DOI: https://doi.org/10.62985/j.huit_ojs.vol26.no2E.400

AN ADAPTIVE NEURAL PI CONTROL FRAMEWORK USING DEEP REINFORCEMENT LEARNING FOR HIGH-PERFORMANCE PMSM SPEED DRIVES

Do Trung Khanh Cong[1], Vo Ngoc Vinh[2,*], Nguyen Minh Tam[1], Le Tien Loc[2]

[1]*Ho Chi Minh City University of Technical Education,*

No. 1 Vo Van Ngan, Thu Duc Ward, Ho Chi Minh City

[2]*Lac Hong University,*

No. 10 Huynh Van Nghe, Tran Bien Ward, Dong Nai Province, Vietnam

*Email: *vongocvinh@lhu.edu.vn*

Received: 21 January 2026; Revised: 11 March 2026; Accepted: 19 April 2026

ABSTRACT

High-performance PMSM speed control remains challenging due to parameter uncertainties, nonlinear dynamics, and load disturbances. Although conventional PI controllers are widely adopted, their fixed gains limit adaptability under varying operating conditions. To overcome this, this paper proposes an adaptive DRL-based neural PI control framework for PMSM drives. The method integrates a DDPG algorithm with an RBF neural network for continuous online gain tuning. Furthermore, an anti-windup mechanism and a reference model–based augmentation are incorporated to ensure closed-loop stability under actuator saturation. The learning objective is formulated to optimize tracking accuracy, transient performance, and disturbance rejection. Simulation results demonstrate that the proposed framework significantly outperforms fixed-gain PI controllers, minimizing settling time, overshoot, and steady-state error under ideal physical constraints. These findings confirm the framework's robustness and efficiency for advanced motor drives.

Keywords: Permanent magnet synchronous motor, adaptive PI control, deep reinforcement learning, DDPG, neural networks.

1. INTRODUCTION

Permanent magnet synchronous motors (PMSMs) have become a dominant choice in high-performance electric drive applications due to their high efficiency, high power density, and fast dynamic response [1], [2]. These advantages make PMSMs particularly suitable for modern applications such as electric vehicles, renewable energy systems, and precision industrial automation. To fully exploit these benefits, advanced control strategies capable of ensuring accurate speed tracking and robust performance under varying operating conditions are required.

Field-oriented control (FOC) is widely applied in PMSM drive systems because it enables the decoupling of torque and flux components by transforming stator quantities into a rotating *dq* reference frame [3]. This decoupling allows PMSMs to exhibit control characteristics similar to separately excited DC motors, thereby simplifying controller design. In practical FOC implementations, proportional–integral (PI) controllers are commonly used in both speed and current control loops owing to their simple structure and ease of industrial implementation [4].

Despite their widespread application, conventional PI controllers with fixed gains suffer from performance degradation when system parameters vary or external disturbances occur [5]. Variations in load torque and magnetic saturation can lead to increased overshoot and longer settling times. Numerous tuning techniques have been proposed to mitigate these issues, including heuristic rules and intelligent optimization algorithms [6]. However, most existing approaches rely on offline tuning, which limits their effectiveness in real-time applications with continuously changing dynamics [7].

Recent advancements in artificial intelligence have paved the way for more sophisticated control strategies. Specifically, deep reinforcement learning (DRL) has been successfully applied to motor drives to handle non-linearity and parameter uncertainties [9], [11]. Several studies have explored the synergy between Deep Deterministic Policy Gradient (DDPG) and traditional PI structures to enhance dynamic performance [8], [10]. Furthermore, theoretical frameworks integrating Lyapunov stability with neural controllers have addressed the long-standing issue of convergence in adaptive systems [12-13].

The integration of deep neural networks into RL frameworks has further enabled applications in continuous and high-dimensional control problems [14]. Nevertheless, directly applying DRL as a standalone controller remains challenging due to concerns related to real-time computational constraints and the perspective of intelligent control in 2025 [15]. To address these challenges, embedding learning algorithms within conventional PI controllers allows the preservation of stability and interpretability while enhancing adaptability. However, only limited attention has been given to continuous online tuning of PI gains during operation while simultaneously integrating stability-enhancing mechanisms such as anti-windup compensation.

Motivated by these observations, this paper proposes an adaptive neural PI control framework based on DRL for high-performance PMSM speed drives. A DDPG algorithm combined with a radial basis function (RBF) neural network is employed to continuously tune PI gains online. An anti-windup mechanism and a reference model–based augmentation are incorporated to ensure stable operation under actuator saturation and dynamic load disturbances.

The main contributions of this paper can be summarized as follows:

- An adaptive neural PI control framework integrating deep reinforcement learning with conventional PI control for PMSM speed regulation.
- A DDPG–RBF-based online gain tuning strategy that enables continuous adaptation without requiring an accurate PMSM model [10].
- The incorporation of anti-windup compensation and reference model–based augmentation to enhance robustness and stability.
- Theoretical consideration of stability using a Lyapunov-based approach to ensure the boundedness of tracking errors [12].
- Simulation results demonstrating improved transient performance compared with a conventional fixed-gain PI controller.

The remainder of this paper is organized as follows. Section 2 presents the mathematical model of the PMSM. Section 3 describes the proposed adaptive DDPG–RBF–PI control scheme. Section 4 analyzes the convergence conditions of the system. Section 5 discusses the simulation results, and finally, Section 6 concludes the paper.

2. FOC MODEL FOR PMSM DRIVE

2.1. PMSM Dynamic Model in dq Axis

The dynamic model of the permanent magnet synchronous motor (PMSM) is expressed in the rotating dq reference frame using the following equations [1]:

$$\begin{cases} v_d = R_s\, i_d + L_d \frac{di_d}{dt} + \omega_e\, L_q\, i_q \\ v_q = R_s\, i_q + L_q \frac{di_q}{dt} + \omega_e\, (L_q\, i_q + \psi_f) \end{cases} \quad (1)$$

The electromagnetic torque generated by the PMSM is calculated as

$$T_e = \frac{3}{2} p[\psi_f\ i_q + (L_d - L_q) i_d\, i_q] \quad (2)$$

Where v_d and v_q are the d- and q- axis stator voltages, i_d, i_q are the d- and q- axis stator currents, R_s is the stator resistance, L_d, L_q are the dq- axis inductances, ψ_f denotes the permanent magnet flux linkage, ω_e is the electrical angular speed and p is the number of pole pairs.

