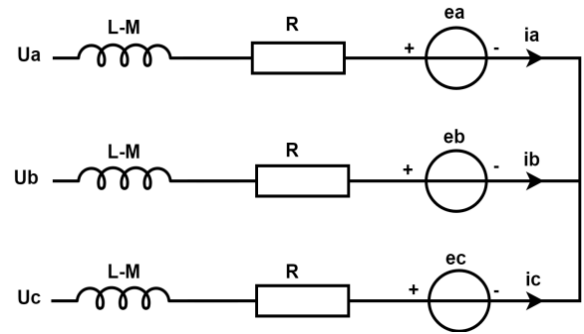


Figure 1. Brushless motor equivalent diagram

Single-phase electrical equation for a BLDC motor can be expressed as:

$$v_a = R \cdot i_a + L \cdot \frac{di_a}{dt} + e_a \quad (1)$$

Phase voltage equation in matrix form for a three-phase BLDC motor is:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = R \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + L \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (2)$$

Phase voltage equation based on line voltage can be derived as:

$$v_{ab} = v_a - v_b = R(i_a - i_b) + L \frac{d(i_a - i_b)}{dt} + (e_a - e_b) \quad (3)$$

Instantaneous electromagnetic power transferred to the rotor is given by:

$$P_{em} = e_a i_a + e_b i_b + e_c i_c \quad (4)$$

Represents the useful power (excluding losses), directly linked to torque production. Assuming no mechanical or parasitic losses:

$$P_{em} = T_e \cdot \Omega \quad (5)$$

The electromagnetic torque can be expressed as:

$$T_e = \frac{3}{2} \cdot P \cdot K_t \cdot \psi \cdot \sin(\theta) \quad (6)$$

Or, in general:

$$T_e = \frac{3}{2} \cdot P \cdot (e_a i_a + e_b i_b + e_c i_c) / \Omega \quad (7)$$

The dynamic equation describing rotor motion is:

$$T_e - T_r = J \cdot \frac{d\Omega}{dt} + B \cdot \Omega \quad (8)$$

3. FUNDAMENTALS OF REINFORCEMENT LEARNING

This section is structured into two subsections. The first presents the fundamental principles of reinforcement learning, detailing its key components and main algorithms, as well as the rationale for algorithm choice, with a particular focus on the algorithm used in this article to design and train the agent dedicated to brushless motor control. The second subsection is devoted to an in-depth analysis of the agent training results, with the aim of evaluating its performance and effectiveness in motor control.

3.1 The fundamental principles of reinforcement learning

Reinforcement learning is a machine learning technique in which an agent learns to perform optimal actions by interacting with an environment [24]. Unlike supervised learning, which requires labeled datasets, reinforcement learning operates through a trial-and-error process, where the agent receives feedback in the form of rewards based on the actions it takes [25]. This makes RL particularly well-suited for problems where it is difficult to define explicit rules or provide comprehensive training data.

Reinforcement learning is based on the components illustrated in Figure 2. These components are defined on a rigorous mathematical foundation, and each plays a vital role in the overall RL process, contributing to decision-making and the optimization of the agent's performance.

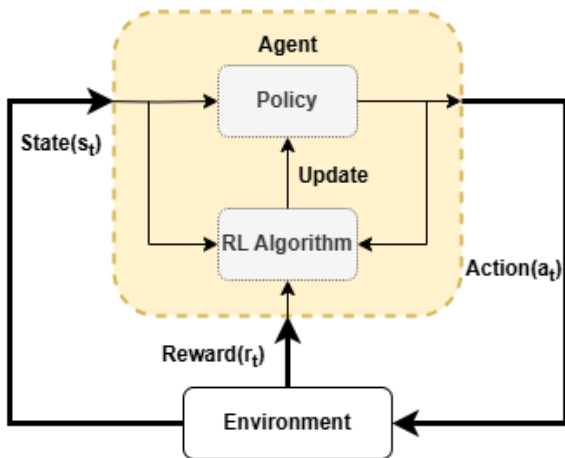


Figure 2. Diagram of the learning process with the various functions

Agent: Acts as a central learning and decision-making entity. It interacts with the environment by observing it, performing appropriate actions and adjusting its behavior according to the rewards received based on the quality of its decisions [26].

Environnement: Describes the external system with which the agent interacts. It provides the agent with observations about its current state, reacts to actions by evolving towards new states, and assigns rewards according to the effects of these actions [27].

States: State, Denoted s , describes the configuration of the environment at a given moment and provides essential information that the agent uses to guide its decisions. In the context of motor control, a state can represent physical quantities such as motor speed, position or current [28].

Actions: Actions, a , represent the decisions or interventions the agent can make to influence the environment. For example, in the case of a motor, this might involve adjusting the applied voltage or switching the inverter control signals. The agent's main objective is to choose the most effective actions to obtain favorable results [29].

Policy: The policy, noted π , defines the agent's decision-making strategy by establishing a correspondence between the observed states and the actions to be taken. It governs the agent's behavior in each situation. A policy can be deterministic, associating a single action with each state, or stochastic, selecting actions according to a probability distribution [30].

Reward: The reward, r , is a digital signal sent by the environment to the agent after each action, indicating the level of relevance or quality of the result obtained. It can be positive, negative or zero, and plays a central role in learning by guiding the agent towards the desired control objectives. For example, a high reward may be assigned when the target speed is reached, while exceeding current limits may result in a penalty. A reward function is often defined to minimize speed error while respecting system constraints [31].

Algorithm: Reinforcement learning algorithms are widely used in motor control, each offering specific advantages depending on its learning approach. Value-based methods, such as Deep Q-Networks (DQN), estimate the value of actions to guide policy selection and have proven effective in high-dimensional control tasks. Policy-based methods, such as Policy Gradient (PG), directly optimize the control policy without relying on a value function, allowing flexible adaptation to complex environments. Actor-critic methods combine both approaches for improved performance [32]. Among these, the Deep Deterministic Policy Gradient (DDPG) algorithm is particularly well-suited for continuous control in motor applications. Building on DDPG, the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm introduces significant improvements by reducing overestimation bias in the critic network, enabling more stable and reliable learning. TD3 stands out as a powerful and efficient solution for precise control of brushless motors in continuous action spaces.

