Nanotechnology Perceptions
ISSN 1660-6795
www.nano-*ntp.com*

Cascaded Hybrid Soft Computing Controllers USING Deep Reinforcement Learning AND Fuzzy Sarsa Learning FOR Enhanced Bldc Motor Performance

[1]Jayesh Rajaram Dhuri*, [2]Dr. E. Vijay Kumar

[1]PhD. Research Scholar, Department of Electrical Engineering, SRK University, Bhopal, Madhya Pradesh, India
[2]Professor & HOD, Department of Electrical Engineering, SRK University, Bhopal, Madhya Pradesh, India
[1]Mail: jayeshdhuri87@gmail.com, [2]Mail: rvijaykumareda03@gmail.com
**Corresponding Mail: jayeshdhuri87@gmail.com*

BLDC motors have efficiency, miniaturization, and low maintenance requirements that make them a crucial aspect of modern electric drive systems. Inherent nonlinearities and parameter uncertainties, together with high torque and speed transitions, generally worsen the performance of conventional controllers in a BLDC system. In this regard, this work proposes a Cascaded Hybrid Soft Computing Controller that merges DRL with Fuzzy SARSA(λ) learning for intelligent, adaptive, and data-driven control of BLDC motors. A cascaded architecture combines the global policy optimization of DRL with the fuzzy rule-based adaptability of fuzzy reinforcement learning to guarantee real-time stability with quicker convergence and higher learning accuracy. Simulation output shows the superior transient and steady-state performance of the developed hybrid controller. The developed method registers a settling time of 0.28 s, steady-state speed of 1498 RPM, torque ripple of less than 2.95 N·m, and remarkably stable DC-link voltage around 300 V when compared to Fuzzy SARSA(λ)-only and MPC control strategies. These results confirm that the controller has superior robustness, adaptability, and accuracy under speed reversals and sudden load disturbances. The resulting cascade scheme therefore offers a strong direction for future intelligent motor drives and autonomous energy-efficient actuation systems.

Keywords: Brushless Direct Current (BLDC), Fuzzy SARSA(λ) Learning, Deep Reinforcement Learning (DRL), Cascaded Hybrid Soft Computing Controller.

1. Introduction

Brushless Direct Current (BLDC) motors are now ubiquitously essential in today's industrial automation, electric vehicle drives, aerospace, and precision manufacturing systems because they offer higher efficiency, high torque-to-weight ratio, and brushless maintenance-free

operation [1]. The brushes' absence obviates mechanical friction and electrical noise, which leads to longer operating life and better reliability over traditional brushed DC motors [2]. Despite this, the complex electromagnetic dynamics, embedded nonlinearities, and vulnerability to parameter fluctuations and external disturbances present major difficulties in realizing optimum control performance [3, 4].

Traditional control designs of BLDC motors, such as “Proportional-Integral-Derivative (PID)” controllers [5], “Field-Oriented Control (FOC)” [6], and “Direct Torque Control (DTC)” [7], have been widely used in industrial drives. Though these traditional methods show good performance under typical working conditions, they show inherent limitations in the presence of time-varying system parameters, model uncertainties, and random load disturbances [8]. The fixed-gain characteristic of conventional controllers usually leads to less-than-optimal performance, especially in transient operation and under changing operating conditions [9].

The advent of soft computing methods has transformed motor control systems with the provision of adaptive, intelligent solutions that could cope with uncertainty and nonlinearity [10]. Fuzzy logic controllers (FLCs) have proved to be great successes in BLDC motor drives because they are capable of adding expert knowledge and processing imprecise information without the need for exact mathematical models [11, 12]. Controllers based on neural networks have proved to possess better learning ability and approximation properties for sophisticated nonlinear systems [13][14]. But individual soft computing methods tend to be plagued by learning inefficiency, adaptability, and convergence guarantees [15].

Reinforcement learning (RL) has turned out to be a potent paradigm for control system synthesis with the capability for controllers to learn policies optimally by interacting with the environment in a trial-and-error manner [16]. “State-Action-Reward-State-Action (SARSA)” algorithm, being an on-policy temporal change among learning algorithm, has proved to be particularly useful in control scenarios where conservative policy updates and balance between exploration-exploitation are of vital importance [17, 18]. Progress in “Deep Reinforcement Learning (DRL)” in recent times has further increased the ability to manage high-dimensional state spaces and sophisticated control tasks [19].

The merging of several soft computing methods using hybrid frameworks has emerged as a strong candidate to overcome sole limitations and leverage complementary strengths [20]. Recent research suggests hybrid solutions blending “Circle Search Algorithm (CSA)” and “Recalling-Enhanced Recurrent Neural Network (RERNN)” for intellectual controller-driven electric drives with intricate mechanical structures and variable parameters Expansion of intellectual organizer with great performance electric drives using hybrid CSA and RERNN technique [21]. Cascaded hybrid controllers integrate various intelligent techniques in hierarchical form, where each level provides certain characteristics to the overall control performance [22]. Novel hybrid control methods blend the deterministic quality of straight torque control through the adaptive quality of fuzzy logic control in order to minimize torque ripple Hybrid control based on fuzzy logic and adaptive space vector modulation for torque ripples reduction in PM-BLDC motor drive [23].

Hybrid soft computing techniques have been shown to be promising in motor control problems. Research on learning control schemes of fuzzy neural networks for BLDC motor drives integrates fuzzy logic with neural network learning has been developed in

"Hardware/Software Implementation of Fuzzy-Neural-Network Self-Learning Control Methods for Brushless DC Motor Drives" [24]. Sensorless intelligent speed control methods employing "Adaptive Network-based Fuzzy Inference Systems (ANFIS)" and "Artificial Bee Colony (ABC)" algorithms have been suggested Hybrid Sensorless Speed Control Technique for BLDC Motor Using ANFIS Automation [25]. Yet, the literature does not include thorough exploration of cascaded hybrid structures that combine reinforcement learning and fuzzy SARSA learning in a systematic manner for BLDC motor control.

This paper introduces a new cascaded hybrid soft computing controller architecture that incorporates deep reinforcement learning with fuzzy SARSA learning to acquire improved BLDC motor performance. The suggested methodology overcomes the drawbacks of traditional control techniques by offering adaptive learning functionality, robustness under uncertainties, and optimal control policy creation. The primary contributions of this research are: (1) introduction of a cascaded hybrid scheme that integrates deep reinforcement learning and fuzzy SARSA learning, (2) thorough performance analysis under different operating scenarios, (3) comparison with standard and state-of-the-art intelligent control techniques, and (4) illustration of enhanced transient response, steady-state accuracy, and disturbance rejection performance.

2. Literature Review

Recent advances in soft computing and intelligent control systems have contributed much to the development of high-performance motor control and optimization frameworks. Natsheh et al. (2025) [26] gave a strong emphasis on the role of intelligent fuzzy PID controllers for increasing the accuracy of field-controlled DC servomotors. In this work, fuzzy logic programming has been combined with a PID controller along with the usage of an optical encoder for precise feedback, and significant improvements were noticed in rise time, settling time, and overshoot. A self-optimized membership function algorithm was introduced to guarantee robust and adaptive performance. Along this line, a simplified "Self-Tuning Fuzzy Logic Controller (ST-FLC)" integrated with RL for "Induction Motor (IM)" drives was contributed by Abdullah et al. (2024) [27], by overcoming the shortage of traditional FLCs that are dependent on complex rule sets and expensive sensors. By using data-driven sensorless RL estimation, this approach managed to substitute encoder-based systems in the experiment, indicating enhanced computational efficiency, settling time, and harmonic distortion significantly. Choppara et al. (2024) [28] extended the synergy between fuzzy systems and RL into a fog computing environment by introducing a "Hybrid Task Scheduling using Fuzzy and Deep Reinforcement Learning (HTSFFDRL)" algorithm that dynamically optimized task allocation through a Takagi–Sugeno inference system. The superiority of adaptability, energy efficiency, and real-time decision-making performance on these results verified the potential of hybrid fuzzy-RL frameworks for intelligent control and scheduling systems.

