


Adaptive Control of a Three-Phase Induction Motor Drive Based on Deep Reinforcement Learning Using the Soft Actor–Critic Algorithm



Chau Thanh Phuong 

Faculty of Electronic Engineering Technology, University of Economics-Technology for Industries, Ha Noi 100000, Vietnam

Corresponding Author Email: chauthanhphuong.uneti@gmail.com

Copyright: ©2026 The author. This article is published by IIETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.590301>

ABSTRACT

Received: 7 January 2026

Revised: 11 March 2026

Accepted: 25 March 2026

Available online: 31 March 2026

Keywords:

deep reinforcement learning, soft actor–critic, adaptive control, induction motor, field-oriented control

Three-phase induction motor drives are widely adopted in industrial systems, typically relying on field-oriented control (FOC) combined with proportional–integral (PI) regulators. Despite their simplicity, these controllers often exhibit degraded performance under parameter uncertainties, nonlinear effects, and load disturbances. In this study, a data-driven control strategy based on deep reinforcement learning (DRL) is proposed, where the Soft Actor–Critic (SAC) algorithm is employed to replace conventional PI regulators in the inner current loops. By incorporating entropy regularization, the SAC agent is capable of achieving a balance between exploration and exploitation, thereby improving robustness and adaptability. The effectiveness of the proposed method is validated through MATLAB/Simulink simulations under various operating conditions. The results indicate that the SAC-based controller ensures stable operation and maintains high tracking accuracy even in the presence of significant parameter variations.

1. INTRODUCTION

Three-phase induction motors continue to play a pivotal role in modern industrial drive systems due to their simple structure, high reliability, low investment and maintenance costs, and stable operation over a wide power range [1, 2]. In the context of Industry 4.0, induction motors remain widely employed in applications such as pumps, fans, conveyors, industrial robots, and automated manufacturing systems, where continuous operation and high durability are required [3]. Among these approaches, field-oriented control (FOC) enables independent regulation of torque and flux, leading to improved dynamic behavior and higher precision [4, 5]. Owing to these advantages, FOC has become a standard control structure in many modern industrial electric drives [6].

In conventional implementations, PI controllers are widely used in both current and speed loops because of their simple structure and ease of tuning [7]. However, these PI controllers are typically designed based on linearized models under the assumption of constant motor parameters. In practice, parameters such as stator resistance, rotor resistance, and inductances can vary significantly due to temperature effects, magnetic saturation, and aging, leading to performance degradation and steady-state errors under load disturbances or parameter uncertainties [8, 9].

To overcome these limitations, various nonlinear and adaptive control strategies have been introduced, including sliding mode control, backstepping, and artificial intelligence–based control approaches [10–12]. Although these methods achieve certain performance improvements, most of them still require relatively accurate mathematical models of the plant, which poses challenges for practical implementation in

industrial drive systems with continuously varying operating conditions [13].

Recently, deep reinforcement learning (DRL) has emerged as a promising alternative for control design. By learning directly from system interactions, DRL eliminates the need for an explicit model and demonstrates strong capability in handling nonlinear and uncertain systems [14, 15]. The ability of DRL to manage nonlinear dynamics has led to its growing adoption in electric drive control as well as power electronics applications [16].

Among state-of-the-art DRL algorithms, Soft Actor–Critic (SAC) has been highly regarded due to its entropy-regularized objective, which jointly optimizes the expected reward and policy entropy, effectively balancing exploration and exploitation during learning [17]. This mechanism allows SAC to generate flexible and robust control policies that are less sensitive to noise and parameter variations [18]. Recent studies have shown that SAC outperforms the DDPG algorithm, particularly in environments with strong uncertainties and disturbances [19, 20].

Motivated by the aforementioned analysis, this study proposes integrating the SAC algorithm into the FOC structure with the aim of improving system adaptability and robustness against parameter uncertainties and load disturbances.

2. THEORETICAL BACKGROUND OF THE SOFT ACTOR–CRITIC ALGORITHM

2.1 Entropy-regularized objective of Soft Actor–Critic

Unlike conventional reinforcement learning (RL) methods

that maximize only the expected return, the SAC algorithm adopts an entropy-regularized objective, which augments the reward with a policy entropy term [17]:

$$J(\pi) = E_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot | s_t)) \right) \right] \quad (1)$$

where, the entropy term $\mathcal{H}(\pi(\cdot | s_t))$ quantifies the randomness of the policy and is given by $-E_{a_t \sim \pi}[\log \pi(a_t | s_t)]$. The coefficient $\alpha > 0$ controls how much exploration is encouraged relative to the objective of maximizing cumulative reward.

The entropy maximization mechanism encourages stochasticity in the learned policy, leading to improved robustness and stability in uncertain and noisy environments.