In this study, the choice of the Twin Delayed Deep Deterministic Policy Gradient algorithm for controlling the speed of a BLDC motor is justified by the fact that TD3 represents an enhanced version of the Deep Deterministic Policy Gradient. DDPG relies on an actor-critic architecture, where the actor $\mu(s|\theta^\mu)$ generates a deterministic action for a given state, and the critic $Q(s,a|\theta^Q)$ estimates the value of that action. The critic is updated by minimizing the following

loss function:

$$L(\theta^Q) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \left[\left(Q(s,a|\theta^Q) - (r + \gamma Q(s', \mu(s'|\theta^{\mu'})|\theta^{Q'})) \right)^2 \right] \quad (9)$$

However, DDPG generally suffers from instability issues due to Q-value overestimation, poorly controlled noisy exploration, and policy update oscillations. To overcome these limitations, TD3 introduces three major enhancements:

- **Clipped Double Q-learning:** Two Q-value estimators Q_1 and Q_2 are used, and the target value is computed using their minimum to reduce overestimation:

$$y = r + \gamma \min_{i=1,2} Q_i(s', a') \text{ where } a' = \mu(s'|\theta^{\mu'}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2) \quad (10)$$

- **Target Policy Smoothing:** Gaussian noise is added to the target action to make the policy more robust to minor variations:

$$a' = \mu(s') + \text{clip}(\epsilon, -c, c) \text{ with } \epsilon \sim \mathcal{N}(0, \sigma^2) \quad (11)$$

This prevents the agent from exploiting irrelevant sharp changes in the value function 3. **Delayed Policy Updates:** The policy network μ , along with the target networks Q' and μ' , are updated less frequently (every d steps), which helps reduce variance and improve learning stability. Thanks to these enhancements, the TD3 algorithm is employed in this study to enhance the robustness of the BLDC motor speed control in the presence of load torque disturbances. Experimental results demonstrate that TD3 achieves superior performance in terms of Integral Square Error (ISE), reduced overshoot, and improved robustness compared to DDPG. Moreover, TD3 generates a smoother control effort, thereby minimizing mechanical and thermal stress on the motor and its

power electronics. Consequently, the selection of TD3 for this application is strongly supported by its ability to learn more stable, accurate, and generalizable control policies in dynamic and noisy environments such as those encountered in BLDC motor systems.

The Table 1 shows the algorithm defining a reward function used by a reinforcement learning (RL) based controller for speed control of a BLDC motor. The algorithm is initiated by estimating the absolute error between speed reference and actual motor speed. If this error is greater than or equal to 1, a quadratic penalty is applied in the form of a negative reward $-e^2$, to discourage large errors. On the other hand, for errors less than 1, a non-linear transformation is applied, assigning the error a negative value of its square root, which reinforces learning even in the case of small deviations. If the error is strictly negative (a rare but covered theoretical case), it is simply reduced to zero. This reward mechanism ensures fast, accurate tracking of the speed setpoint, while guaranteeing learning stability and robustness to disturbances. By dynamically decoupling from the error amplitude, this function enables the RL agent to adjust the control signal efficiently to minimize speed deviation in non-linear and uncertain environments.

Table 1. An algorithm for the process of learning

Algorithm: Reward Function	
1:	$e \leftarrow \text{error} $
2:	if $e \geq 1$ then
3:	reward $\leftarrow -(e^2)$
4:	else
5:	if $e \geq 0$ then
6:	$e \leftarrow -\sqrt{e}$
7:	else
8:	$e \leftarrow 0$
9:	end if
10:	end if

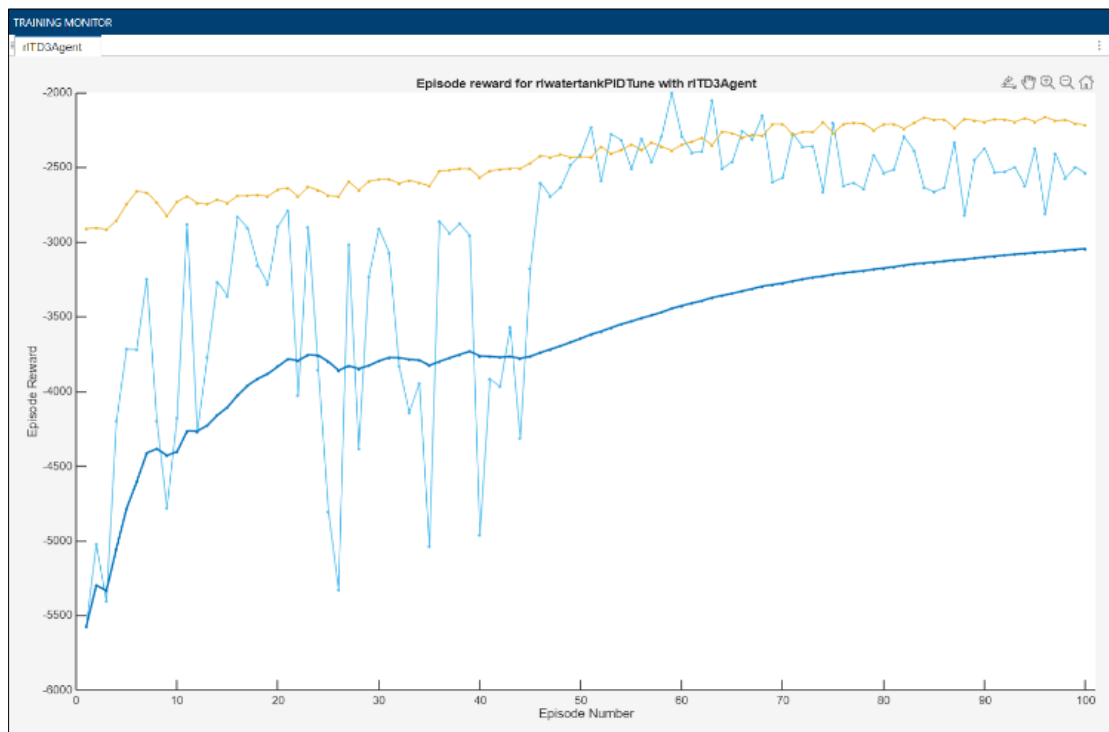


Figure 3. The evolution of reward per episode during training of a TD3 agent

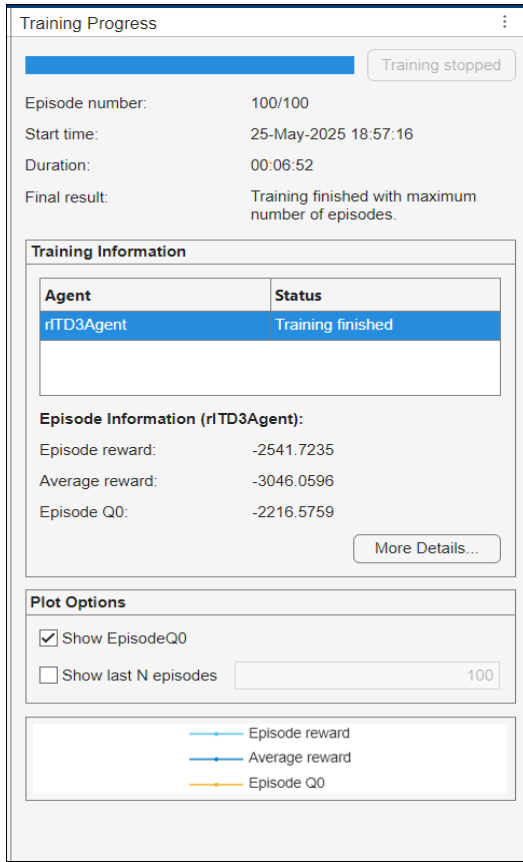


Figure 4. The table of training progress

3.2 Training agent and results analysis

The Figure 3 illustrates the evolution of the reward per episode during training of a Twin Delayed Deep Deterministic Policy Gradient agent for speed control of a BLDC motor. The x-axis represents the episode number, while the y-axis shows the reward obtained, reflecting the control performance. The curves show both rewards per episode (thin, irregular lines) and rolling averages (thick lines), highlighting the agent's progressive improvement over time. Initially, rewards are low, reflecting an inefficient control strategy. However, with learning, rewards increase steadily, indicating that the agent is progressively optimizing motor speed control. Convergence towards a higher reward reflects a stabilization of the agent's strategy, and thus a more efficient and stable speed control.