The PMSM model in the dq reference frame is implemented in MATLAB/Simulink using the parameters summarized in Table 1:

Table 1. System parameters

No.	Parameters	Value
1	Stator resistance R_s	1.3Ω
2	Inductance $L_d = L_q$	8.3e-3 H
3	Link flux ψ_f	0.175 Wb
4	Inertia J	0.0027 kg.m²
5	Coefficient of friction B	0.0004924 Nm/rad
6	Pole pair p	4

2.2. FOC Control Principle

The Field-Oriented Control (FOC) method transforms the three-phase stator currents (i_A, i_B, i_C) into the synchronous dq reference frame using Clarke and Park transformations.

This allows the decoupling of the stator currents into two independent components the d-axis current i_d, which regulates the magnetic flux, and the q-axis current i_q, which controls the electromagnetic torque.

By enforcing i_d = 0, the torque production is maximized, and the PMSM behaves similarly to a DC motor with linear torque characteristics [1]. Under this condition, the electromagnetic torque can be expressed as:

$$T_e = \frac{3}{2} p\, \psi_f\, i_q \quad (3)$$

The FOC control structure adopted in this study consists of three main stages:

1. Current Measurement and Transformation: The stator currents (i_A, i_B, i_C)are measured and transformed into dq components (i_d, i_q).

2. Current Control: Two independent PI controllers regulate i_d and i_q to generate the voltage references (v_d, v_q).

3. Voltage Synthesis: The dq voltages are converted back to three-phase quantities via inverse transformations, and a space vector PWM (SVPWM) scheme is used to generate the inverter switching signals.

The overall FOC structure is illustrated in Fig. 1 [2].

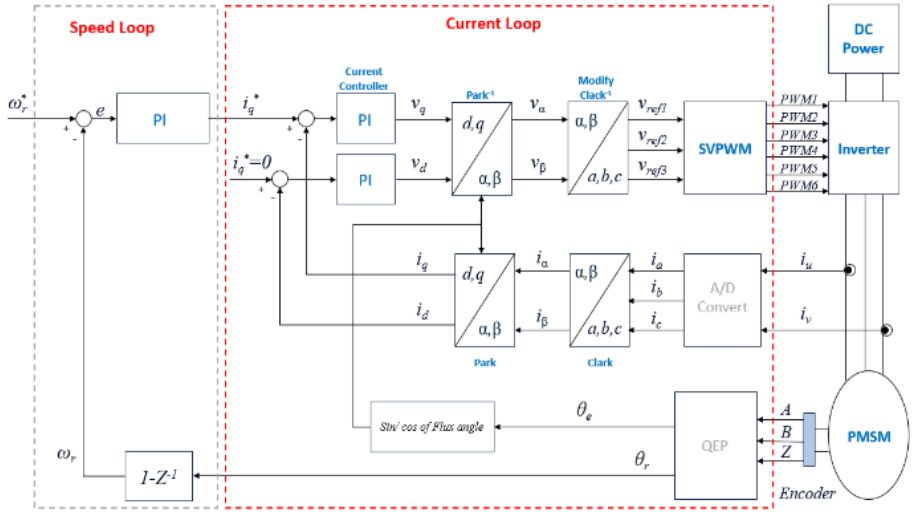

Fig.1. FOC model [2]

2.3. PI controller

The proportional–integral (PI) controller is one of the widely used control strategies in industrial applications due to its simple structure, robustness, and ease of implementation. In PMSM speed control systems employing field-oriented control (FOC), PI controllers are commonly applied to independently regulate the d-axis and q-axis currents, enabling effective control of magnetic flux and electromagnetic torque.

The control output of a PI controller is defined as [3]:

$$u(t) = K_p e(t) + K_i \int e(t)\, dt \quad (4)$$

Where $u(t)$ control signal (voltage reference for inverter), $e(t)$ denotes the control error between the reference and measured values,K_p and K_i are the proportional and integral gains, respectively. The proportional term provides a fast response to instantaneous errors, while the integral term eliminates steady-state error and improves tracking accuracy.

In the FOC scheme, two PI controllers are employed to regulate the d–q current components independently. The d-axis controller is responsible for flux regulation through i_d, while the q-axis controller determines the torque by controlling i_q. The resulting voltage commands (v_d, v_q) are converted back to three-phase signals via inverse transformations and applied to the inverter.

In practice, PI controllers with fixed gains tend to lose performance when system parameters vary or when the operating condition becomes highly nonlinear. This limitation motivates the use of adaptive tuning approaches, where learning-based methods such as deep reinforcement learning can be incorporated to improve robustness and dynamic response.

3. PROPOSED DDPG-RBF-PI CONTROLLER

The proposed DDPG–RBF–PI controller integrates a Deep Deterministic Policy Gradient (DDPG) algorithm with a radial basis function (RBF) neural network to adaptively tune the PI

gains (K_P, K_i) in real time. Unlike conventional PI controllers with fixed parameters, the proposed method enables continuous online adaptation to system uncertainties, nonlinearities, and load disturbances.

The overall control architecture is shown in Fig. 2. The system operates in a multi-rate structure: the speed control loop runs at 1 kHz, the current control loop at 8 kHz, and the PWM generation at 16 kHz. The DDPG–RBF–PI controller is embedded in the speed loop to update the PI gains online, while the inner current loops ensure fast dynamic response and accurate torque control.

Fig. 2 illustrates the structural integration of the learning-based controller within the FOC framework, including the dq current loops, coordinate transformations, and SVPWM module. The operational workflow of the learning and adaptation process is presented in Fig. 3.