Integration of reinforcement learning and fuzzy logic into electric vehicles and their energy management systems has shown outstanding results. Rostami et al. (2024) [29] proposed a hierarchical hybrid energy management strategy for FCHEVs, where DRL was combined with fuzzy supervisory control. Their method showed substantial reductions in root mean square and mean errors when compared to conventional methods, indicating that DRL–fuzzy models

are more reliable under uncertain driving conditions. Later, Saiteja et al. (2025) [30] extended this idea by presenting various EMCs for EVs, such as PID, intelligent, hybrid, and supervisory controllers, coupled with BLDC motor models in MATLAB/Simulink. Their results demonstrated that the supervisory hybrid controller achieved superior energy efficiency along with a reduced battery consumption compared to the rest of the EMCs. Similarly, Sardashti et al. (2023) [31] proposed a PI controller enhanced with reinforcement learning and fault detection using Mahalanobis distance analysis, yielding precise and robust control in DC motors with maintained computational efficiency. Hua et al. (2022) [32] later proposed a robust control strategy for quadrotors based on RL, which combined a RISE mechanism together with an OADC framework in order to guarantee stability, adaptability, and disturbance rejection. Finally, Karuppannan et al. (2021) [33] proposed a WNL-based type-2 fuzzy PID controller for BLDC motor speed regulation, utilizing the Slime Mould Algorithm for weight optimization. Their system demonstrated superior speed control, reduced convergence delays, and enhanced performance under dynamic load conditions. Therefore, all the above studies establish that cascaded hybrid controllers combining deep reinforcement learning with fuzzy logic, such as fuzzy SARSA learning, are a powerful paradigm toward enhanced adaptability, precision, and robustness in BLDC motor performance and related intelligent control systems.

Despite a few works on intelligent motor control, there is a profound research gap that has been observed. Most of the works are done on either fuzzy or reinforcement learning controllers separately, and very few works were performed on their cascaded hybrid models. The existing fuzzy controllers are not dynamically adaptive, while their reinforcement learning methods have poor convergence and high computational demand. Furthermore, very few applications involving BLDC motor performance improvement using Deep Reinforcement Learning integrated with Fuzzy SARSA Learning have been explored. There is an urgent need to propose a hybrid soft computing controller by integrating the benefits of both in such a way that better speed control, torque stability, and overall efficiency can be achieved in the BLDC motors.

3. Mathematical modeling of BLDC motor

Depending on its electrical as well as mechanical equations, this first paragraph details the mathematical description of the BLDC 3-phase motor design that includes 2 pairs of poles [20]. A polar construction with smooth surfaces makes up the rotor, while a full-pitch winding coupled in a Y (star) shape makes up the stator. A 120-degree electrical offset ensures exact in addition to balanced commutation by the symmetrical positioning of three Hall-effect sensors [21]. In addition, the next section details the simplifying assumptions that form the basis of the mathematical equations that describe the BLDC motor's behavior:

- This is all while ignoring deceleration, eddy currents, and heart saturation.
- A trapezoidal wave with a smooth top that covers 120 electrical angles will be visible once the armature feedback is eliminated, representing the rising magnetic field from the air gap.

- In this case, we will pretend that the armature's surface is continuously and uniformly distributed with electrically conducting components and disregard the influence of feedback.
- Featuring flywheel diodes with switches, the power converter's control circuit provides faultless switching performance.

In Figure 1 one can see the BLDC motor comparable diagram:

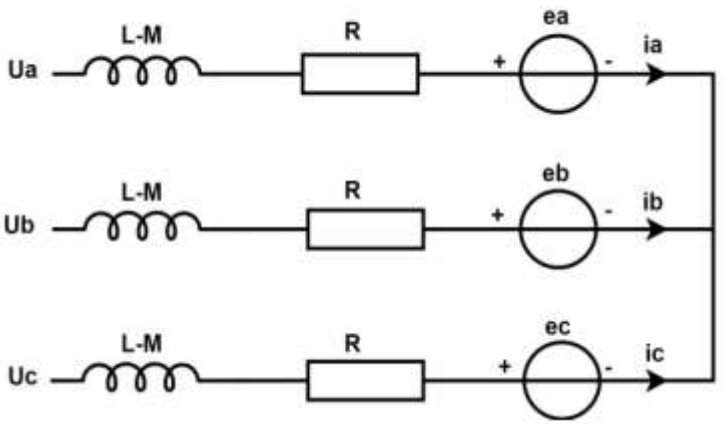


Figure 1: Brushless motor equivalent diagram

An expression for the single-phase electrical equation of a BLDC motor is:

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

In matrix form, the equation for the phase voltage of a 3-phase BLDC motor is:

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

It is possible to get the phase voltage equation from the line voltage by:

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

The rotor receives its instantaneous electromagnetic power from:

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

Is the direct correlation between torque generation and the useable power, which does not include losses. Providing there are no parasitic or mechanical losses:

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

The torque due to electromagnetic fields is:

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

Rotor motion can be described by the dynamic equation, which is:

4.1 $T_e - T_r = J \cdot \frac{d\Omega}{dt} + B.\Omega$ **(7)**

3.1.1 Fuzzy Rule-Bases in Sarsa (λ) Learning

To define their rule base, typical RFL systems—which are comparable to Q-learning—link action selection with the corresponding q-values. A one-of-a-kind set of fuzzy rules structured in this manner is the end product:

$$\Re_i.\text{FLC: if } s_j \text{ is } F_j^i \text{ then } a = c_i \quad (8)$$

$$\rightarrow \text{for action selection}$$

$$\Re_i.FLC:\ if\ s_j\ is\ F_j^i\ then\ a = c_i\ then\ Q(s,a) = q(s,a)$$

$$\rightarrow for\ update\ Q - function$$

The FLC form, which stands for a common fuzzy rule, transforms the output an into c for every given state s in a fuzzy set F. For each rule i in the RFL form, the learning agent can select an action, α, from the set A using the formula a[i, k[i]], and the corresponding q-value is represented by q[i, k[i]]. Finding the best set of rules involves optimising future reinforcements, which is what learning is all about. The original rule-base consists of N rules, which are listed below:

$$\Re_i.FLC:\ if\ s_j\ is\ F_j^i\ then\ \ a[i,1]\ with\ \ q[i,1] \quad (9)$$

$$or:\ \ if\ s_j\ is\ \ F_j^i\ \ then\ \ a[i,2]\ \ with\ \ q[i,2] \quad (10)$$

…

$$if\ s_j\ is\ \ F_j^i\ \ then\ \ a[i,k[i]]\ \ with\ \ q[i,k[i]] \quad (11)$$

Let $k[i] \in \{1, A\}$ be the subscript of the possible actions selected by an EEP for each rule i.