Within the SAC algorithm, value functions incorporate entropy and are therefore referred to as ‘‘soft’’ value functions. Specifically, the soft Q-function is defined by:

$$Q^{\pi}(s_t, a_t) = r(s_t, a_t) + \gamma E_{s_{t+1} \sim \mathbb{P}}[V^{\pi}(s_{t+1})] \quad (2)$$

While the associated soft state–value function is given as:

$$V^{\pi}(s_t) = E_{a_t \sim \pi}[Q^{\pi}(s_t, a_t) - \alpha \log \pi(a_t | s_t)] \quad (3)$$

These definitions lead to the soft Bellman equations, which serve as the theoretical foundation for updating the critic networks in the SAC approach.

2.2 Architecture of the Actor–Critic model in Soft Actor–Critic

The SAC algorithm employs an Actor–Critic architecture consisting of: A stochastic policy network (Actor) $\pi_{\theta}(a | s)$, parameterized by θ ; Two independent action–value networks (Critics) $Q_{\phi_1}(s, a)$ and $Q_{\phi_2}(s, a)$; Corresponding target networks $Q_{\phi_1'}$ and $Q_{\phi_2'}$ [16].

The use of double critics mitigates overestimation bias during value function approximation.

2.2.1 Critic network update

Each critic is updated through the minimization of the following loss function [19].

$$\mathcal{L}(\phi_i) = E_{(s_t, a_t, r_t, s_{t+1})} \left[(Q_{\phi_i}(s_t, a_t) - y_t)^2 \right] \quad (4)$$

where, the target value y_t is computed as:

$$y_t = r_t + \gamma \left(\min_{i=1,2} Q_{\phi_i'}(s_{t+1}, a_{t+1}) - \alpha \log \pi_{\theta}(a_{t+1} | s_{t+1}) \right) \quad (5)$$

with $a_{t+1} \sim \pi_{\theta}(\cdot | s_{t+1})$.

2.2.2 Actor network update

The update of the policy network (Actor) is performed by minimizing the following objective function [17]:

$$\mathcal{L}(\theta) = E_{s_t \sim \mathbb{D}, a_t \sim \pi_{\theta}} \left[\alpha \log \pi_{\theta}(a_t | s_t) - \min_{i=1,2} Q_{\phi_i}(s_t, a_t) \right] \quad (6)$$

This objective drives the policy toward actions with high expected value while maintaining sufficient entropy.

2.3 Automatic entropy temperature adjustment

An important feature of SAC is the automatic tuning of the entropy temperature parameter α . Rather than using a fixed value, the parameter α is adaptively learned by via the minimization of the following objective [18]:

$$\mathcal{L}(\alpha) = E_{a_t \sim \pi_{\theta}} [-\alpha (\log \pi_{\theta}(a_t | s_t) + \mathcal{H}_{\text{tre}})] \quad (7)$$

where, \mathcal{H}_{tre} is a predefined target entropy. This adaptive mechanism enables SAC to dynamically balance exploration and exploitation throughout the learning process.

Due to its stochastic policy, entropy-regularized optimization, and off-policy learning capability, SAC is particularly suitable for nonlinear control systems with disturbances and model uncertainties. In the context of induction motor drives, SAC enables the controller to learn adaptive control policies directly from system interactions without relying on an accurate mathematical model of the motor. Consequently, the SAC algorithm provides an effective framework for improving robustness and dynamic performance in induction motor control systems.

3. DESIGN OF THE SOFT ACTOR–CRITIC CONTROLLER FOR THE INDUCTION MOTOR

This section presents the design of the proposed SAC–based controller integrated into the FOC framework for a three-phase induction motor drive. The mathematical model of the induction motor in the synchronous $d - q$ reference frame is first introduced, followed by the SAC-based current control formulation, including state–action representation, reward design, training strategy, and stability analysis.

3.1 Induction motor model in the d - q reference frame

By considering a synchronous d - q coordinate system aligned with the rotor flux, the dynamic equations of the three-phase induction motor are given as follows [4, 5]:

$$\begin{cases} \frac{di_d}{dt} = -\frac{R_s}{L_s} i_d + \omega_e i_q + \frac{1}{L_s} v_d - \frac{L_m}{L_s L_r} \frac{d\psi_r}{dt} \\ \frac{di_q}{dt} = -\frac{R_s}{L_s} i_q - \omega_e i_d + \frac{1}{L_s} v_q - \frac{L_m}{L_s L_r} \omega_r \psi_r \end{cases} \quad (8)$$

where, i_d, i_q are the stator currents along the d - and q -axes, respectively; v_d, v_q denote the control voltages; R_s, L_s are the stator resistance and inductance; L_m, L_r represent the magnetizing inductance and rotor inductance; ω_e is the synchronous electrical angular speed; ψ_r denotes the rotor flux magnitude.

The electromagnetic torque is expressed as:

$$T_e = \frac{3P}{2} \frac{L_m}{L_r} \psi_r i_q \quad (9)$$

where, P is the number of poles.