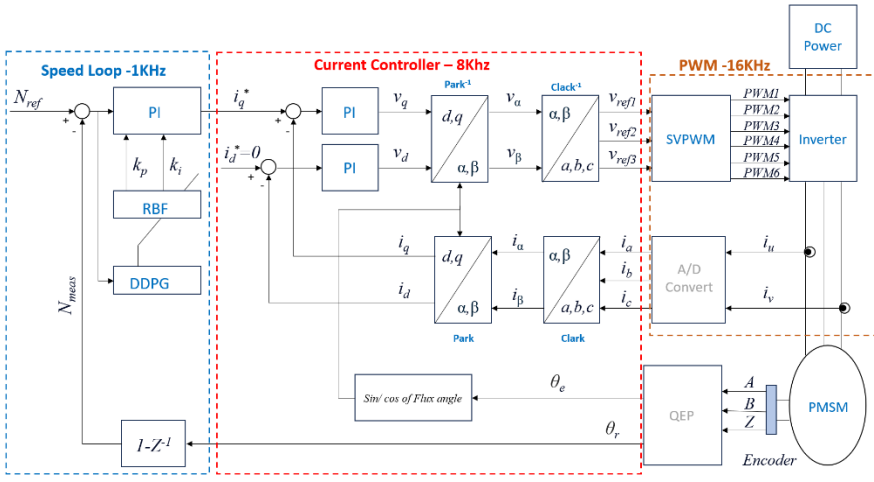

Fig.2. DDPG-RBF-PI controller model

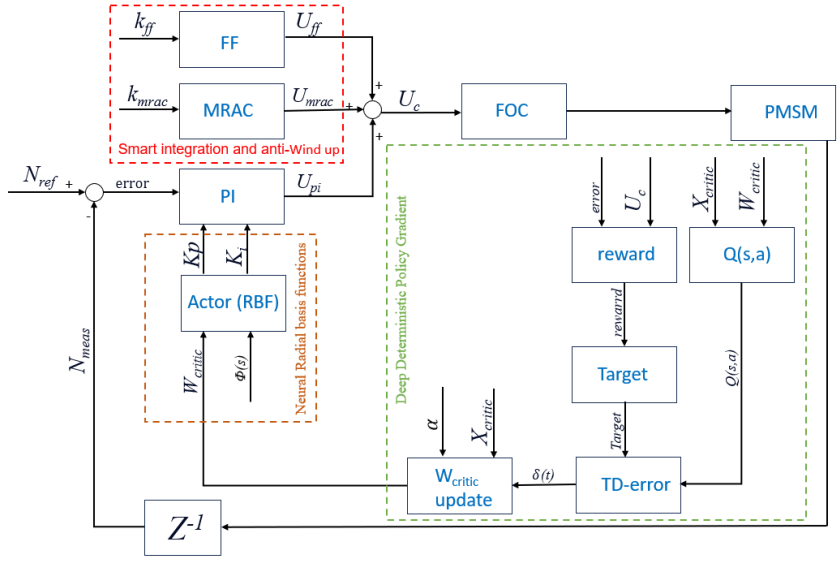

Fig.3. DDPG-RBF-PI's workflow

The proposed strategy integrates the following key components:

- Actor (RBF network): generates adaptive PI gains (K_p, K_i) based on current system states.
- Critic: evaluates the action-value function Q (s, a) and provides policy gradients for

Actor updates.

- Reward and TD-error computation ensures stable learning through temporal difference updates.
- Additional enhancements: feedforward compensation, MRAC-style augmentation, and anti-windup mechanisms to improve robustness and transient performance.

This architecture allows real-time online tuning of PI gains, resulting in improved convergence speed, disturbance rejection, and reduced steady-state error.

3.1. Neural Network-Based PI Controller

The proposed controller employs a feedforward RBF neural network to adaptively estimate the optimal PI gains (*Kp*, *Ki*) in real time. The input vector consists of the filtered reference speed, filtered measured speed, and filtered speed error, normalized to improve learning stability:

$$x = [\frac{Nref\_filt}{4000}, \frac{Nmeas\_filt}{4000}, \frac{e_{filt}}{1500}] \tag{5}$$

This normalization ensures consistent scaling of input signals and enhances learning efficiency.

3.2. Smart Integration and Anti-Windup Mechanism

To enhance robustness under actuator saturation and prevent integrator windup, a smart anti-windup mechanism is incorporated. A back-calculation scheme is employed to remove excess integral action when saturation occurs [4]:

$$I \leftarrow I - \beta(u_{clamp} - u_{unclamp}) \tag{6}$$

Where $\beta = 0.2$

$u_{clamp} = sat(u_{unclamp}; u_{min}, u_{max})$ and $u_{unclamp} = u_{PI} + u_{ff} + u_{MRAC}$

To reduce rise time and alleviate the PI controller burden, a feedforward compensation term derived from the reference speed is introduced [4], [5]:

$$u_{ff}(t) = k_{ff}\ \mathrm{Nref_{filt}} \tag{7}$$

$$u_{ff} = sat(u_{ff}(t), -u_{ff\_max}, u_{ff\_max}) \tag{8}$$

Where k_{ff} is selected sufficiently small to preserve closed-loop stability margins. This term compensates the predictable torque demand, allowing the PI controller to focus on disturbances and model uncertainties.

To improve transient robustness under rapid load variations, an MRAC-inspired compensation based on filtered error dynamics is applied [4], [5]:

$$e(t) = N_{\mathrm{ref\_filt}} - N_{meas\_filt} \tag{9}$$

$$u_{MRAC}(t) = k_{MRAC}\ \dot{e_f}(t) \tag{10}$$

Where $\dot{e_f}(t)$ is obtained through a first-order low-pass filter to attenuate noise.

The integral state is updated continuously with a leakage term to avoid drift during prolonged saturation:

$$I \leftarrow (1 - \vartheta)I + e_{filt}\ dt \tag{11}$$

With $\vartheta \in [0, 0.02]$. To further prevent excessive accumulation, a dynamic integral bound is enforced:

$$|I| < I_{max}(e) = I_0 + \mu|e_{filt}| \tag{12}$$

The final unclamped control law is expressed as:

$$u_{unclamp} = K_p\, e_{filt} + K_i\, I + u_{ff} + u_{MRAC} \tag{13}$$

This integrated strategy achieves effective anti-windup and improved transient performance without requiring explicit integral enable/disable logic.