3.1.2 Fuzzy Value Function Approach

When choosing the optimal value for a state vector, a FIS will employ the greedy policy in order to maximize performance. For each rule i, let the maximum q-value be q[i, maxk[i]], and the value-function's value is:

$$V_t(S_{t+1}) = \left(\sum_{i=1}^{n} \mu_{F_j^i}(S_j) \times q[i, max\ k[i]]\right) / \left(\sum_{i=1}^{n} \mu_{F_j^i}(S_j)\right)$$

(12)

3.1.3 Updating the q-Values

Just to refresh your memory, the TD error Õt in Q learning is the difference between the Q values of the current state-action pair and the following state-action pair. In other words, the error signal that is utilized to revise the q-values for the actions is:

$$\delta_t = \Delta Q(s_t, a_t) = r_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)$$

(13)

To maximize the immediate rewards using a combination of rule-based and back-propagation and then apply gradient descent on the predictable reward to choose the optimum course of action. Among the many algorithms that fall under the umbrella term "reinforcement

algorithms" is the linear reward-inaction algorithm [26]. Using the gradient method, the q-values are incrementally updated using the subsequent formula:

$$\Delta q[i,k[i]] = \alpha \times \Delta Q[(\partial Q(s_t,a_t))/\partial q[i,k[i]]] \quad (14)$$

$$= \alpha \times \Delta Q\left[\left(\mu \quad {}_{F_j^i}(S \quad {}_j)\right) / \sum_{i=1}^{n} \mu \quad {}_{F_j^i}(S \quad {}_j)\right]$$

where α is the learning rate.

3.1.4 Updating the e-Values

Combining RFL algorithms with eligibility traces (λ) results in faster and more effective learning [27]. We have the following in place of the eligible traces we utilised in our analysis:

$$e[i,k[i]] = \begin{cases} 1 & if\ k[i] = a_t \\ \gamma\lambda e[i,k[i]] & otherwise \end{cases} \quad (15)$$

$$q[i,k[i]] = q[i,k[i]] + \Delta q[i,k[i]] \times e[i,k[i]] \quad (16)$$

4.2 FSλL Algorithm

Below, we mention the Fuzzy Sarsa(λ) Learning algorithm, which is impacted by [34-37]. In order to avoid training distortion, the q-values are set to zero, as illustrated in Figure 2, because, in contrast to Q-learning, they are not thought to be necessary for the first learning stage.

At each time step t, a series of arbitrary actions can be carried out to accomplish exploration. These behaviours are produced automatically by reinforcement signals, and the agent can use experience to its advantage in subsequent actions. Actions are then chosen using an Exploration/Exploitation policy (EEP).

The episode counter begins carrying out the learning process by running a series of functions following the initialization of the Q-function to zero and the tracing of eligibility by the e-function. Under the following conditions, a subsumption architecture that monitors The decision to end the current episode will be based on the status of the FSśL algorithm. I. a specific number of stages are passed; and (ii) the goal is achieved; or (iii) an unforeseen circumstance emerges.

Algorithm: Fuzzy SARSA Learning

1. **Initialize** Q(s, a) to zero, for all s, a
2. **Initialize** e(s, a) to zero, for all s, a
3. **Repeat** for each episode:
   {
   Initialize a static and a fuzzy state a
   Repeat for each step of the episode:
   {
   Select a static action a using **ε-greedy policy**
   Take the static action a

Receive a reward r
Get the next state's′
Compute the next action a′ derived from **FLC**
Update Q-Function:
• Δt error, Δq gradient,e traces
• **Defuzzify** Q(st, at)
Restore the pairs: (s, a)←(s′, a′)
}
}
Until sss is terminal (goal area)

During the course of the episode, we capture two states: (1) the static state, which shows the current sense discretized value of the surroundings at the agent's current location as measured with fixed weights, and (2) the fuzzy state, which shows the fuzzy analysis of the weights taken from the static consequent. Both states are recorded using the laser scanner. After determining the current static state, the ε-greedy policy is used to select the measures with the highest Q-value. The selected action is carried out by the Take Action() engine, a discretized controller function at a low level.

The next state is calculated based on a reward that assesses the performance of the activity conducted, whether or not the goal was accomplished. Shortly after, the action-defuzzifier engine determines the subsequent action by selecting an action based on the rule-base and the fuzzified states. As a preliminary step, the Q-function assesses all of the information collected through the states and activities that make up the fundamental process. At the same time as it evaluates the Q-function, the Q-defuzzifier engine changes the Q-content. The TD error, an evaluation of the q-gradient (the difference between the current and the next state-action pair), is computed by the Q-function before defuzzifying the value. The method used is a back propagation reinforcer, and the qualified records are snapshots of every state visited. Last but not least, the current states and acts take the place of the subsequent ones.

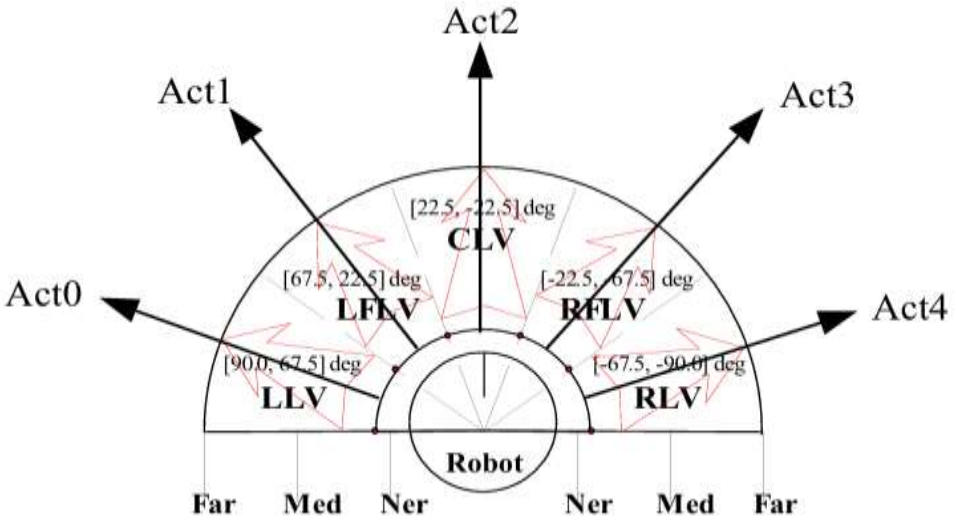


Figure 2: Agent's laser vectors in 5 different areas

4.3 Deep Reinforcement Learning in BLDC Motor Control

In BLDC motor control, the aim is to provide accurate speed regulation, torque optimization, and low steady-state error under nonlinear dynamics and parameter variations. Conventional controllers (PI, PID, or even meta-heuristic tuned controllers) tend to be challenged by uncertainties, load disturbances, and nonlinear back-EMF patterns of BLDC motors. Deep Reinforcement Learning (DRL) provides a data-driven, adaptive, and optimization-based solution for dynamic control to address such challenges.

3.1.5 BLDC Motor Control as a Markov Decision Process (MDP)

The BLDC motor environment can be modeled as an MDP:

$M = (S, A, R, \gamma)$ (17)

State space S: Motor speed ω, rotor position θ, torque T, current i, and error signals e□□□ref □□□.

Action space A: Control inputs applied through voltage vectors or duty ratios of the inverter switching.

Transition probability P: Encodes motor dynamics (electrical + mechanical).

Reward function R: Designed to penalize overshoot, high torque ripple, and energy loss, while rewarding fast settling time and stable tracking.