Figure 4 shows a summary of the training of an rlTD3Agent agent applied to the speed control of a BLDC motor via a TD3-based reinforcement algorithm. Training took place over 100 episodes and lasted about 7 minutes, ending normally after reaching the maximum number of episodes. Rewards indicated control performance: the final episode reward was -2541.72, and the average reward over all episodes was -3046.06, showing progressive improvement. The Q0 value for the episode (-2216.58) reflects the estimated quality of the policy in the initial state. These results show that the agent has learned a relatively effective, albeit perfectible, control strategy for stabilizing engine speed while minimizing error.

4. SIMULATION DETAILS

This section aims to provide a detailed description of the MATLAB/Simulink simulation used to implement the speed

control of a Brushless DC motor, based on the Twin Delayed Deep Deterministic Policy Gradient Algorithm Figure 5. The simulation diagram is structured into four main functional blocks, each serving a specific role in the learning and control loop. The first block defines the reference speed profile, representing the desired trajectory that the motor must follow. This reference is compared to the actual motor speed to generate a tracking error, which is passed to the Calculate Reward block. This block evaluates the effectiveness of the action selected by the TD3 agent, with the goal of minimizing the tracking error. The aim is to guide the reinforcement learning agent toward a more optimal and high-performance behavior over time.

The reward function is constructed using two penalty terms:

- The first, called errorPenalty, is the squared tracking error weighted by a constant factor:

$$\text{errorPenalty} = K_1 \cdot (\text{error})^2 \text{ with } K_1 = 1.125$$

- The second, named actionPenalty, penalizes the magnitude of the control action (i.e., the applied voltage), also scaled by a factor:

$$\text{actionPenalty} = K_2 \cdot u^2 \text{ with } K_2 = 0.012$$

The final reward is then calculated as:

$$\text{Reward} = -(\text{errorPenalty} + \text{actionPenalty})$$

Consequently, the smaller the tracking error and the smoother the control action, the higher (i.e., less negative) the reward, which drives the TD3 agent to learn precise, efficient, and energy-conscious control strategies.

The Generate Observations, block processes the error between the reference and actual motor speeds, as well as the integral of this error, to form a meaningful representation of the system's current state. These signals are then transformed into normalized observations, which serve as inputs to the TD3 agent. Based on these observations, the agent generates a continuous control action in this case, a DC voltage signal which is applied to the BLDC motor in order to regulate its speed.

This action is transmitted to the BLDC motor model, which is mathematically described by a set of differential equations capturing both electrical dynamics (including inductance L , resistance R , and torque constant K_t) and mechanical dynamics (such as moment of inertia J , friction coefficient B , and external load torque). The model also includes back electromotive force (EMF) feedback through the constant K_e , ensuring a highly realistic and physically accurate simulation of the motor's behavior.

For the Terminate Early block, enables early termination of the simulation if specific performance or safety criteria are violated. This mechanism enhances training efficiency by avoiding unnecessary computations during episodes where learning is no longer productive or could lead to unstable behaviors.

5. RESULTS AND DISCUSSION

This section aims to present and analyze the results obtained from the simulation carried out using MATLAB Simulink for the speed control of a brushless motor, by using one of the

machine learning tools, which is reinforcement learning. The analysis of the results is conducted through two scenarios to thoroughly evaluate the validity of the controller used. The first focuses on the tracking performance of the imposed speed

reference, and the second on robustness, by varying the load torque to observe the system's response. The parameters used in this simulation are presented in the Table 2.

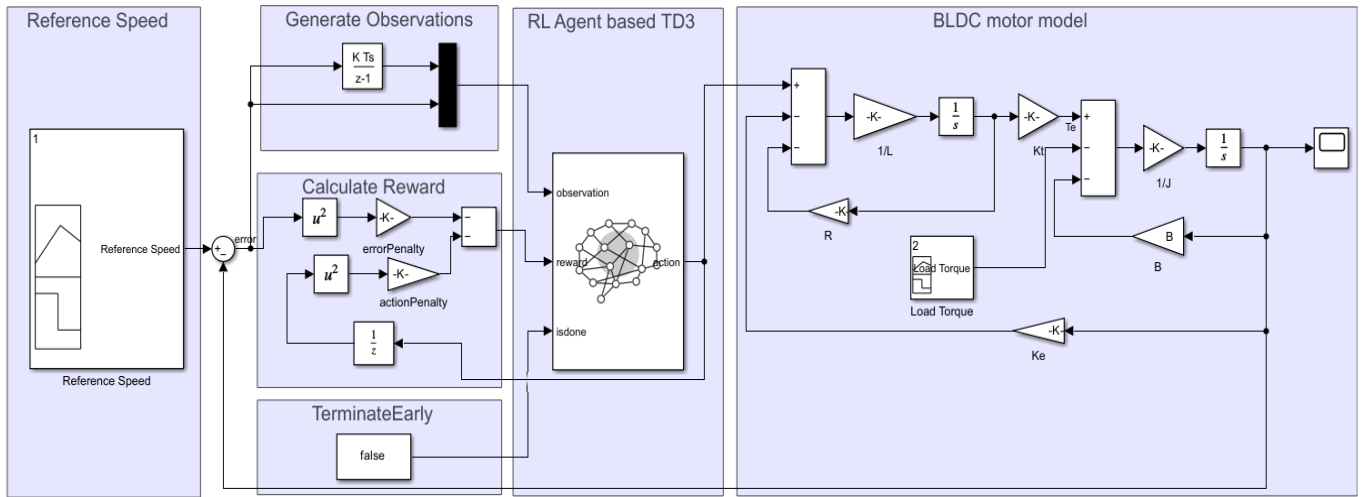


Figure 5. Simulation diagram of this BLDC motor speed control study using RL

Table 2. Parameters for the simulation [33]

Variable	Mark
Vdc	300 V
R	1.5 Ω
L	0.0115 H
M	0.005 H
K_t	60.3e-3 Nm/A
K_e	60.3e-3 Vs/rad
J	0.001 Kg.m ²
B	0.002 N.m

interaction of the agent, which forms the basis of the controller, with the speed changes. Due to the proportionality between the brushless motor speed and the voltage applied to the motor, the controller generates a voltage of 125 V for a speed of 80 rad/s and 156 V for a speed of 100 rad/s. These results in this section are obtained in the absence of load torque, as shown in Figure 9.