3.3. Deep Deterministic Policy Gradient–Based Control Strategy

The proposed adaptive controller is developed using the Deep Deterministic Policy Gradient (DDPG) framework, which is well suited for continuous action spaces such as online PI gain tuning. The controller interacts continuously with the PMSM environment to learn an optimal policy.

3.3.1. RBF-Based Actor Network

The Actor employs an RBF neural network to generate adaptive PI gains (K_p, K_i) from the system state $s = [N_{ref}, N_{meas}, e_{filt}]$. Each RBF neuron uses a Gaussian activation function [6]:

$$\varphi_i(s) = exp(\frac{(s-c_i)^2}{2.\delta_i^2}), \qquad \text{i=1, 2, 3, ..., N} \tag{14}$$

The Actor output is computed as:

$$o = W_{actor}\, \varphi(s) \tag{15}$$

and normalized using sigmoid functions:

$$s_{1=\delta(o_1)},\ s_{2=\delta(o_2)} \tag{16}$$

The adaptive PI gains are obtained as:

$$K_p = K_{p\_min} + \left(K_{p\_max} - K_{p\_min}\right) s_1 \tag{17}$$

$$K_i = K_{i\_min} + \left(K_{i\_max} - K_{i\_min}\right) s_2 \tag{18}$$

3.3.2. Critic Network and Learning Process

The Critic network evaluates the action-value function $Q(s, a)$ using the concatenated state-action vector [6-8] [14]:

$$Q_{(s,a)} = W_{Critc}\, X_{\boldsymbol{Critic}} \tag{19}$$

To improve learning effectiveness and ensure physically meaningful control behavior, the reward function is formulated as the negative of a composite cost function that incorporates tracking accuracy, transient performance, control effort, and constraint handling.

The instantaneous reward is defined as [6–8]:

$$r(t) = -J(t) \tag{20}$$

where the total cost function J(t) is expressed as:

$$J(t) = J_e + J_{OS} + J_u + J_b + J_I - J_{reward} \tag{21}$$

Where:

$J_e = \omega_e |e(t)|$ Tracking error term

$J_{OS} = 180.OS(t) + 2200.OS_n^2(t)$ Overshoot penalty

$J_u = 0.9|u(t)|$ Control effort penalty

$J_I = 0.35 \int_0^t |e(\tau)| d\tau$ Integral error term, this component penalizes accumulated tracking error and improves steady-state accuracy.

$$J_b = \begin{cases} 4500\left(\frac{OS}{E_{max}}\right)^2, OS > 1.5 \\ 0, otherwise \end{cases} + \begin{cases} 4500\left(\frac{OS}{E_{max}}\right)^2, OS > 6 \\ 0, otherwise \end{cases} + \begin{cases} 1500, |u| > 0.9|u_{\max}| \\ 0, otherwise \end{cases}$$

This barrier function enforces physical constraints by penalizing excessive overshoot and near-saturation control signals.

$$J_{reward} = \begin{cases} 30, |e| < 5, OS < 1 \\ 10, |e| < 15, OS < 2 \\ 0, otherwise \end{cases}$$

A reward shaping mechanism is introduced to encourage precise tracking under small error conditions. This term provides additional incentives for achieving high-accuracy steady-state performance [12-13].

To further enhance stability and provide a physically interpretable learning objective, the reward function is extended by incorporating a Lyapunov-based energy term.

This formulation enables the reinforcement learning agent to explicitly minimize the system energy and enforce stability-aware adaptation of the PI gains.

A Lyapunov candidate function is defined based on the normalized system states as[12-13]:

$$V(t) = 0.5e_n^2(t) + 0.07I_n^2(t) + 0.035\dot{e}_n^2(t) + 0.018u_n^2(t) + 0.1e_n^4(t) + 0.18OS_n^2(t) \quad (22)$$

where $e_n = \frac{e}{E_{max}}, I_n, \dot{e}_n, u_n$: denote the normalized tracking error, integral state, error derivative, and control signal, respectively. This function represents the generalized energy of the closed-loop system.

To evaluate the stability behavior during learning, a Lyapunov dissipation index is introduced [12-13]:

$$D_V(t) = \frac{V(t) - V(t - T_s)}{Ts} - \lambda_V . V(t) \quad (23)$$

where $\lambda_V > 0$ is a design parameter. A negative value of $D_V(t)$ indicates that the system dissipates energy and satisfies a practical exponential stability condition.

The reward function(20) is formulated as the negative of a composite cost function[13]:

$$r(t) = -(J_e + J_{OS} + J_u + J_b + J_I + J_V + J_C - J_{reward}) \quad (24)$$

Where $J_V = \begin{cases} 1500D_V(t), & D_V(t) > 0 \\ 10D_V(t), D_V(t) \le 0 \end{cases}$ This term penalizes energy increase and encourages energy dissipation, thereby enforcing stability during the learning process.

$J_C = 500\hat{V}(t) + 300|V_{td}(t)|$ where $\hat{V}(t)$ is the Lyapunov function estimated by the Critic network and $V_{td}(t)$ is the temporal difference error. This term improves learning consistency.

The temporal-difference (TD) error is defined as:

$$\delta(t) = r(t) - Q_{(s,a)} \quad (25)$$

The Critic weights are updated via gradient descent:

$$W_{critic} \leftarrow W_{critic} + \alpha\delta(t)X_{critic}^T \quad (26)$$

3.3.3. Actor Network Update

The Actor network generates the adaptive PI gains (K_p, K_i). and is updated using the deterministic policy gradient provided by the Critic. Since the Critic is modeled as a linear

function of the state–action vector, the gradient of the action-value function with respect to the action $a = [K_p, K_i]^T$ can be directly obtained from the corresponding Critic weights [14]:

$$\frac{\delta Q\ (s,a)}{\delta a} = \begin{bmatrix} \frac{\delta Q}{\delta K_p} \\ \frac{\delta Q}{\delta K_i} \end{bmatrix} \tag{27}$$

For the RBF-based Actor, the network output is computed as:

$$o_i = W_{actor}^{(i,:)}\ \varphi(s), i \in \{p, i\} \tag{28}$$

followed by a sigmoid normalization. The Actor weights are updated according to[14]:

$$W_{actor}^{(i,j)} \leftarrow W_{actor}^{(i,j)} + \alpha_a \frac{\delta Q}{\delta a_i}\left(K_{i\_max} - K_{i\_min}\right)\sigma_i.\left(1 - \sigma_i\right)\ \varphi_i \tag{29}$$

This update enables smooth and stable online adaptation of the PI gains, allowing the controller to maximize the expected long-term reward and improve PMSM speed regulation performance.