The mechanical dynamics of the induction motor are governed by:

$$J \frac{d\omega_r}{dt} = T_e - T_L - B\omega_r \quad (10)$$

where, J is the inertia parameter, B defines the viscous friction effect, and T_L indicates the torque imposed by the external load.

3.2 Soft Actor–Critic-based control architecture

In the conventional FOC structure, the d -axis stator current i_d is used to regulate the rotor flux, while the q -axis stator current i_q determines the electromagnetic torque. In this study, the SAC algorithm is integrated into the inner current control loops, replacing the conventional PI controllers.

Figure 1 illustrates the proposed closed-loop control architecture, in which the SAC-based current controllers are embedded within the classical FOC framework.

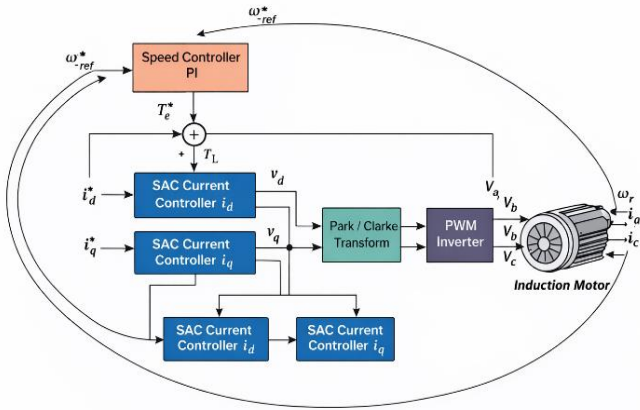


Figure 1. Schematic representation of the induction motor speed control system incorporating a Soft Actor–Critic (SAC) controller in the current loops

The proposed control structure consists of the following components:

- Outer loop: a PI-based speed controller that generates the reference torque or i_d^* ;
- Inner loops: SAC-based current controllers regulating the d - and q -axis stator currents i_d and i_q ;
- Coordinate transformation and power stage: Park–Clarke transformation blocks and a PWM voltage source inverter.

This hybrid architecture preserves the decoupling principle and physical interpretability of the conventional FOC scheme while significantly enhancing adaptability and robustness through DRL.

3.3 Soft Actor–Critic formulation for current control

The current control problem is represented as a Markov Decision Process (MDP) to enable the application of SAC [14, 15].

The state vector is selected as:

$$s_t = [e_{id}(t) \quad e_{iq}(t) \quad i_d(t) \quad i_q(t) \quad \omega_r(t)]^T \quad (11)$$

where, $e_{id} = i_d^{\text{ref}} - i_d$, $e_{iq} = i_q^{\text{ref}} - i_q$.

Here, e_{id} and e_{iq} denote the current tracking errors in the d - and q -axes, respectively; i_d^{ref} and i_q^{ref} are the reference currents

generated by the outer control loop; i_d, i_q are the measured stator currents; and ω_r is the rotor angular speed.

This state representation provides sufficient information for the SAC agent to regulate both flux and torque dynamics.

The action selected by the SAC agent corresponds to the control voltages applied to the inverter:

$$a_t = [v_d(t) \quad v_q(t)]^T$$

which are constrained by the inverter voltage limits:

$$|v_d| \leq V_{max}, |v_q| \leq V_{max}.$$

These constraints ensure safe operation of the power converter and prevent voltage saturation.

The reward function plays a key role in shaping the learning behavior of the SAC agent. In this study, the reward is designed to minimize current tracking errors while penalizing excessive control effort [15]:

$$r_t = - \left(w_d e_{id}^2 + w_q e_{iq}^2 + w_v (v_d^2 + v_q^2) \right) \quad (12)$$

where, $w_d > 0, w_q > 0$, and $w_v > 0$ are weighting coefficients.

This reward formulation encourages the agent to:

- accurately track the reference currents,
- suppress oscillations and disturbances,
- limit excessive voltage commands and avoid inverter saturation.

3.4 Soft Actor–Critic policy update for current control

Based on the formulation in Section 2, the two critic networks are updated by minimizing the following loss function:

$$\mathcal{L}(\phi_i) = E \left[\left(Q_{\phi_i}(s_t, a_t) - r_t - \gamma \min_{j=1,2} Q_{\phi_j'}(s_{t+1}, a_{t+1}) + \gamma \alpha \log \pi_{\theta}(a_{t+1} | s_{t+1}) \right)^2 \right] \quad (13)$$

where, $a_{t+1} \sim \pi_{\theta}(\cdot | s_{t+1})$, γ is the discount factor, and α denotes the entropy temperature.

The actor (policy) network is updated by minimizing [16].

$$\mathcal{L}(\theta) = E \left[\alpha \log \pi_{\theta}(a_t | s_t) - \min_{i=1,2} Q_{\phi_i}(s_t, a_t) \right] \quad (14)$$

This objective drives the policy toward actions with high expected value while maintaining sufficient exploration through entropy maximization.