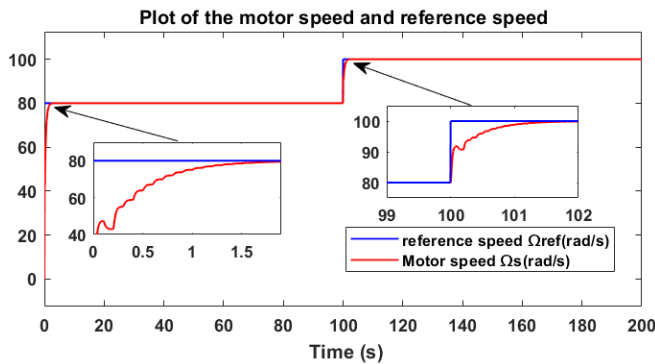


Figure 6. Motor speed variation during change in reference speed without load condition

5.1 First scenario: Tracking performances

5.1.1 Performance tracking under no-load conditions

The purpose of this part of the first scenario is to study the motor's response at no load, without applying any load torque. Figure 6 shows the motor response with a reference variation from 80 rad/s to 100 rad/s. For the first startup interval, the motor speed follows the 80 rad/s reference after just 1.5 seconds, and for the second interval, it follows the 100 rad/s reference after just 1.2 seconds. As for the error, it vanishes, as shown in Figure 7. Figure 8 illustrates the controller's output voltage during the speed variations, clearly showing the

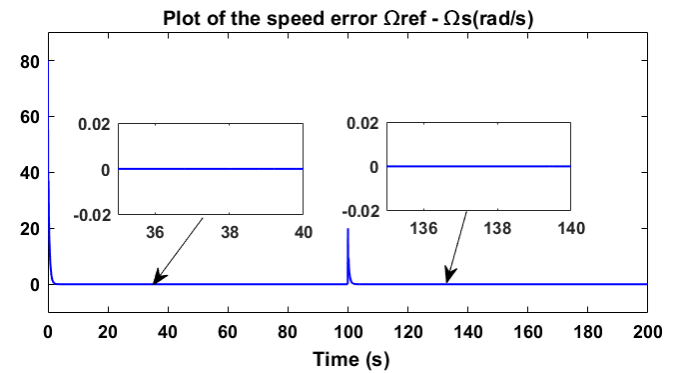


Figure 7. Speed tracking error with null load torque

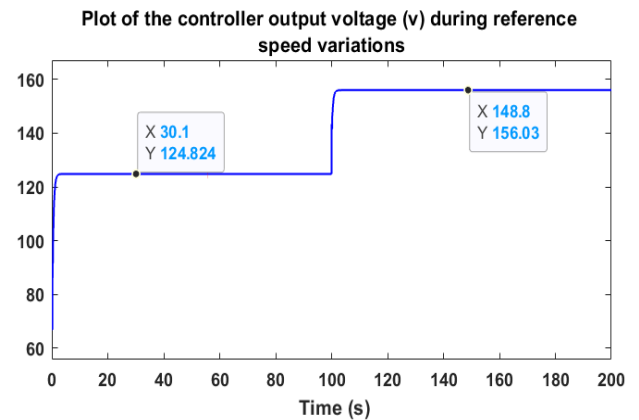


Figure 8. Controller output voltage during speed reference variations without load torque

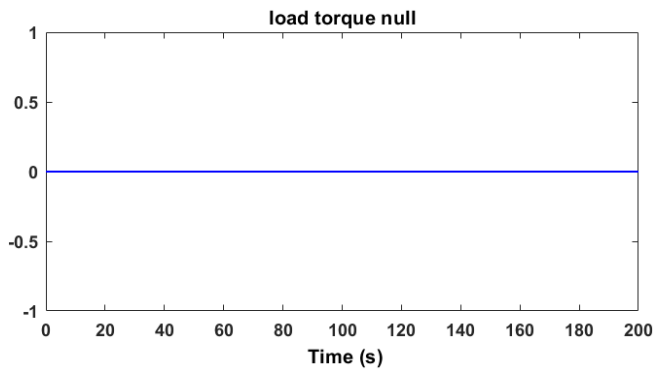


Figure 9. Load torque applied to the motor

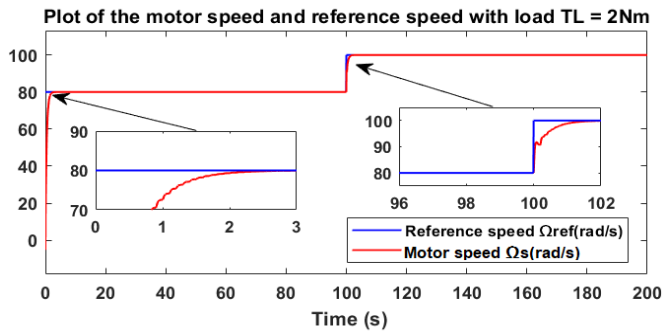


Figure 10. Motor speed variation during change in reference speed with load condition

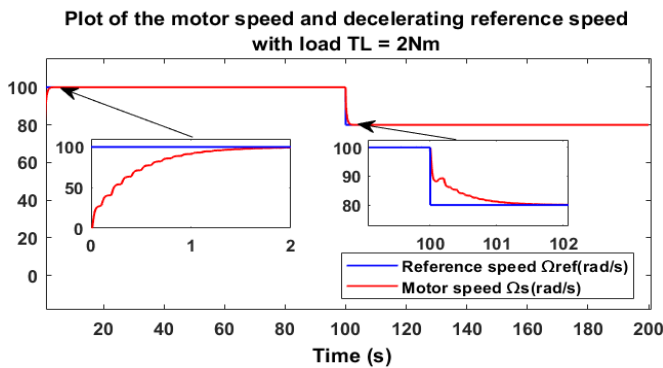


Figure 11. Motor speed variation when decreasing reference speed with load torque

5.1.2 Performance tracking under load conditions

In order to further evaluate the performance, this second part of the first scenario applies a constant load torque to the motor, $T = 2 \text{ Nm}$. As already mentioned in the first part, Figure 10 shows the motor's response to a variation in the reference speed from 80 rad/s to 100 rad/s, but this time with a load torque equal to 2 Nm. At startup, the motor follows the reference after 1.9 seconds, and for the second variation, the response follows the reference after 1.5 seconds. Similarly, when decreasing the speed from 100 rad/s to 80 rad/s, the results are the same, as shown in Figure 11. Regarding the error between the two speeds, it vanishes, as shown in Figure 12. Figure 13 shows the controller output applied to the brushless motor. The controller generates a voltage of 174.5 V to reach the reference speed of 80 rad/s, and 205.7 V to reach the speed of 100 rad/s. These results are obtained under the application of a constant load torque of 2 Nm, as shown in Figure 14. To more rigorously evaluate the speed tracking performance, a more complex reference speed scenario was

adopted, as illustrated in Figure 15. The resulting curve demonstrates accurate and fast speed tracking, even in the presence of abrupt changes in the reference, with a response time ranging from 0.5 s to 1.8 s.