4. CONVERGENCE CONDITIONS

The convergence of the proposed DDPG-RBF-PI controller is ensured when the Temporal Difference (TD) error gradually approaches zero [6-8][14]:

$$\delta(t) = \ target(t) + \ \gamma\ Q_{(s',a')} - Q_{(s,a)} \tag{30}$$

When δ(t) →0, the Critic accurately approximates the action-value function Q(s,a), which is a necessary condition for the Actor to converge to optimal PI gains.

To guarantee closed-loop stability and avoid actuator saturation, the adaptive PI gains are constrained within predefined bounds:

$$K_{p\_min} < K_p < K_{p\_max} \tag{31}$$

$$K_{i\_min} < K_i < K_{i\_max} \tag{31}$$

Furthermore, stable learning requires the Critic network to approximate a Lipschitz-continuous action-value function [6], [8]:

$$\| \ Q_{(s_1,a_1)} - \ Q_{(s_2,a_2)} \| \leq \ L \ \| \ (s_1, a_1) - (s_2, a_2) \ \| \tag{33}$$

Under these conditions, the proposed controller achieves stable and bounded online adaptation.

5. SIMULATION RESULT

Simulation studies are conducted in MATLAB/Simulink using the FOC-based PMSM model with parameters listed in Table I. The control system operates at a 1 ms sampling period. The proposed DDPG-RBF-PI controller is benchmarked against a conventional PI controller tuned by the Ziegler–Nichols method (PI-ZN). Speed responses and the online evolution of PI gains are illustrated in Figs. 4-6.

5.1. Constant Speed Tracking (No-Load)

Under a constant reference speed profile without external load, both controllers are able to track the reference with acceptable accuracy, as shown in Fig. 4. The proposed DDPG–RBF–PI controller demonstrates a faster transient response, with a rise time of approximately 0.025s, compared to about 0.045s for the conventional PI-ZN controller.

In addition, the proposed method exhibits a smoother response with negligible overshoot, while the PI-ZN controller shows a small overshoot of around 1–2 rpm during certain transitions. Overall, the results indicate that the proposed controller can improve transient performance while maintaining stable and non-oscillatory behavior.

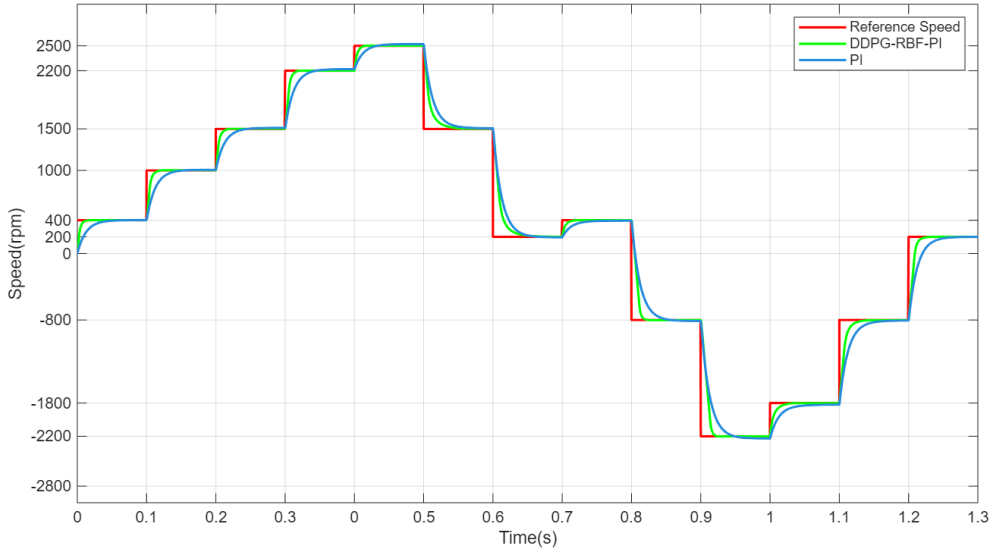

Fig.4. Speed response under no-load conditions

5.2. Sudden Load Disturbance

A step load torque of 12 Nm is applied at t = 0.5 s to evaluate disturbance rejection performance. As shown in Fig. 5, both controllers experience a speed drop at the disturbance instant. However, the proposed DDPG–RBF–PI controller exhibits a rapid recovery and maintains the speed close to the reference value with only a small transient deviation.

In contrast, the conventional PI controller shows a significantly larger speed drop and a much slower recovery, requiring a longer time to return toward the reference. This indicates that the proposed method provides improved disturbance rejection capability and better robustness under load variations.

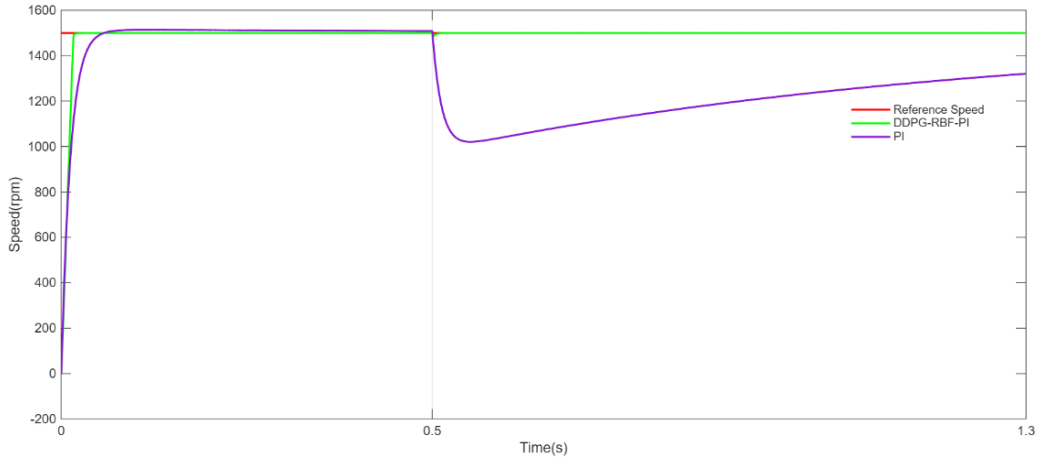

Fig. 5. Fast response to sudden load changes

To further evaluate the control performance, the dq-axis stator currents are analyzed, as shown in Fig. 6. The q-axis current i_q, which is directly related to torque production, exhibits rapid transient responses corresponding to speed changes and disturbance events.