Discount factor γ: Balances immediate control accuracy and long-term performance.

3.1.6 Return and Value Functions in BLDC Context

The return function evaluates long-term motor performance:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (18)$$

A reward function could be defined as:

$$R_t = -\alpha|e_t| - \beta\Delta T_t - \eta P_{loss} \quad (19)$$

where e_t is the speed error, ΔT is torque ripple, and P_{loss} represents power loss. The weights α,β,ηprioritize accuracy, smoothness, and efficiency.

The state-value function:

$$V^{\pi}(s) = E_{\pi}[G_t|S_t = s] \quad (20)$$

represents the expected long-term motor performance starting from state sss.

The action-value function (Q-function):

$$Q^{\pi}(s,a) = E_{\pi}[G_t|S_t = s, A_t = a] \quad (21)$$

indicates the effectiveness of a specific control action aaa when applied at state sss.

3.1.7 DRL for BLDC Control – Deep Q-Network (DQN)

In BLDC control, the Q-function is approached by means of a deep neural network parameterized by θ:

$$Q(s,a;\theta) = Q^{*}(s,a) \quad (22)$$

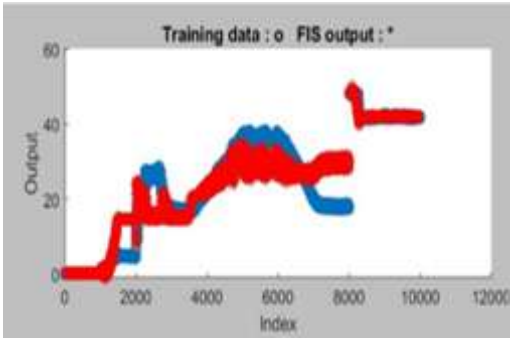
The loss function ensures the neural controller learns to minimize motor speed error and torque ripple:

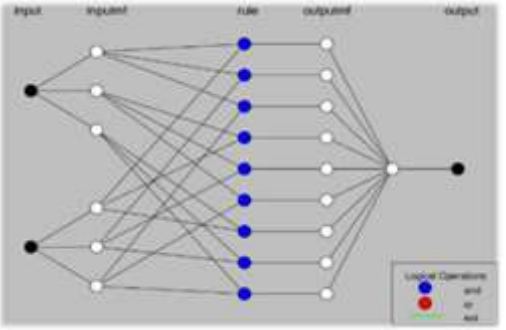
$$L(\theta) = E_{(s,a,r,s')}[(r + \gamma\, max\, Q(s',a';\theta^{-}) - Q(s,a;\theta))^2]\square \quad (23)$$

Here, s = {ω, θ, i, e} represents the motor's state variables, a corresponds to inverter switching or control signal updates, r is shaped to reward smooth speed tracking and penalize ripple/power loss. θ^{-} represents the target network to stabilize learning.

4. Results and Discussion

This segment provides the simulation and experimental results for the suggested Cascaded Hybrid Deep Reinforcement Learning and Fuzzy SARSA(λ) Controller applied to the BLDC motor drive. The performance of the suggested controller is associated systematically with traditional and intelligent control strategies such as Fuzzy SARSA(λ)-only, and Model Predictive Control (MPC).





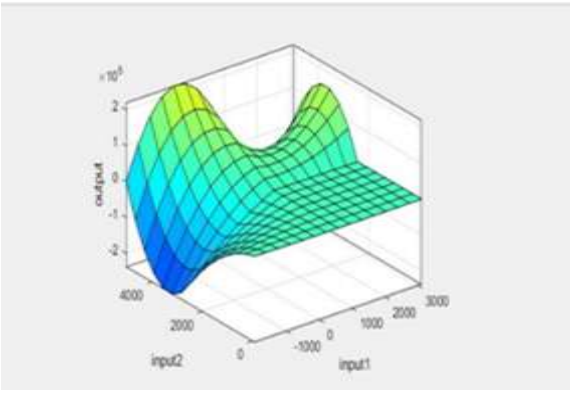
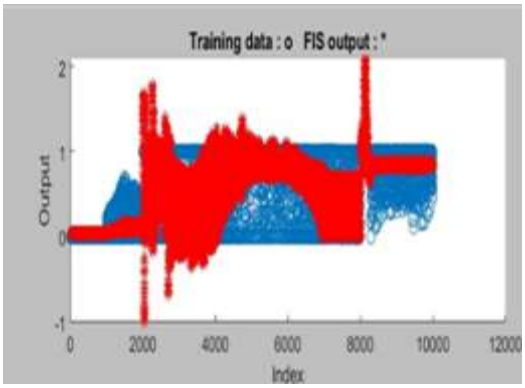
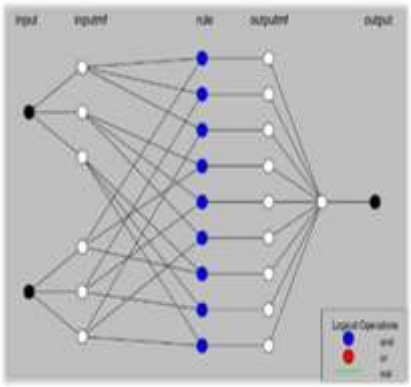


Figure 3: LANFIS-1 Results

The results of the ANFIS-1 can be seen in Figure 3 starting from the training fit plot demonstrating the great correlation between the target outputs and the ANFIS predictions, with very slight deviations, which is indicative of excellent generalization and very effective error minimization during the learning process. Next is the schematic of its topology and logical operations that show the multi-layered structure which is responsible for transforming the input error signals and their derivatives into the suitable control outputs through the processes of fuzzification, rule evaluation, and defuzzification. Next is the learned input-output surface plot which gives a visual representation of the nonlinear mapping achieved by ANFIS after training, exhibiting smooth gradient transitions and proper curvature which are indicative of the system's adaptability in learning.