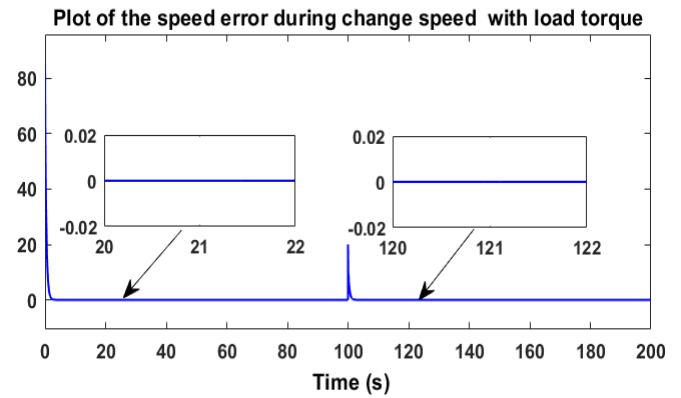


Figure 12. Speed tracking error with load condition

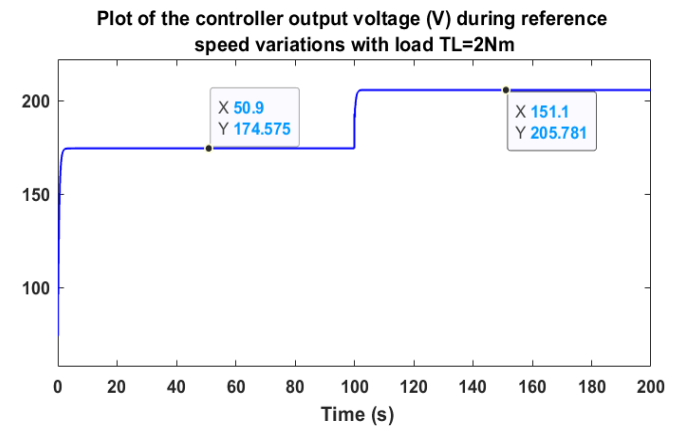


Figure 13. Controller output voltage during speed reference variations with load torque

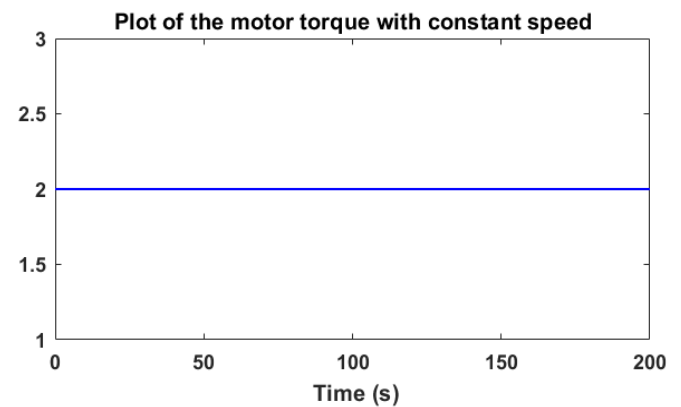


Figure 14. Load condition applied to the motor

5.2 Second scenario: Robustness performances

Regarding the robustness of the controller against external disturbances such as load torque variations, two tests were carried out to properly evaluate its robustness. The first test involves varying the torque in an increasing manner, from 2 Nm to 4 Nm. Figure 16 shows that the motor speed closely follows the desired value of 100 rad/s, with a downward deviation of about 20% from 100 rad/s, which is expected

during a torque increase. However, the speed returns to its reference value after only 1.8 seconds, which sufficiently demonstrates the effectiveness of the controller in handling torque variations. Figure 17 shows that the speed error is eliminated. The voltage generated by the controller is illustrated in Figure 18, where it can be seen that the motor supply voltage increases from approximately 205.7V to 255.5V as the load torque varies from 2 Nm to 4 Nm. These measurements were taken under a variable load torque increasing from 2 Nm to 4 Nm, as shown in Figure 19.

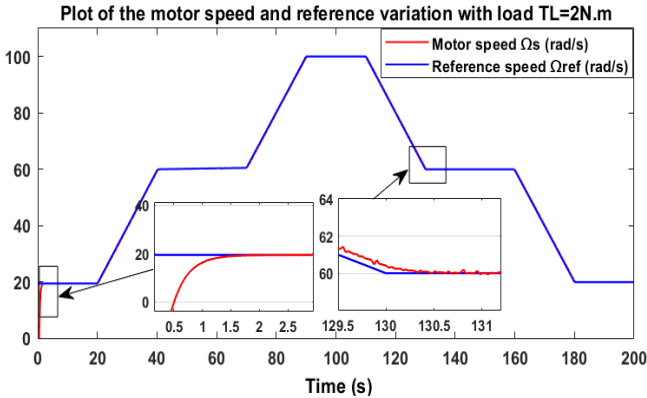


Figure 15. Load condition applied to the motor

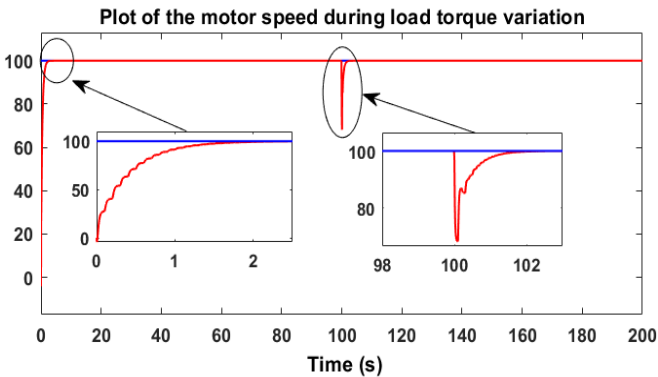


Figure 16. Variations in motor speed as load torque increases

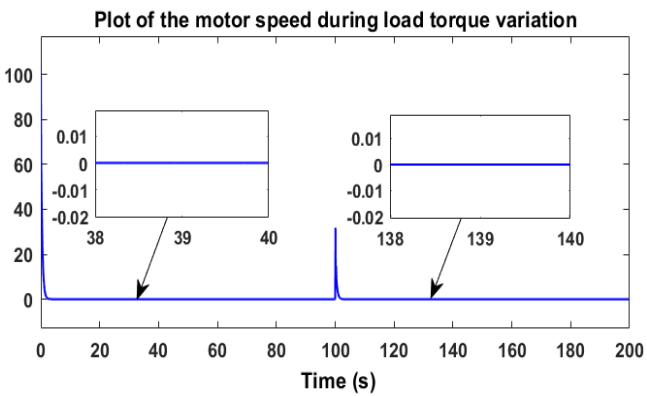


Figure 17. Speed error during load torque variation and growth

Regarding the second robustness test, it consists of varying the load torque in a decreasing manner, from 4 Nm to 2 Nm, in order to observe the response of the controller-motor system. Figure 20 shows that the motor speed perfectly follows the desired value of 100 rad/s, with an overshoot of about 20% relative to 100 rad/s, which is logical when the load torque