Sharp peaks in i_q can be observed during acceleration and deceleration phases, indicating that the controller generates sufficient torque to achieve fast dynamic response.

In contrast, the d-axis current i_d remains close to zero throughout the operation, with only small transient deviations during abrupt changes. This confirms that the flux component is effectively regulated and the decoupling property of the FOC scheme is well maintained.

Overall, the results demonstrate that the proposed controller achieves fast torque response through i_q while preserving flux regulation via i_d, thereby ensuring stable and efficient FOC operation.

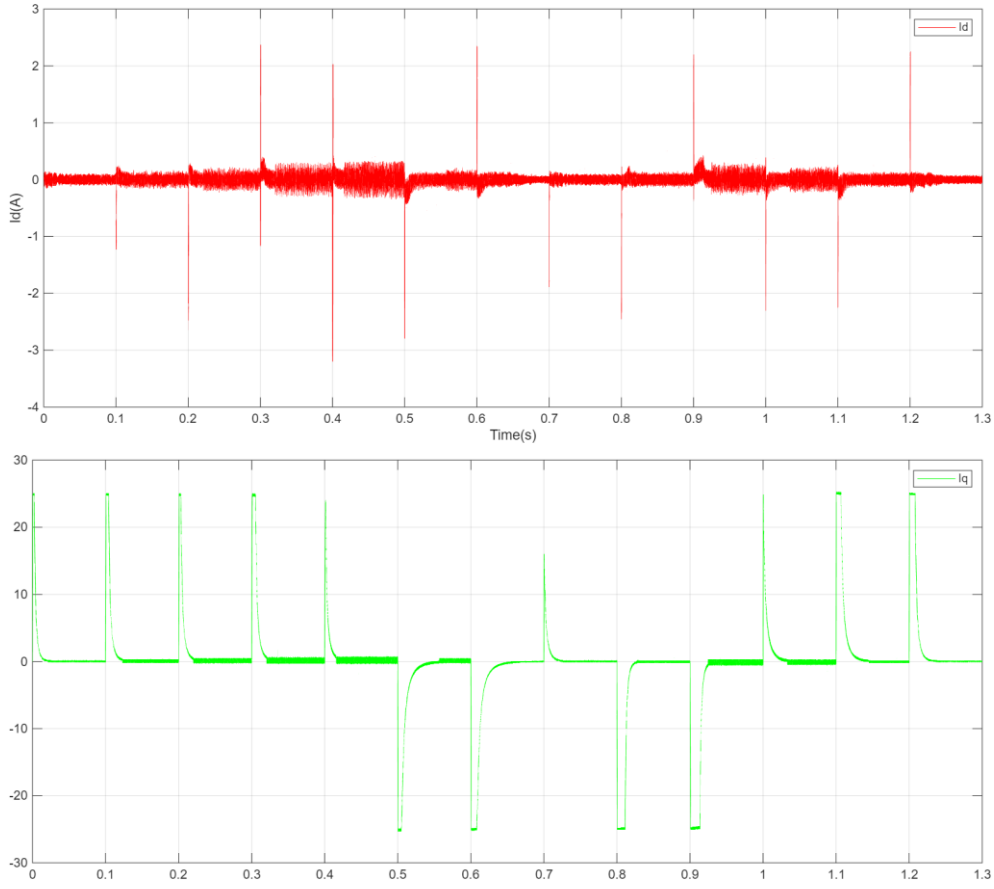

Fig.6. i_d, i_q value

5.3. Online Gain Adaptation

To further validate the stability characteristics of the proposed control strategy, the Lyapunov function V(t), its dissipation index $D_V(t)$, and the reward signal are analyzed, as shown in Fig.7. The Lyapunov function V(t) remains bounded and exhibits rapid decay after each transient, indicating that the system energy is effectively reduced following speed transitions and disturbance events. During steady-state operation, V(t) converges to a small value, confirming that the system operates close to equilibrium. The Lyapunov dissipation index $D_V(t)$is predominantly negative during the transient periods, which demonstrates that the system dissipates energy and satisfies the practical stability condition. Although short

positive spikes appear at the instants of abrupt reference changes or disturbances, they are immediately followed by negative values, indicating that the system quickly restores stability.