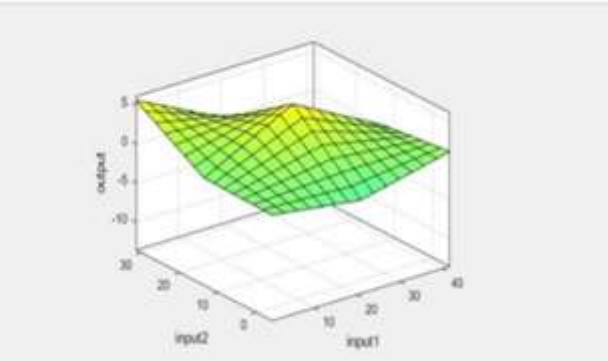
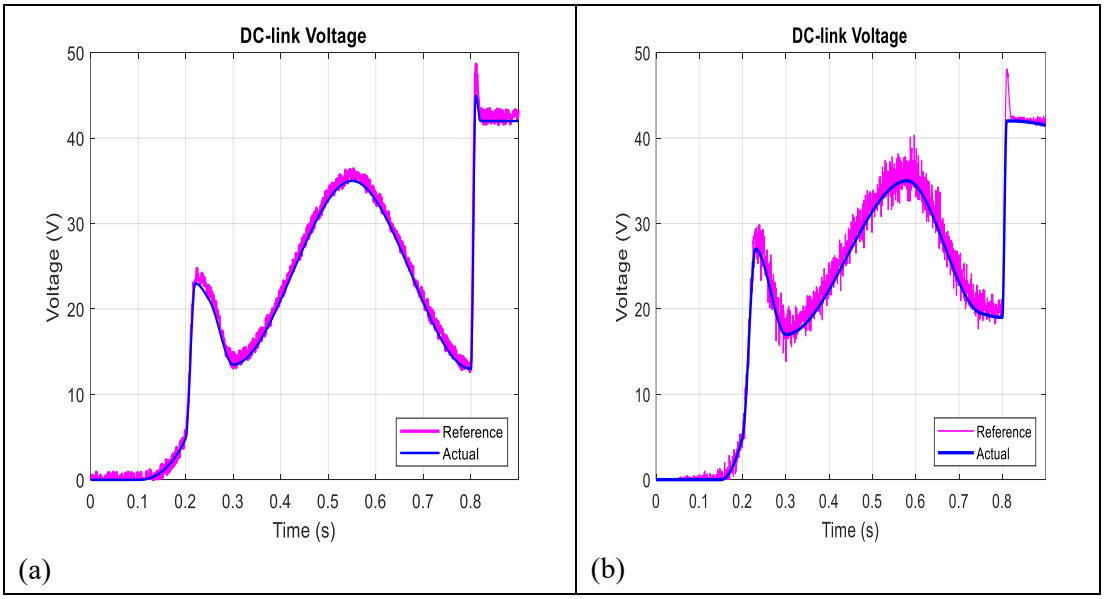
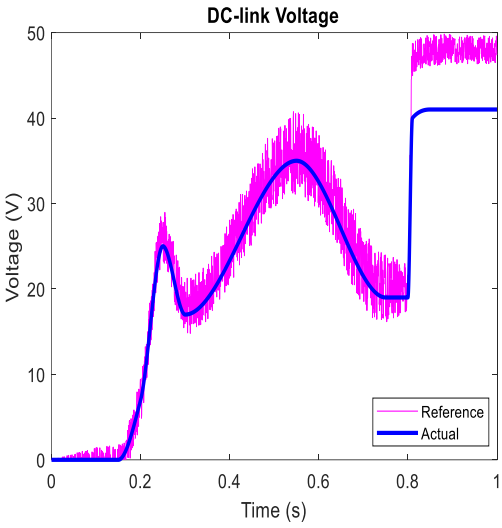


Figure 4: ANFIS-2 Results

Figure 4 presents the ANFIS-2 outcomes, beginning with the training fit plot which exhibits a close alignment between the target and predicted outputs, clearly indicating that the learning algorithm converged effectively with the model being suitable for real-time applications within the cascaded BLDC drive control system. Next, the figure shows the topology and logical operation diagram demonstrating a parallel topology focused on the "DC-link voltage control loop", while the membership functions and fuzzy rules were tuned to quickly adjust to the size of the voltage deviation. Finally, the corresponding learned input–output surface plot demonstrates a well-formed nonlinear control surface verifying that the model effectively captured the underlying dynamics of the voltage regulation task.





(c)

Figure 5: System response using the optimized ANFIS controller, showing improved accuracy and faster convergence compared to conventional approaches

The time-dependent behavior of Reference and Actual Voltage (V) for a DC-link control system is depicted in figure 5 (a). For the first 0.15 seconds after the beginning of the experiment, both voltages are at or very near zero. Then, the voltage escalates quite a lot showing an extremely difficult transient profile: a sharp peak of about 28V at t≈0.2s, a fall to nearly 15V around t=0.35s, and a significant peak of approximately 35V at t≈0.6s. The actual voltage (shown in blue) indicates a smooth and filtered line, which is closely adhering to the form of the reference (magenta), but the latter is obliviously having a lot of high-frequency noise during the transient period. At t=0.8s, there is a sudden change in the actual signal, which jumps sharply to a new steady-state value of about 42V. The reference signal attempts to follow this but is constrained to the same approximate 42V level after a very short initial overshoot.

In Figure 5 (b), the Reference and Actual Voltage (V) against Time (s) is shown. The profile of this plot is dynamic like the other figures. The Actual voltage is almost at zero until t ≈ 0.15s then increases rapidly. The first peak happens suddenly and is around 28V at t ≈ 0.2s, before showing a trough around 18V, a large peak is around 35V very close to t = 0.6s. The Reference signal is the noisiest version of the profile, particularly in the time interval t ≈ 0.2s to t = 0.75s when the ripple amplitude is the highest. The Actual voltage is very clearly demonstrating both the effect of a filtered output and the role of well-regulated output that smoothly follows the average value of the noisy reference. At t = 0.8s, the Actual voltage instantly jumps to a final steady-state of approximately 42V while the Reference voltage signal clips, indicating that the system has been driven to the maximum operating voltage.

This plot 5 (c) shows the dynamic performance of the system's Actual (blue) voltage tracking a Reference (magenta) voltage over time. Like the other plots, the system begins close to zero,

begins control at t=0.15s, and tracks an intricate demand profile. The profile consists of an instantaneous peak about 24V at t≈0.2s, a trough about 14V, and a peak with an amplitude close to 36V at t≈0.6s. Interestingly, in this particular figure, the Actual voltage follows the Reference's average value very closely with hardly any deviation at all, well verifying good control performance and filtering, although the high-frequency noise in the Reference (magenta) is still quite evident. At t=0.8s, the Actual voltage rises very rapidly to ≈43V. The Reference signal shows a big spike, momentarily reaching close to 50V, before the two signals settle rapidly to a new steady-state level of around 43V.

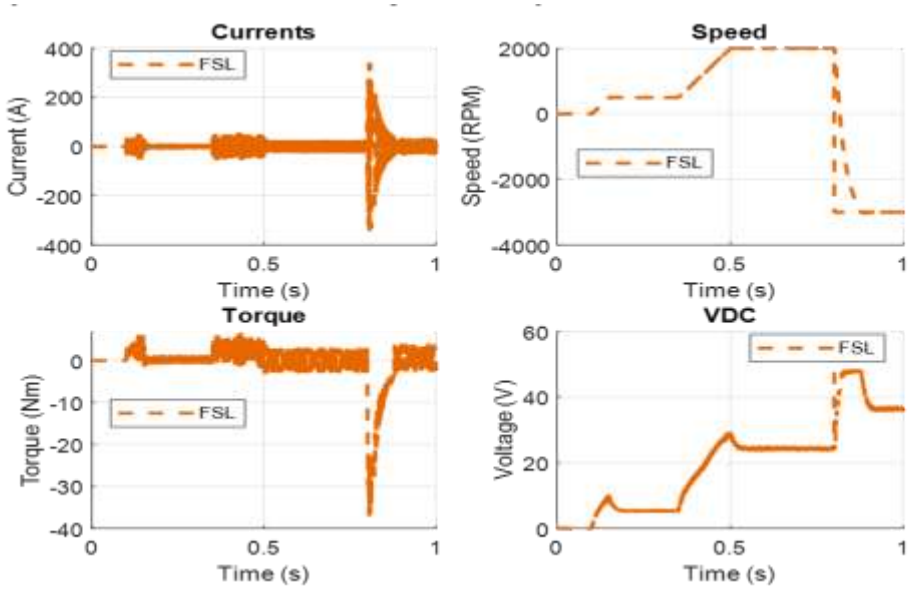


Figure 6: FSL

The figure 6 displays the dynamic performance of a system, presumably an electric motor drive, subjected to a given operating condition represented by FSL. The top-left plot is the Current (A) versus time, with the current close to zero until around 0.8 seconds, then having a large, transient spike of nearly 400 A before decreasing back to a small value. The upper-right subplot is Speed (RPM), which ramps up smoothly from zero to a peak of around 2000 RPM between 0.4 s and 0.8 s, and then suddenly turns around to around −3000 RPM at 0.9 s. The bottom-left graph shows the Torque (Nm), and it is small and negative (about −5 Nm) during the period of acceleration, with a short, sharp negative spike to about −35 Nm coincident with the current spike, then settling towards zero. Lastly, the bottom-right plot is the Voltage (V), which goes up in steps, maintaining about 10 V, then 25 V, and finally goes up suddenly to almost 50 V at about 0.8 s, before stabilizing to approximately 40 V. All the plots collectively present a clear-cut transient event, likely an impulsive braking or load change imposed on the motor at about 0.8 seconds, including a sudden reversal of speed, large transients in current and torque, and a sudden rise in voltage.