To ensure training stability and convergence, the state and action signals are normalized as:

$$\tilde{s}_t = \frac{s_t - s_{min}}{s_{max} - s_{min}}, \tilde{a}_t = \tanh(a_t) \quad (15)$$

The hyperbolic tangent function $\tanh(\cdot)$ is employed to automatically bound the output voltages within a safe

operating range, thereby improving robustness and preventing actuator saturation.

The training process is conducted in a simulation environment consisting of the induction motor and the voltage source inverter. The interaction data (s_t, a_t, r_t, s_{t+1}) are stored in a replay buffer \mathcal{D} for off-policy learning.

The target networks are updated using a soft update rule:

$$\phi'_i \leftarrow \tau \phi_i + (1 - \tau) \phi'_i \quad (16)$$

where, $0 < \tau \ll 1$ is the smoothing factor.

This training strategy improves learning stability and enables the SAC agent to achieve robust current control performance under parameter uncertainties and external disturbances.

3.5 Stability analysis and proof of the Soft Actor–Critic field-oriented control system

In the FOC framework, if the stator currents i_d and i_q accurately track their reference values i_d^* and i_q^* , the mechanical dynamics of the induction motor can be approximated as a stable first-order linear system. Consequently, the overall stability of the drive system can be reduced to the stability analysis of the inner current control loops.

Consider the current tracking errors in the synchronous dq reference frame: $e_d = i_d - i_d^*$, $e_q = i_q - i_q^*$.

From the induction motor model, the error dynamics can be expressed in the following general form:

$$\dot{e} = f(e, x) + g(e, x)u \quad (17)$$

where, $e = [e_d \ e_q]^T$, $u = [v_d \ v_q]^T$ is the control input generated by the SAC controller; $f(\cdot)$ represents the nonlinear terms and disturbances arising from parameter uncertainties and load variations; $g(\cdot)$ denotes the control gain matrix, which is bounded and locally invertible.

A Lyapunov candidate function for the inner current control loop is chosen as [11]:

$$V(e) = \frac{1}{2} e^T e = \frac{1}{2} (e_d^2 + e_q^2) \quad (18)$$

Clearly, $V(e) > 0$, $\forall e \neq 0$, $V(0) = 0$.

Taking the time derivative yields:

$$\dot{V}(e) = e^T \dot{e} \quad (19)$$

Substituting the error dynamics gives:

$$\dot{V}(e) = e^T (f(e, x) + g(e, x)u) \quad (20)$$

The SAC algorithm learns a control policy $u = \pi_\theta(s)$ by minimizing the expected cumulative cost, with the reward function designed as:

$$r_t = -(w_d e_d^2 + w_q e_q^2 + w_u |u|^2) \quad (21)$$

Maximizing the discounted cumulative reward is equivalent to minimizing the expected cost functional:

$$J = E \left[\int_0^\infty (e^T Q e + u^T R u) dt \right] \quad (22)$$

where, $Q = \text{diag}(w_d, w_q) > 0$, $R = w_u I > 0$.

Therefore, the SAC policy approximates a nonlinear optimal control law in which: current tracking errors are heavily penalized, control inputs are constrained and regularized.

Since SAC is a function-approximation-based learning algorithm, it cannot guarantee that $\dot{V}(e) < 0 \forall e$.

However, once the policy has converged, stability in the expected sense can be established:

$$E[\dot{V}(e)] \leq -\lambda E[|e|^2], \lambda > 0 \quad (23)$$

This inequality implies:

$$E[V(e(t))] \leq V(e(0)) e^{-\lambda t} \quad (24)$$

Accordingly, the current control subsystem can be considered asymptotically stable in terms of mean-square convergence, which ensures practical stability and robustness of the overall SAC–FOC-controlled induction motor drive.

4. SIMULATION AND PERFORMANCE EVALUATION

4.1 Simulation environment and system parameters

The simulation studies were carried out in the MATLAB/Simulink environment, incorporating a three-phase induction motor model and a two-level PWM voltage source inverter. To ensure the reproducibility of the results, the hyperparameters of the SAC algorithm are clearly specified. The actor and critic networks consist of hidden layers with an appropriate number of neurons to balance accuracy and computational cost. The training parameters, such as the learning rate, minibatch size, and training steps, are selected based on simulation experiments to achieve stable learning performance. Table 1 summarizes the main hyperparameters used in this study.

Table 1. Hyperparameters of the Soft Actor–Critic (SAC) algorithm

Parameter	Value	Description
Actor network structure	2 hidden layers (128–128)	Fully connected layers
Critic network structure	2 hidden layers (128–128)	Twin Q-networks
Activation function	ReLU	Used in hidden layers
Learning rate (actor)	3×10^{-4}	Policy network learning rate
Learning rate (critic)	3×10^{-4}	Q-network learning rate
Discount factor γ	0.99	Future reward discount
Entropy coefficient α	Automatically tuned	Entropy regularization
Replay buffer size	10^6	Experience storage
Mini-batch size	256	Training batch size
Training steps	50000	Total training iterations

The main parameters of the induction motor used in the

simulations are listed in Table 2 [7].