decreases. However, the motor returns to its reference speed after only 1.6 seconds, which sufficiently demonstrates the effectiveness of this controller in handling torque variations. Figure 21 shows that the speed error cancels out. The voltage generated by the controller is illustrated in Figure 22, where it can be seen that the motor supply voltage drops from approximately 255.5 V to 205.7 V as the load torque decreases from 4 Nm to 2 Nm. These measurements were taken under a decreasing variable load torque from 4 Nm to 2 Nm, as shown in Figure 23. These two robustness tests demonstrate the reliability and the advantages of using reinforcement learning-based agents employing the Twin Delayed Deep Deterministic Policy Gradient algorithm in the regulation and control of brushless motors. To further evaluate the robustness of the TD3-based reinforcement learning controller, Figure 24 illustrates its resilience against sudden and rapid load torque disturbances ranging between 0 N.m, 1 N.m, and 2 N.m showed in the Figure 25. The controller maintains a high-performance response with ripple limited between 20% and 30%, and a response time ranging from 0.5 s to 1.8 s.

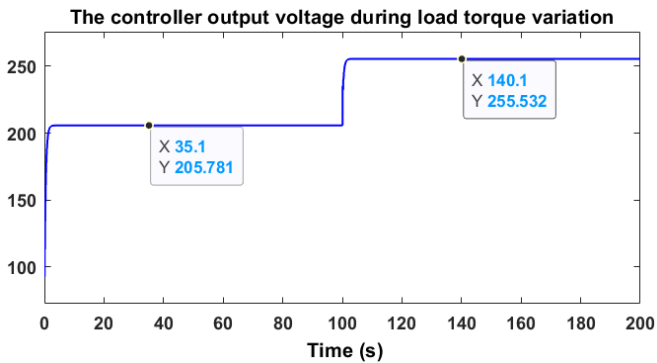


Figure 18. Controller output voltage as load torque increases

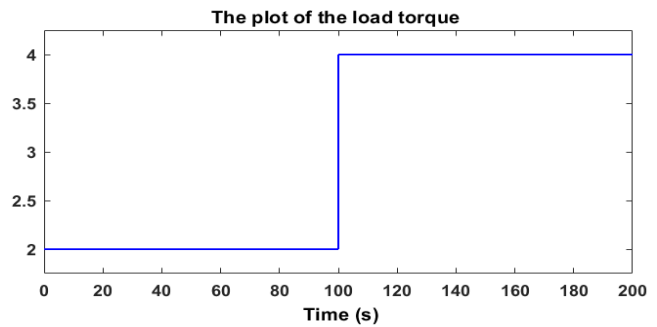


Figure 19. Increasing load condition applied to the motor

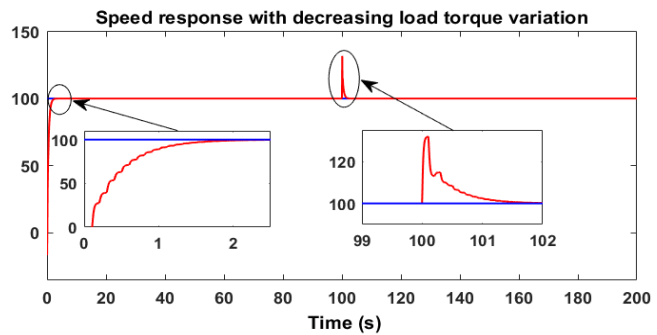


Figure 20. Variations in motor speed as load torque decreases

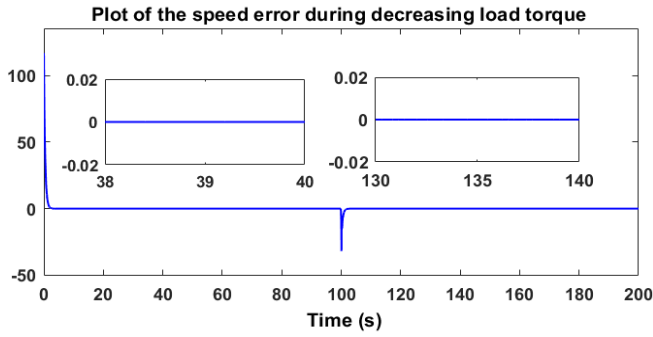


Figure 21. Speed error when varying and decreasing load torque

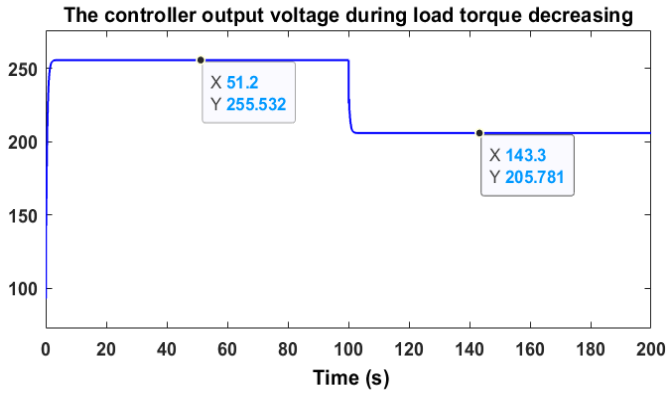


Figure 22. Controller output voltage as load torque decreases

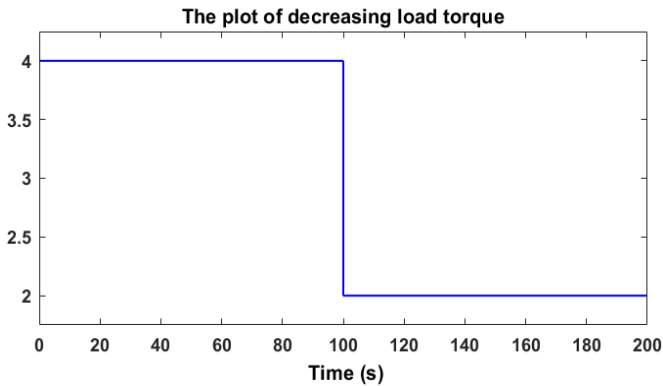


Figure 23. Decreasing load condition applied to the motor

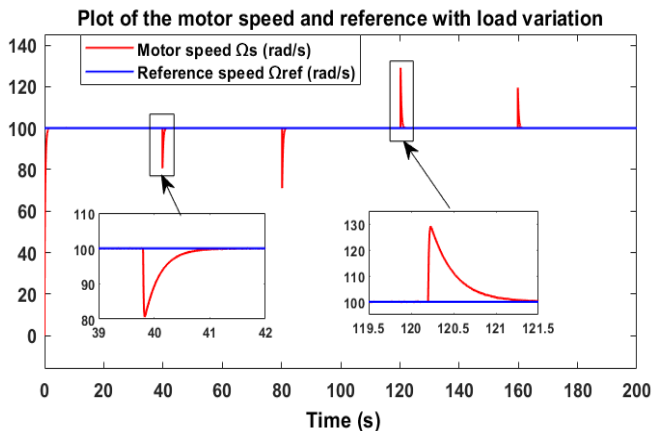


Figure 24. Speed response to load torque variations

As part of the evaluation and enhancement of the approach