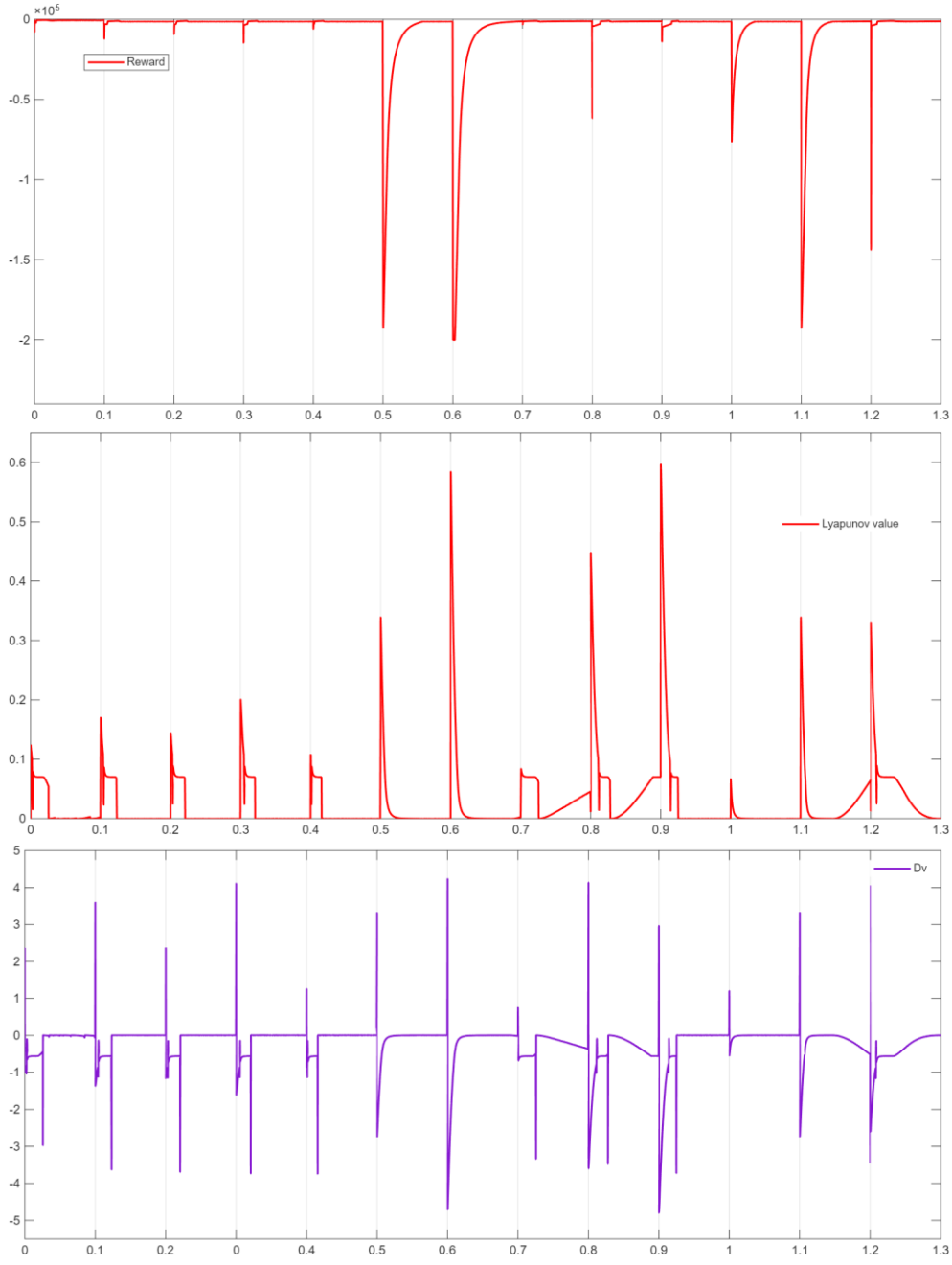

Fig.7. Reward and Lyapunov values

The reward signal reflects the same behavior, where large negative values occur during transient events due to tracking error and disturbance effects and gradually return to higher values as the system stabilizes. This behavior confirms that the learning process is aligned with both performance optimization and stability objectives.

Overall, the combined behavior of V(t), $D_V(t)$ and the reward signal provides strong evidence that the proposed controller ensures stable operation while maintaining effective dynamic performance.

Fig. 8 shows the real-time evolution of the adaptive gains K_p and K_i. During large transients, both gains increase to enhance responsiveness, then decrease once steady-state tracking is achieved, reducing control effort and noise sensitivity. This adaptive behavior enables consistent performance across operating conditions without manual returning, which is not possible with fixed-gain PI controllers.

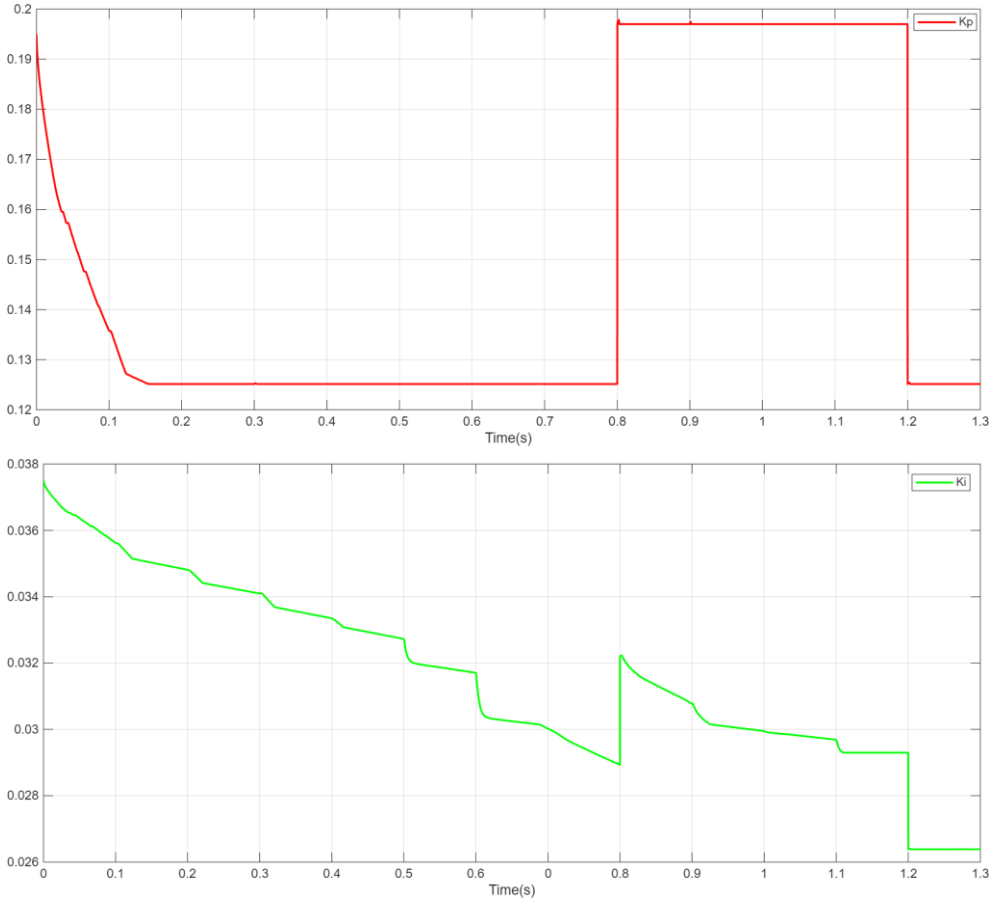

Fig.8. Real-time evolution of the rate increase K_p, K_i

Overall, the proposed controller consistently demonstrates faster transient response, superior disturbance rejection, and robust stability compared with the conventional PI-ZN controller, highlighting its suitability for PMSM drives operating under dynamic and uncertain conditions.