Table 2. Induction motor parameters

Parameter	Symbol	Value
Rated power	P_n	3 kW
Rated voltage	U_n	380 V
Stator resistance	R_s	1.2 Ω
Rotor resistance	R_r	1.0 Ω
Stator inductance	L_s	0.17 H
Rotor inductance	L_r	0.17 H
Magnetizing inductance	L_m	0.165 H
Moment of inertia	J	0.02 kg·m ²

The control sampling period was selected as $T_s = 100 \mu\text{s}$. For the proportional–integral field-oriented control (PI–FOC) scheme, the proportional and integral gains of the PI controller are selected to achieve a fast transient response, small overshoot, and low steady-state error. The tuning process is performed through repeated simulation experiments until the best control performance for the motor drive system is obtained. The final parameter values of the PI controller are used as the baseline method for comparison in this study. Two control structures were compared in the simulations. The PI–FOC scheme uses PI controllers for both the speed loop and the current loops (i_d, i_q). In contrast, the SAC–FOC approach preserves the PI-based speed loop but replaces the current controllers with SAC-based ones. Both approaches employed the same FOC framework, rotor flux observer, and pulse width modulation (PWM) inverter to ensure a fair comparison.

4.2 Simulation scenarios

The performance of the proposed RL–based control strategy is investigated through a set of simulation scenarios based on a three-phase induction motor drive model implemented in MATLAB/Simulink. The SAC–FOC scheme is systematically compared with the conventional PI–FOC approach under identical operating conditions. The simulation parameters of the motor, inverter, and controllers are kept constant throughout the simulations to ensure an objective comparison. The total simulation time is set to 0.5 s, and the following specific scenarios are considered:

Scenario 1: Speed reference variation

The motor speed reference is varied in a ramp form from 600 rpm to 1000 rpm during the time interval from 0.1 s to 0.3 s. This scenario is designed to evaluate the speed tracking capability, transient performance, and steady-state error of the control system.

Scenario 2: Stator current response

The stator current components are observed during both transient and steady-state operating conditions to assess current tracking accuracy, oscillation level, and waveform smoothness. This analysis directly reflects the control quality of the inner current control loops.

Scenario 3: Load disturbance rejection

A step load torque disturbance is applied to the motor shaft, where the load torque abruptly increases from 10 Nm to 60 Nm at 0.1 s and then decreases to 30 Nm at 0.4 s. This scenario is used to evaluate the disturbance rejection capability and speed recovery performance of the control system.

4.3 Simulation results and discussion

To evaluate the learning effectiveness of the SAC algorithm,

the training process is illustrated through the cumulative reward in Figure 2 and the loss function value versus training iterations in Figure 3. The results show that the agent’s reward gradually increases during the learning process, while the critic network loss decreases and progressively converges. This indicates that the SAC algorithm is capable of learning an effective control policy and achieving stable training performance.

The simulation results indicate that both control strategies ensure stable system operation under all considered scenarios. However, the RL–based controller exhibits several advantages over the conventional PI controller.

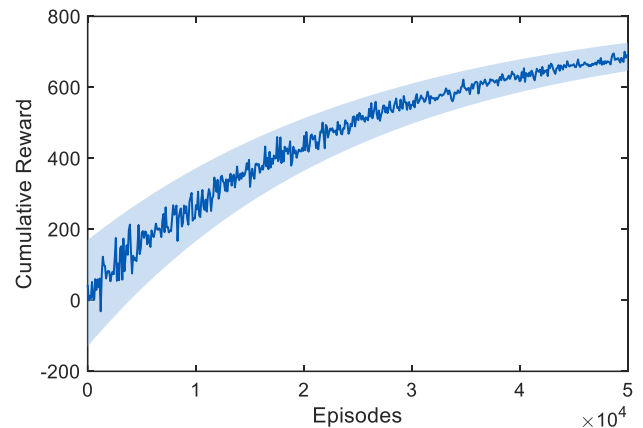


Figure 2. Training reward versus episodes during the learning process of the Soft Actor–Critic (SAC) agent

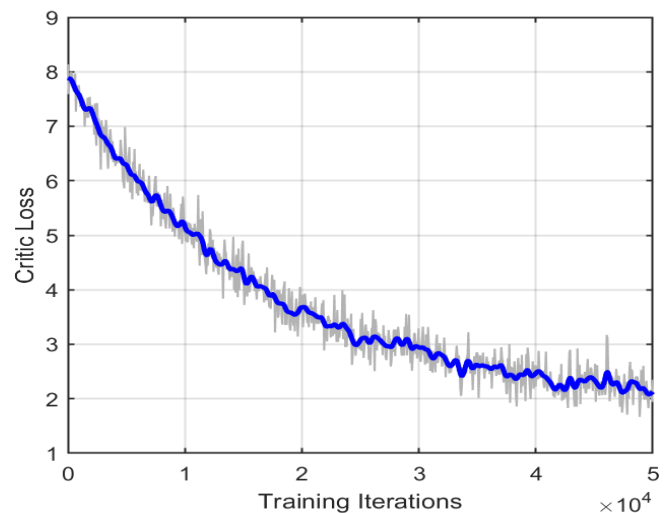


Figure 3. Critic loss versus training iterations during the Soft Actor–Critic (SAC) training process