studied in this article, which focuses on speed regulation of a BLDC motor using a Reinforcement Learning-based controller (TD3 algorithm), this section presents a comparative study between the RL controller and two conventional controllers: PID and Fuzzy Logic. The comparison is based on four key criteria: robustness to disturbances, real-time adaptation capability, design complexity, and maintenance requirements. It is worth noting that the performance of the RL-based controller, as demonstrated in this work, could be further improved with a more powerful PC and CPU, since the approach requires a significant training time. The results of this comparative analysis are summarized in the following Table 3.

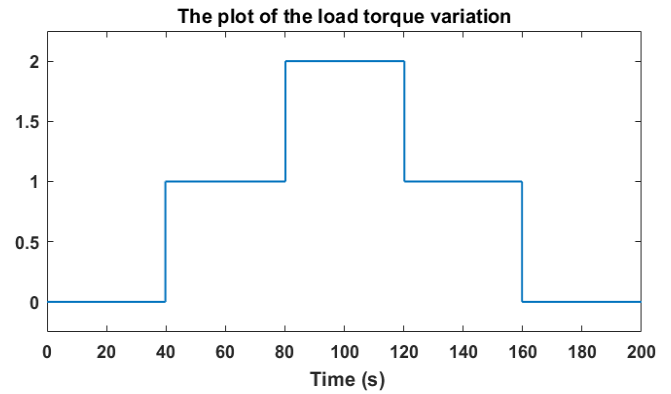


Figure 25. The variation in load torque applied to the motor

Table 3. Comparison table between the three controllers

Criterion	PID	Fuzzy Logic	RL Based TD3
Robustness to disturbances	Average	Good	Excellent
Real-time adaptation	No	Partial	Yes (self-adjustment through continuous learning)
Design complexity	Low (gain tuning)	Medium (rule base to be defined)	High (initial training required)
Maintenance	Regular manual tuning	Rule updates possible	Self-adaptation, no manual retuning required

6. CONCLUSIONS

This article aims to study and simulate the development of a controller based on machine learning, specifically using reinforcement learning, built upon the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm for the control of a brushless motor. The results clearly demonstrate the performance in terms of tracking the motor speed set as a reference, as well as the robustness in maintaining this tracking despite external constraints, such as a variable load torque. This article also confirms the relevance of using AI-based controllers, particularly reinforcement learning, in the field of electric motor control, especially in the automotive sector. The next steps, as future perspectives, include experimentally validating this RL-based control approach in the laboratory and applying it to both synchronous and

asynchronous motors to better highlight the potential of this method for nonlinear systems. Comparisons with other controllers such as ANN, ANFIS, Fuzzy Logic Controllers, and the PID controller are also planned.

ACKNOWLEDGMENT

This work was carried out with the support of the Centre National de la Recherche Scientifique et Technique (CNRST), as part of the "PhD-Associate Scholarship - PASS" program.