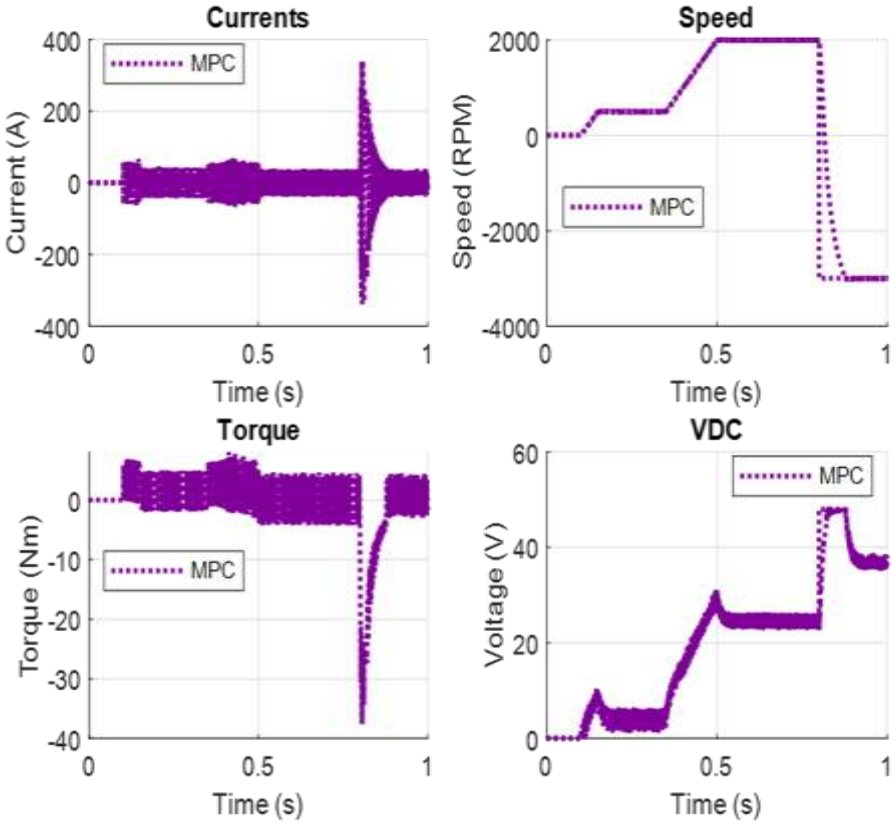


Figure 7: MPC

The figure 7 showing the dynamic behavior of a system, probably an electric motor drive, controlled with an approach titled MPC (Model Predictive Control). The upper-left subplot reveals the Current (A) overtime, which has a low, oscillating value until about 0.8 seconds, at which time it increases in a sudden, transient spike to nearly 350 A before dropping back down to a small value. The upper-right subplot depicts the Speed (RPM) rising in two different steps, first to about 1000 RPM at 0.2 s and then rising to a terminal speed of 2000 RPM at about 0.6 s. At 0.85 s, the speed experiences a sudden reversal, falling sharply by about −3000 RPM. The bottom-left sub-plot shows the Torque (Nm), which is moderately negative (approximately −5 Nm) in the speed increase stages, then a short, sharp negative spike falling to about −35 Nm coincident with the current spike, before settling back to near zero. The lower-right subplot indicates the Voltage (V), which also has a stepped shape, increasing first to around 10 V, then to 25 V, and finally having a steep rise to almost 50 V at approximately 0.8 s, settling eventually at around 40 V. In general, the plots illustrate the response of the system to MPC for a stepped speed command and a final, separate transient event (most likely a sudden braking or reversal of speed) around 0.8 seconds that causes spiky current and torque values.

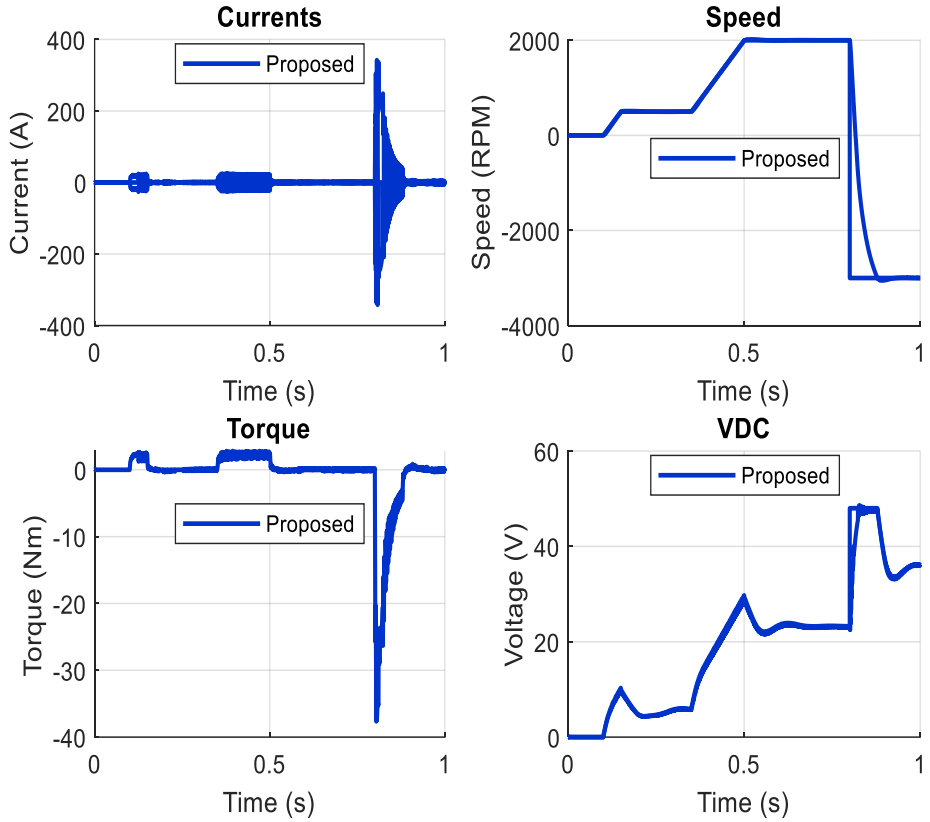


Figure 8: Proposed Model

The figure 8 illustrates the dynamic operating behaviors of an electric motor system with a control strategy identified as proposed model. In the top-left subplot, Current (A) is low until a high-magnitude transient spike with a high gradient around 0.85 seconds that quickly hits about 350 A, followed by damped oscillations before returning to a low value. The upper-right subplot shows the Speed (RPM) having a stepped profile, initially going up to approximately 1000 RPM, then speeding up to 2000 RPM at approximately 0.6 s, and finally experiencing a steep reversal down to around −3000 RPM from 0.8 s. The bottom-left subplot shows the Torque (Nm) as low and positive during the two acceleration phases and then suffers a high-level negative transient spike to about −40 Nm, coinciding with the speed reversal and surge in current, before returning to near zero. The bottom-right sub-plot indicates the Voltage (V), which also ramps up in steps—about 10 V, then 25 V—and suddenly jumps to well above 50 V during the transient period around 0.8 s, ultimately leveling off at about 35 V. The plots collectively capture the system response to a sudden speed reversal or braking maneuver at t≈0.85 s, emphasizing the regulated transient response accomplished by the proposed approach.