Specifically, in the speed reference variation scenario (Figure 4), the SAC–FOC approach provides faster speed tracking, smaller overshoot, and shorter settling time compared with the PI–FOC method. The steady-state error is nearly zero, demonstrating the strong adaptability of the RL–based controller.

For the stator current responses (Figures 5 and 6), the SAC–FOC controller generates smoother current waveforms with lower oscillation amplitude and reduced RMS values compared to the PI–FOC scheme, particularly during transient conditions. These results indicate a significant improvement in the control quality of the inner current loops.

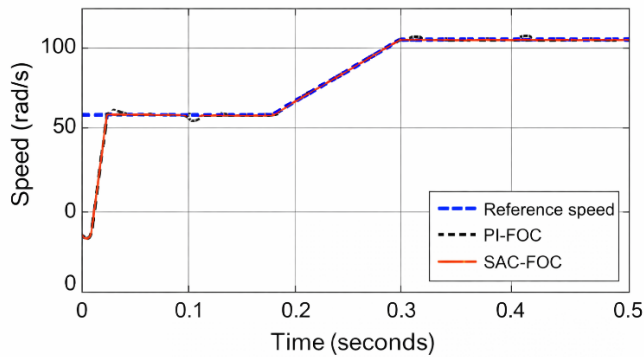


Figure 4. Speed response of the system using PI-FOC and SAC-FOC controllers
 Note: PI = proportional–integral; FOC = field-oriented control; SAC = Soft Actor–Critic

plots are fully consistent with the quantitative performance indices reported.

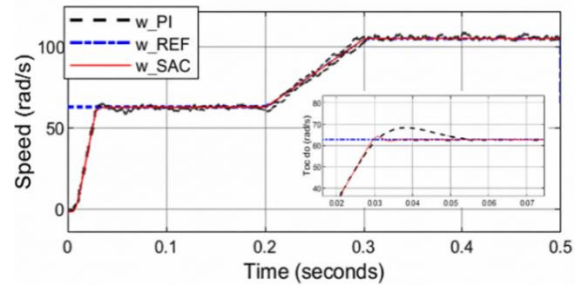


Figure 7. Speed response of the system with SAC and PI controllers under load variations
 Note: PI = proportional–integral; SAC = Soft Actor–Critic

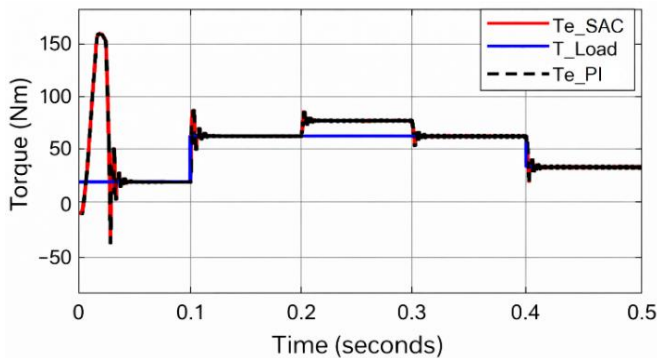


Figure 5. Torque response of the system using PI-FOC and SAC-FOC controllers
 Note: PI = proportional–integral; FOC = field-oriented control; SAC = Soft Actor–Critic

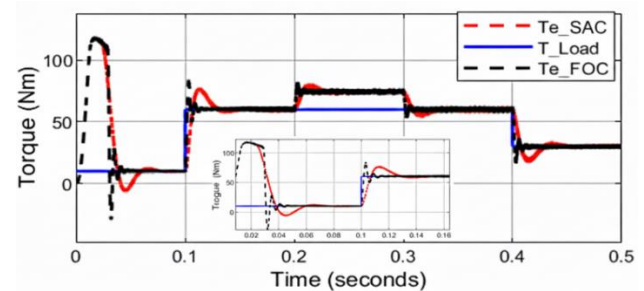


Figure 8. Torque response of the system with SAC and PI controllers under load variations
 Note: PI = proportional–integral; SAC = Soft Actor–Critic