REFERENCES

- [1] Dakheel, H.S., Abdullah, Z.B., Shneen, S.W. (2023). Advanced optimal GA-PID controller for BLDC motor. *Bulletin of Electrical Engineering and Informatics*, 12(4): 2077-2086. <https://doi.org/10.11591/eei.v12i4.4649>
- [2] Wang, T., Wang, H., Hu, H., Lu, X., Zhao, S. (2022). An adaptive fuzzy PID controller for speed control of brushless direct current motor. *SN Applied Sciences*, 4(3): 71. <https://doi.org/10.1007/s42452-022-04957-6>
- [3] Liu, Z., Fang, L., Jiang, D., Qu, R. (2022). A machine-learning-based fault diagnosis method with adaptive secondary sampling for multiphase drive systems. *IEEE Transactions on Power Electronics*, 37(8): 8767-8772. <https://doi.org/10.1109/TPEL.2022.3153797>
- [4] Matsuura, K., Akatsu, K. (2020). A motor control method by using Machine learning. In 2020 23rd International Conference on Electrical Machines and Systems (ICEMS), Hamamatsu, Japan, pp. 652-655. <https://doi.org/10.23919/ICEMS50442.2020.9290989>
- [5] Guan, Q., Yao, X., Lin, Z., Wang, J., Lu, H.H.C., Fernando, T., Zhang, X. (2024). A Robust Control Scheme for PMSM based on Integral Reinforcement Learning. *IEEE Transactions on Transportation Electrification*, 11(1): 4214-4223. <https://doi.org/10.1109/TTE.2024.3455574>
- [6] Krishnan, G.H., Prabhu, S., Reddy, L.S., Chandrasekar, P., Kumar, T.D., Ushaswini, P.H. (2024t). ANN based speed control of BLDC motor using six-step inverter. In 2024 10th International Conference on Electrical Energy Systems (ICEES), Chennai, India, pp. 1-6. <https://doi.org/10.1109/ICEES61253.2024.10776817>
- [7] Kristiyono, R., Wiyono, W. (2021). Autotuning fuzzy PID controller for speed control of BLDC motor. *Journal of Robotics and Control (JRC)*, 2(5): 400-407. <https://doi.org/10.18196/jrc.25114>
- [8] Kroičs, K., Būmanis, A. (2024). BLDC motor speed control with digital adaptive PID-fuzzy controller and reduced harmonic content. *Energies*, 17(6): 1311. <https://doi.org/10.3390/en17061311>
- [9] Kaul, S., Tiwari, N., Yadav, S., Kumar, A. (2021). Comparative analysis and controller design for BLDC motor using PID and adaptive PID controller. *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, 14(6): 671-682. <https://doi.org/10.2174/2352096514666210823152446>
- [10] Schenke, M., Kirchgässner, W., Wallscheid, O. (2019). Controller design for electrical drives by deep reinforcement learning: A proof of concept. *IEEE Transactions on Industrial Informatics*, 16(7): 4650-4658. <https://doi.org/10.1109/TII.2019.2948387>
- [11] Hollins, Z. (2025). Dynamic performance-oriented PID optimization for BLDC motor speed control via enhanced firefly algorithm. *Journal of Computer Technology and Software*, 4(2). <https://doi.org/10.5281/zenodo.14984957>
- [12] Taiwo, O., Ezugwu, A.E., Oyelade, O.N., Almutairi, M.S. (2022). Enhanced intelligent smart home control and security system based on deep learning model. *Wireless Communications and Mobile Computing*, 2022(1): 9307961. <https://doi.org/10.1155/2022/9307961>
- [13] Maghfiroh, H., Ramelan, A., Adriyanto, F. (2021). Fuzzy-PID in BLDC motor speed control using MATLAB/Simulink. *Journal of Robotics and Control (JRC)*, 3(1): 8-13. <https://doi.org/10.18196/jrc.v3i1.10964>
- [14] A Mohammed Eltoum, M., Hussein, A., Abido, M.A. (2021). Hybrid fuzzy fractional-order PID-based speed control for brushless DC motor. *Arabian Journal for Science and Engineering*, 46(10): 9423-9435. <https://doi.org/10.1007/s13369-020-05262-3>
- [15] Zhang, S., Wallscheid, O., Porrmann, M. (2023). Machine learning for the control and monitoring of electric machine drives: Advances and trends. *IEEE Open Journal of Industry Applications*, 4: 188-214. <https://doi.org/10.1109/OJIA.2023.3284717>
- [16] Raia, M.R., Ciceo, S., Chauvicourt, F., Martis, C. (2023). Multi-attribute machine learning model for electrical motors performance prediction. *Applied Sciences*, 13(3): 1395. <https://doi.org/10.3390/app13031395>
- [17] Metwly, M.Y., Luckett, B., Clark, L., He, J., Xie, B. (2025). Neural network based digital twin health monitoring of BLDC motor drives for robots. In 2025 IEEE Applied Power Electronics Conference and Exposition (APEC), Atlanta, GA, USA, pp. 919-924. <https://doi.org/10.1109/APEC48143.2025.10977518>
- [18] Singh, A., Yadav, S., Sarangi, S., De, S., Singh, A.K., Singh, R.K. (2025). Optimizing BLDC motor performance: A study of PID, artificial neural network and fuzzy logic controllers. In 2025 International Conference on Innovation in Computing and Engineering (ICE), Greater Noida, India, pp. 1-6. <https://doi.org/10.1109/ICE63309.2025.10984074>
- [19] Boumaalif, Y., Ouadi, H., Giri, F. (2024). Optimum reference current generation for switched reluctance motor torque ripple minimization in electric vehicle applications. *IFAC-PapersOnLine*, 58(13): 703-708. <https://doi.org/10.1016/j.ifacol.2024.07.564>
- [20] Percival, D. (2024). Perancangan Kontroler Neural Network Untuk Sistem Kontrol Kecepatan Motor BLDC (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).
- [21] Fathoni, K., Apriaskar, E., Salim, N.A., Hidayat, S., Suni, A.F., Hastawan, A.F., Iksan, N. (2024). Performance investigation of model predictive control for brushless DC motor. *IOP Conference Series: Earth and Environmental Science*, 1381(1): 012009. <https://doi.org/10.1088/1755-1315/1381/1/012009>
- [22] Nicola, M., Nicola, C.I., Selișteanu, D., Șendrescu, D. (2025). Rapid control prototyping of sensorless control system for PMSM based on multi-agent reinforcement learning and fractional order sliding mode control. *Engineering Science and Technology, an International Journal*, 66: 102054.

- <https://doi.org/10.1016/j.jestch.2025.102054>
- [23] Kazemikia, D. (2024). Reinforcement learning for motor control: A comprehensive review. arXiv preprint arXiv:2412.17936. <https://doi.org/10.48550/arXiv.2412.17936>
- [24] Naresh, K., Pattnaik, S. (2025). Sensorless speed control of BLDC motor drive using Volt/Hertz method. *International Journal of Electronics*, 1-26. <https://doi.org/10.1080/00207217.2025.2484699>
- [25] Hu, Z., Zhang, Y., Li, M., Liao, Y. (2025). Speed optimization control of a permanent magnet synchronous motor based on TD3. *Energies*, 18(4): 901. <https://doi.org/10.3390/en18040901>
- [26] Traue, A., Book, G., Kirchgässner, W., Wallscheid, O. (2020). Toward a reinforcement learning environment toolbox for intelligent electric motor control. *IEEE Transactions on Neural Networks and Learning Systems*, 33(3): 919-928. <https://doi.org/10.1109/TNNLS.2020.3029573>
- [27] Book, G., Traue, A., Balakrishna, P., Brosch, A., Schenke, M., Hanke, S., Kirchgässner, W., Wallscheid, O. (2021). Transferring online reinforcement learning for electric motor control from simulation to real-world experiments. *IEEE Open Journal of Power Electronics*, 2: 187-201. <https://doi.org/10.1109/OJPEL.2021.3065877>
- [28] Kudelina, K., Vaimann, T., Asad, B., Rassölkin, A., Kallaste, A., Demidova, G. (2021). Trends and challenges in intelligent condition monitoring of electrical machines using machine learning. *Applied Sciences*, 11(6): 2761. <https://doi.org/10.3390/app11062761>
- [29] Pangerang, F., Samman, F.A., Zainuddin, Z., Sadjad, R.S. (2025). Variable loaded brushless DC motor with six step commutation PID-based speed controller optimized by PSO algorithm. *Bulletin of Electrical Engineering and Informatics*, 14(1): 132-142. <https://doi.org/10.11591/eei.v14i1.8618>
- [30] Karuppannan, A., Muthusamy, M. (2021). Wavelet neural learning-based type-2 fuzzy PID controller for speed regulation in BLDC motor. *Neural Computing and Applications*, 33(20): 13481-13503. <https://doi.org/10.1007/s00521-021-05971-2>
- [31] Hussain, Q., Noor, A.S.M., Qureshi, M.M., Li, J., Rahman, A.U., Bakry, A., Mahmood, T., Rehman, A. (2025). Reinforcement learning based route optimization model to enhance energy efficiency in internet of vehicles. *Scientific Reports*, 15(1): 3113. <https://doi.org/10.1038/s41598-025-86608-5>
- [32] Mehmood, Y. (2025). Reinforcement learning in intelligent applications: Algorithms and case studies. *Journal of AI Range*, 2(1): 12-22.
- [33] Hasni, A., Lassioui, A., El Fadil, H., El Asri, Y., El Ancary, M., Bentalhik, I., El Jeilani, S., Nady, S. (2025). Cruise control of an electric vehicle propelled by a BLDC motor. In *EPJ Web of Conferences*, 330: 06008. <https://doi.org/10.1051/epjconf/202533006008>