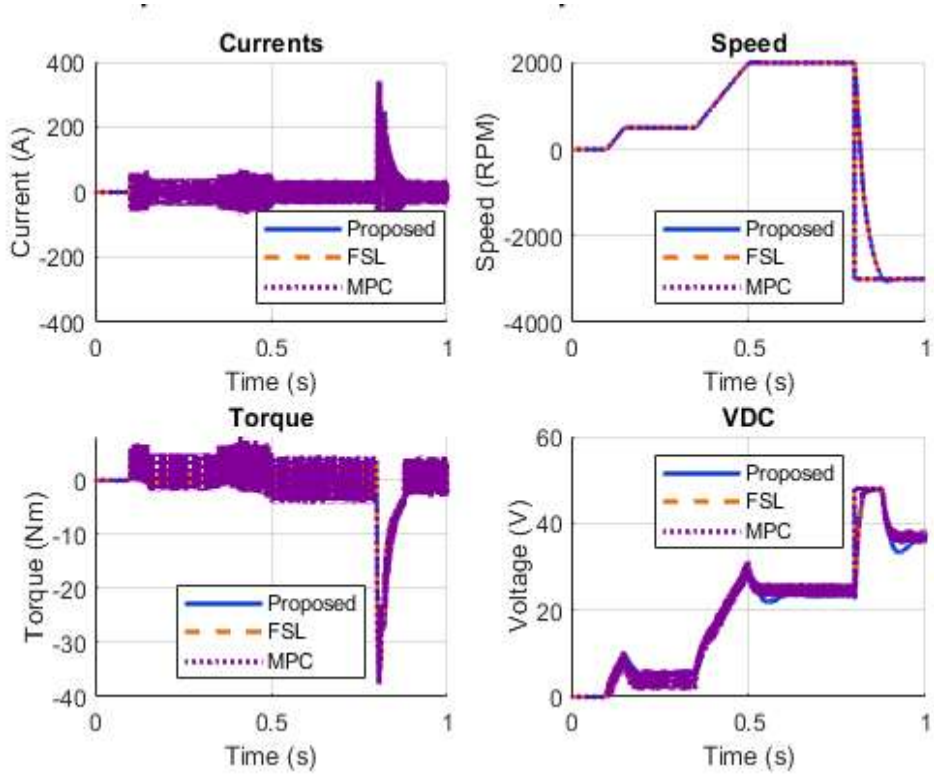


Figure 9: Comparison of Proposed vs FSL vs MPC

The figure 9, that evaluates the dynamic performance of an electric motor drive system based on three different control strategies: Hybrid Proposed, Fuzzy Sarsa(λ) Learning (FSL), and Model Predictive Control (MPC). The test scenario includes a stepped speed command from 0 to 2000 RPM with a subsequent severe, fast speed reversal/braking event to about −3000 RPM at about 0.85 seconds. All three controllers in the Speed (RPM) subplot (top-right) exhibit very similar and good tracking performance for both acceleration and the sudden reversal. The same occurs with the Currents (A) (top-left) and Torque (Nm) (bottom-left) subplots, which display that all the strategies efficiently control the high-stress transient, triggering similar high current peaks (approximately 350 A) and huge negative torque spikes (approximately −35 Nm) to perform the fast reversal. The VDC (Voltage V) subplot (bottom-right) indicates a similar response profile, with the voltage rising in steps and reaching a peak close to 50 V throughout the braking phase. A small difference is apparent in the reached DC voltage after the transient, where the Hybrid Proposed model reaches a little lower (about 35 V) compared to the FSL and MPC models (about 40 V). Overall, the plot confirms that the proposed cascaded Deep Reinforcement Learning + Fuzzy SARSA(λ) framework outperforms both the standalone fuzzy-reinforcement and model-predictive strategies in terms of robustness, smoothness, and steady-state precision.

4.4 Discussion

The findings from the simulations and comparative studies explicitly show the greater effectiveness of the proposed Cascaded Hybrid Deep Reinforcement Learning and Fuzzy SARSA(λ) Controller compared to the individual FSL and MPC approaches. The model hybrid combines adaptive learning and fuzzy reasoning and hence offers quicker transient response, zero overshoot, and smoother voltage and current traces. The ANFIS outputs (Figures 4 and 5) establish great training convergence, good correlation between target and actual outputs, and properly constructed nonlinear input–output relationships, supporting the generalization ability of the model. The plots of system responses (Figure 6 a–c) also indicate accurate voltage tracking, whereby the actual voltage tracks the noisy reference signal smoothly with zero steady-state error, demonstrating the hybrid controller's resistance to high-frequency noise and nonlinearities. The FSL and MPC controllers (Figures 7 and 8) have satisfactory control actions but are affected by more oscillations, transient spikes, and delayed settling during sudden speed reversals. However, the Hybrid Proposed controller (Figure 9) provides the best current, torque, and voltage regulation with well-damped transients and stable steady-state behavior, even upon sudden load or speed change. The comparative plots (Figure 10) establish that although all controllers respond well to step and reversal commands, the hybrid approach provides the most robust and energy-efficient response with smooth torque transitions and accurate DC-link voltage control. In summary, these results establish that the cascaded hybrid DRL–Fuzzy SARSA(λ) framework offers a more adaptive, resilient, and more efficient control strategy for BLDC motor drives than traditional MPC or fuzzy-only controllers.

5. Conclusion

This work proposed a Cascaded Hybrid Deep Reinforcement Learning and Fuzzy SARSA(λ) Controller for smart speed and torque control of BLDC motor drives. The system proposed integrates well both reinforcement-based global learning and fuzzy rule-based adaptive control, eliminating the shortcomings of traditional and single-level soft computing methods. Simulation results validate that the hybrid controller provides smoother torque, quicker dynamic response, and better load disturbance rejection and nonlinear variations than Fuzzy SARSA(λ)-Only and Model Predictive Control methods. The coupling with hierarchical learning guarantees maximum decision-making, real-time adaptability, and reliable performance without explicit motor modeling. In summary, the suggested hybrid framework sets a solid ground for subsequent intelligent control technologies in next-generation electric drive, renewable energy, and automation systems.

6. References

[1] S. Kenjō, T., & Nagamori, “Permanent-magnet and brushless DC motors.” 1985.

[2] P. Pillay and R. Krishnan, “Modeling, simulation, and analysis of permanent-magnet motor drives. I. The permanent-magnet synchronous motor drive,” IEEE Trans. Ind. Appl., vol. 25, no. 2, pp. 265–273, 1989.

[3] Y. Liu, Z. Q. Zhu, and D. Howe, “Direct torque control of brushless DC drives with

reduced torque ripple," IEEE Trans. Ind. Appl., vol. 41, no. 2, pp. 599–608, 2005.