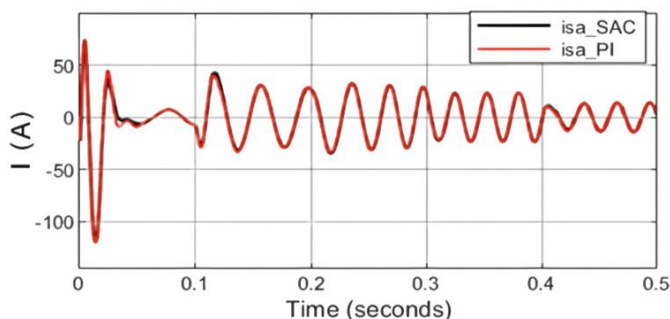


Figure 6. Current response of the system using PI-FOC and SAC-FOC controllers
 Note: PI = proportional–integral; FOC = field-oriented control; SAC = Soft Actor–Critic

When subjected to load torque disturbances (Figures 7 and 8), the system employing SAC–FOC exhibits superior disturbance rejection performance, characterized by smaller speed deviations and faster recovery compared to the system using PI–FOC. In contrast, the PI controller shows larger oscillations and a longer recovery time.

Overall, the simulation results confirm that the RL–based controller not only improves transient response performance but also enhances disturbance rejection capability and system robustness under varying operating conditions.

The simulation results presented in Figures 4–8 are summarized and quantitatively evaluated in Table 3. It can be observed that the qualitative observations derived from the

Table 3. Quantitative performance comparison between PI–FOC and SAC–FOC

Performance index	PI–FOC	SAC–FOC
Speed overshoot (%)	8.5	2.1
Settling time (s)	0.18	0.09
Steady-state speed error (rad/s)	±1.2	±0.2
Recovery time under load disturbance (s)	0.15	0.06
RMS stator current (A)	6.4	5.6
Peak current oscillation (A)	1.9	0.8

Note: PI = proportional–integral; FOC = field-oriented control; SAC = Soft Actor–Critic

Specifically, the speed overshoot under the SAC–FOC scheme is significantly smaller than that of the PI–FOC approach, which agrees well with the smoother and less oscillatory speed response observed in Figure 6. This improvement is clearly reflected in Table 2, where the overshoot is reduced from 8.5% to 2.1%.

The settling time of the system using SAC–FOC is also reduced by nearly half compared to PI–FOC, consistent with the speed response plots showing a faster convergence to steady-state operation after speed reference changes. This result is quantitatively confirmed by the reduction of the settling time from 0.18 s to 0.09 s, as reported in Table 3.

Regarding current control performance, the stator current waveforms under SAC–FOC are smoother and exhibit lower oscillation amplitudes than those under PI–FOC, as observed in Figures 6–8. This is further validated by the decrease in the RMS stator current from 6.4 A to 5.6 A, indicating a significant reduction in current losses and electrical stress.

Therefore, the quantitative indices presented in Table 3 clearly and consistently confirm the advantages of the SAC–

FOC controller observed in the simulation results. The strong agreement between qualitative and quantitative analyses demonstrates the effectiveness and reliability of the proposed approach.

5. CONCLUSIONS

This study develops a SAC-based FOC strategy for three-phase induction motor drives, where the inner-loop PI controllers are replaced by learning-based controllers. This modification improves robustness against nonlinear dynamics, parameter variations, and external disturbances. Simulation findings reveal that the SAC-FOC approach provides faster response, higher tracking accuracy, and better disturbance rejection than the conventional PI-FOC scheme. These results confirm the strong potential of DRL-based controllers for high-performance electric drive applications operating under varying conditions. Moreover, the proposed control strategy does not rely on an accurate motor model, which reduces tuning effort and improves implementation flexibility.

In this study, the proposed control method is mainly evaluated through simulation experiments. In future work, the SAC-based controller will be implemented on real hardware platforms such as DSPs or microcontrollers to evaluate its real-time performance. In addition, issues related to computational latency and algorithm optimization for embedded systems will be further investigated.

ACKNOWLEDGMENT

This research was supported by Faculty of Electrical Engineering-Automation, University of Economics-Technology for Industries, Vietnam.