Qualitative ablation analysis indicates that removing individual components (feedforward, MRAC-style augmentation, or anti-windup) degrades performance in distinct ways, including slower rise time, increased peak deviation under load disturbances, and prolonged settling due to integral accumulation. These findings confirm the complementary roles of the proposed control components in achieving fast transient response, robust disturbance rejection, and stable operation under actuator saturation.

6. CONCLUSION

This work presented an adaptive DDPG–RBF–PI controller for PMSM speed control, combining an RBF-based actor, a TD-learning critic, and practical enhancements including feedforward compensation, MRAC-inspired adaptation, and anti-windup.

The controller updates the PI gains K_p, K_i online based on filtered system states, allowing it to adapt to parameter variations, modeling uncertainties and load disturbances without manual retuning.

Simulation results indicate that the proposed controller provides improved transient response and disturbance rejection compared to a conventional Ziegler–Nichols tuned PI controller. In particular, faster rise time and negligible overshoot are observed under no-load conditions, while more effective recovery behavior is achieved under step load disturbances. The coordinated action of feedforward compensation, adaptive error shaping, and anti-windup contributes to smooth control effort and stable system performance.

Overall, the proposed approach offers a flexible and self-adaptive solution for PMSM drives, maintaining accurate speed tracking with limited overshoot and improved robustness under dynamic operating conditions. The results suggest that learning-based gain adaptation is a promising direction for high-performance motor control applications.

Despite these advantages, several limitations should be acknowledged. The proposed method introduces additional computational complexity compared to fixed-gain PI controllers, which may increase implementation requirements on embedded platforms. Moreover, its performance depends on the selection of learning parameters, which may require tuning for different scenarios.

Finally, the validation in this study is limited to simulation results, and experimental verification on real hardware will be necessary to confirm practical applicability.

Future work will focus on embedded implementation and hardware-in-the-loop validation to assess real-time feasibility and further evaluate performance under practical operating conditions.

REFERENCES

[1] B. K. Bose, *Modern Power Electronics and AC Drives*, Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[2] R. Krishnan, *Permanent Magnet Synchronous and Brushless DC Motors*, Boca Raton, FL, USA: CRC Press, 2010, doi: https://doi.org/10.1201/9781420014235

[3] P. Vas, *Vector Control of AC Machines*, Oxford, U.K.: Oxford Univ. Press, 1990.

[4] K. J. Åström and T. Hägglund, *Advanced PID Control*, Research Triangle Park, NC, USA: ISA-Instrumentation, Systems, and Automation Society, 2006.

[5] M. S. Boucherit, "Adaptive speed control of permanent magnet synchronous motor," in *Proc. Int. Conf. Electrical, Electronic and Computer Engineering (ICEEC'04)*, 2004, pp. 1–6, doi: https://doi.org/10.1109/ICEEC.2004.1374633

[6] O. Dogru, K. Velswamy, F. Ibrahim, Y. Wu, A. Sundaramoorthy, B. Huang, S. Xu, M. Nixon, and N. H. Bell, "Reinforcement Learning Approach to Autonomous PID Tuning", in *Proceedings of the 2022 American Control Conference (ACC), Atlanta, GA, USA*, 2022, pp. 2691–2696, doi: https://doi.org/10.23919/ACC53348.2022.9867687

[7] W. J. Shipman and L. Coetzee, "Reinforcement learning and deep neural networks for PI controller tuning," in *Proc. 18th IFAC Symp. Control, Optimization and Automation in Mining, Mineral and Metal Processing (MMM 2019)*, 2019, pp. 1–6, doi: https://doi.org/10.1016/j.ifacol.2019.09.173

[8] P. Lu, J. Xiao, F. Zhou, and W. Hu, "Adaptive proportional–integral robust control of an uncertain robotic manipulator based on deep deterministic policy gradient," *Guangzhou Univ., Guangzhou, China, Tech. Rep.*, 2021, doi: https://doi.org/10.3390/math9172055

[9] W. Xu, J. Wang, D. Liu, *et al.*, "A deep reinforcement learning-based control method for electric linear loading systems" *J Mech Sci Technol 39, 7815–7827*, 2025. Doi: https://doi.org/10.1007/s12206-025-1139-8

[10] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," *arXiv:1801.01290*, 2018, doi: https://doi.org/10.48550/arXiv.1801.01290

[11] A. Kumar et al., "Deep learning-enabled adaptive control of electric drives," *IEEE Trans. Power Electron.*, Apr. 2023, vol. 38, no. 4, pp. 5000–5012, doi: https://doi.org/10.1109/ACCESS.2025.3648165

[12] W Cui, Y Jiang, B Zhang, "Reinforcement learning for optimal primary frequency control: A Lyapunov approach,"*IEEE Transactions on Power Systems*, 2023, vol. 38, no. 2, pp. 1676–1688, doi: https://doi.org/10.1109/TPWRS.2022.3176525

[13] J. Yao, M. Han, and X. Yin, "Lyapunov-based distributed reinforcement learning control with stability guarantee," *Computers & Chemical Engineering*, Apr. 2025, vol. 195, Art. no. 108979, doi: https://doi.org/10.1016/j.compchemeng.2024.108979

[14] F. Li and Y. Du, Eds., *Deep Learning for Power System Applications*. Cham, Switzerland: Springer, 2024, doi: https://doi.org/10.1007/978-3-031-45357-1

[15] M. J. Hossain and S. Debnath, "Neural-Network-Based Intelligent Control for Permanent Magnet Motors: A 2025 Perspective," *Journal of Control and Automation*, vol. 15, no. 2, pp. 112-128, Feb. 2025, doi: https://doi.org/10.1038/s41598-025-02396-y