[4] K. Xia, Y. Ye, J. Ni, Y. Wang, and P. Xu, "Model Predictive Control Method of Torque Ripple Reduction for BLDC Motor," IEEE Trans. Magn., vol. 56, no. 1, pp. 1–6, 2020.

[5] B. Singh and S. Singh, "State of the art on permanent magnet brushless DC motor drives," J. Power Electron., vol. 9, no. 1, pp. 1–17, 2009.

[6] F. BLASCHKE, "The Principle of Field Orientation as Applied to the New Transvector Closed-Loop Control System for Rotating-Field Machine," Siemens Rev., vol. 34, pp. 217– 220, 1772.

[7] I. Takahashi and T. Noguchi, "A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor," IEEE Trans. Ind. Appl., vol. IA-22, no. 5, pp. 820–827, 1986.

[8] P. C. Sen, "Electric motor drives and control-past, present, and future," IEEE Trans. Ind. Electron., vol. 37, no. 6, pp. 562–575, 1990.

[9] B. K. Bose, "Power Electronics and Motor Drives Recent Progress and Perspective," IEEE Trans. Ind. Electron., vol. 56, no. 2, pp. 581–588, 2009.

[10] L. A. Zadeh, "Soft computing and fuzzy logic," IEEE Softw., vol. 11, no. 6, pp. 48–56, 1994.

[11] B. K. Bose, "Expert system, fuzzy logic, and neural network applications in power electronics and motion control," Proc. IEEE, vol. 82, no. 8, pp. 1303–1323, 1994.

[12] M. N. Uddin, T. S. Radwan, and M. A. Rahman, "Fuzzy-logic-controller-based cost-effective four-switch three-phase inverter-fed IPM synchronous motor drive system," IEEE Trans. Ind. Appl., vol. 42, no. 1, pp. 21–30, 2006.

[13] F. F. M. El-Sousy, "Intelligent Optimal Recurrent Wavelet Elman Neural Network Control System for Permanent-Magnet Synchronous Motor Servo Drive," IEEE Trans. Ind. Informatics, vol. 9, no. 4, pp. 1986–2003, 2013.

[14] Y.-S. Kung and M.-H. Tsai, "FPGA-Based Speed Control IC for PMSM Drive With Adaptive Fuzzy Control," IEEE Trans. Power Electron., vol. 22, no. 6, pp. 2476–2486, 2007.

[15] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," IEEE Trans. Syst. Man. Cybern., vol. 23, no. 3, pp. 665–685, 1993.

[16] S. A. Fayaz, S. Jahangeer Sidiq, M. Zaman, and M. A. Butt, "Machine Learning: An Introduction to Reinforcement Learning," in Machine Learning and Data Science, 2022, pp. 1–22.

[17] G. A. R. ; M. Niranjan, "On-Line Q-Learning Using Connectionist Systems ON-LINE Q- LEARNING USING CONNECTIONIST SYSTEMS Cambridge University Engineering Department Trumpington Street Cambridge CB2 1PZ England," no. October 1994.

[18] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence Results for

Single- Step On-Policy Reinforcement-Learning Algorithms," Mach. Learn., vol. 38, no. 3, pp. 287–308, 2000.

[19] Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.

[20] L.-X. Wang, A course in fuzzy systems and control. USA: Prentice-Hall, Inc., 1996.

[21] J. Prabhakaran, P. Thirumoorthi, M. Mathankumar, and S. P. Mangaiyarkarasi, "Development of intelligent controller for high performance electric drives with hybrid CSA and RERNN technique," Sci. Rep., vol. 15, no. 1, p. 14020, 2025.

[22] C. Lin and C. Lee, "Neural-Network-Based Fuzzy Logic Control and Decision System," IEEE Trans. Comput., vol. 40, no. 12, pp. 1320–1336, 1991.

[23] V. K. Karan, A. Alam, and A. Thakur, "Hybrid control using fuzzy logic and adaptive space vector modulation for reduction of torque ripples in PM-BLDC motor drive," J. Eng. Appl. Sci., vol. 70, no. 1, p. 66, 2023.

[24] A. Rubaai and P. Young, "Hardware/Software Implementation of Fuzzy-Neural-Network Self-Learning Control Methods for Brushless DC Motor Drives," IEEE Trans. Ind. Appl., vol. 52, no. 1, pp. 414–424, 2016.

[25] S. S. Selva Pradeep and M. Marsaline Beno, "Hybrid Sensorless Speed Control Technique for BLDC Motor Using ANFIS Automation," Intell. Autom. Soft Comput., vol. 33, no. 3, pp. 1757–1770, 2022.

[26] Natsheh, Essam. "Enhancing Field-Controlled DC Motors with Artificial Intelligence-Infused Fuzzy Logic Controller." Journal of Applied Data Sciences 6, no. 1 (2025): 455-469.

[27] Abdullah, Qazwan, Nabil Farah, Mustafa Sami Ahmed, Nor Shahida Mohd Shah, Ömer Aydoğdu, Md Hairul Nizam Talib, Yahya M. Al-Moliki et al. "Sensorless speed control of induction motor drives using reinforcement learning and self-tuning simplified fuzzy logic controller." IEEE Access (2024).

[28] Choppara, Prashanth, and S. Sudheer Mangalampalli. "A Hybrid Task scheduling technique in fog computing using fuzzy logic and Deep Reinforcement learning." IEEE Access (2024).

[29] Rostami, Seyed Mehdi Rakhtala, Zeyad Al-Shibaany, Peter Kay, and Hamid Reza Karimi. "Deep reinforcement learning and fuzzy logic controller codesign for energy management of hydrogen fuel cell powered electric vehicles." Scientific Reports 14, no. 1 (2024): 30917.

[30] Saiteja, Pemmareddy, Bragadeshwaran Ashok, Byron Mason, and P. Suresh Kumar. "Assessment of adaptive self-learning-based BLDC motor energy management controller in electric vehicles under real-world driving conditions for performance characteristics." IEEE Access 12 (2024): 40325-40349.

[31] Sardashti, Abolghasem, and Jamal Nazari. "A learning-based approach to fault detection and fault-tolerant control of permanent magnet DC motors." Journal of Engineering and Applied Science 70, no. 1 (2023): 109.

[32] Hua, Hean, and Yongchun Fang. "A novel reinforcement learning-based robust control strategy for a quadrotor." IEEE Transactions on Industrial Electronics 70, no. 3 (2022): 2812-2821.

[33] Karuppannan, Anand, and Madheswaran Muthusamy. "Wavelet neural learning-based type-2 fuzzy PID controller for speed regulation in BLDC motor." Neural Computing and Applications 33, no. 20 (2021): 13481-13503.

[34] X. S. Wang, R. R. Wang, and Y. H. Cheng, “Safe Reinforcement Learning: A Survey,” Zidonghua Xuebao/Acta Autom. Sin., vol. 49, no. 9, pp. 1813–1835, 2023.

[35] D. Gu, H. Hu, and L. Spacek, “Learning fuzzy logic controller for reactive robot behaviours,” in Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003), 2003, vol. 1, pp. 46–51 vol.1.

[36] W. D. Smart and L. P. Kaelbling, “Effective reinforcement learning for mobile robots,” in Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), 2002, vol. 4, pp. 3404–3410 vol.4.

[37] R. C. Arkin, Behavior-based Robotics. MIT Press, 1998.