REFERENCES

- [1] Bose, B.K. (2008). Power electronics and motor drives recent progress and perspective. *IEEE Transactions On Industrial Electronics*, 56(2): 581-588. <https://doi.org/10.1109/TIE.2008.2002726>
- [2] Liu, C., Chau, K.T., Lee, C.H., Song, Z. (2020). A critical review of advanced electric machines and control strategies for electric vehicles. *Proceedings of the IEEE*, 109(6): 1004-1028. <https://doi.org/10.1109/JPROC.2020.3041417>
- [3] Qi, X., Holtz, J. (2020). Modeling and control of low switching frequency high-performance induction motor drives. *IEEE Transactions on Industrial Electronics*, 67(6): 4402-4410. <https://doi.org/10.1109/TIE.2019.2924602>
- [4] Youssef, O.E.M., Hussien, M.G., Hassan, A.E.W. (2022). A new simplified sensorless direct stator field-oriented control of induction motor drives. *Frontiers in Energy Research*, 10: 961529. <https://doi.org/10.3389/fenrg.2022.961529>
- [5] Ding, C.W., Tung, P.C. (2025). A new approach to field-oriented control that substantially improves the efficiency of an induction motor with speed control. *Applied Sciences*, 15(9): 4845. <https://doi.org/10.3390/app15094845>
- [6] Nevoloso, C., Di Tommaso, A.O., Miceli, R., Scaglione, G., Foti, S., Testa, A. (2024). Impact analysis of FOC-based synchronous PWM strategy on traction induction motor drives performance. In *2024 International Conference on Electrical Machines (ICEM)*, Torino, Italy, pp. 1-7. <https://doi.org/10.1109/ICEM60801.2024.10700225>
- [7] Zaky, M.S. (2015). A self-tuning PI controller for the speed control of electrical motor drives. *Electric Power Systems Research*, 119: 293-303. <https://doi.org/10.1016/j.epsr.2014.10.004>
- [8] Martín, C., Bermúdez, M., Barrero, F., Arahal, M.R., Kestelyn, X., Durán, M.J. (2017). Sensitivity of predictive controllers to parameter variation in five-phase induction motor drives. *Control Engineering Practice*, 68: 23-31. <https://doi.org/10.1016/j.conengprac.2017.08.001>
- [9] Adigintla, S., Aware, M.V. (2023). Robust fractional order speed controllers for induction motor under parameter variations and low speed operating regions. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 70(3): 1119-1123. <https://doi.org/10.1109/TCSII.2022.3220526>
- [10] Wu, L., Liu, J., Vazquez, S., Mazumder, S.K. (2022). Sliding mode control in power converters and drives: A review. *IEEE/CAA Journal of Automatica Sinica*, 9(3): 392-406. <https://doi.org/10.1109/JAS.2021.1004380>
- [11] Accetta, A., Cirrincione, M., Di Girolamo, S., D'Ippolito, F., Pucci, M., Sferlazza, A. (2025). Robust nonlinear control for induction motor drives based on adaptive disturbance compensation. *IEEE Transactions on Industry Applications*, 61(2): 3163-3173. <https://doi.org/10.1109/TIA.2025.3532561>
- [12] Chuyen, T.D., Van Hoa, R., Co, H.D., Huong, T.T., Ha, P.T.T., Linh, B.T.H., Nguyen, T.L. (2022). Improving control quality of PMSM drive systems based on adaptive fuzzy sliding control method. *International Journal of Power Electronics and Drive Systems (IJPEDS)*, 13(2): 835-845. <https://doi.org/10.11591/ijpeds.v13.i2.pp835-845>
- [13] El-Sousy, F.F.M., Abuhasel, K.A. (2018). Nonlinear robust optimal control via adaptive dynamic programming of permanent-magnet linear synchronous motor drive for uncertain two-axis motion control system. In *2018 IEEE Industry Applications Society Annual Meeting (IAS)*, Portland, OR, USA, pp. 1-12. <https://doi.org/10.1109/IAS.2018.8544612>
- [14] Sutton, R.S., Barto, A.G. (2020). *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA, USA: MIT Press.
- [15] Wang, X., Wang, S., Liang, X., Zhao, D., et al. (2024). Deep reinforcement learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4): 5064-5078. <https://doi.org/10.1109/TNNLS.2022.3207346>
- [16] Deng, H., Zhao, Y., Nguyen, A.T., Huang, C. (2023). Fault-tolerant predictive control with deep-reinforcement-learning-based torque distribution for four in-wheel motor drive electric vehicles. *IEEE/ASME Transactions on Mechatronics*, 28(2): 668-680. <https://doi.org/10.1109/TMECH.2022.3233705>
- [17] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., Levine, S. (2018). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

- <https://doi.org/10.48550/arXiv.1812.05905>
- [18] Jha, A.V., Bommur, S.R.R., Rao, V.S.S., Muralikrishna, S. (2024). Soft actor-critic algorithm in high-dimensional continuous control tasks. In 2024 International Conference on Modeling, Simulation & Intelligent Computing (MoSICom), Dubai, United Arab Emirates, pp. 535-540. <https://doi.org/10.1109/MoSICom63082.2024.10881478>
- [19] Liu, Y., Man, K.L., Li, G., Payne, T.R., Yue, Y. (2024). Evaluating and selecting deep reinforcement learning models for optimal dynamic pricing: A Systematic comparison of PPO, DDPG, and SAC. In 2024 8th International Conference on Control Engineering and Artificial Intelligence (CCEAI), pp. 215-219. <https://doi.org/10.1145/3640824.3640871>
- [20] Liu, S. (2024). An evaluation of DDPG, TD3, SAC, and PPO: Deep reinforcement learning algorithms for controlling continuous system. In Proceedings of the 2023 International Conference on Data Science, Advanced Algorithm and Intelligent Computing (DAI 2023), Atlantis Press, pp. 15-24. https://doi.org/10.2991/978-94-6463-370-2_3