An Adaptive Weight Update Based Supervised Learning Predictive Control Method for PMSM

Sen Hong, Yixiao Luo, *Member, IEEE*, Kai Yang, *Senior Member, IEEE*, Shuangxia Niu, *Senior Member, IEEE*, and Jincheng Yu, *Member, IEEE*

***Abstract*—The existing learning-based predictive control methods rely on large amounts of data for offline training. To address this issue, this article proposes a supervised learning-based continuous control set predictive method, which only requires the sampling data at the current time to achieve online updates of the controller, significantly reducing implementation complexity and data requirements. Specifically, the proposed method approximates the ideal input using a radial basis function neural network and updates the weight matrix via gradient descent. Through the derivation of the relationship between state error and tracking error, the tracking error term in the update law is replaced, enabling the online update of the approximator. Considering the inherent approximation error of the neural network and external disturbances, a robust control term is introduced to correct the approximated input. Finally, the proposed method is validated on an experimental platform.**

***Index Terms*—Model predictive control (MPC), neural network (NN), permanent magnet synchronous motor (PMSM), supervised learning.**

I. Introduction

MODEL predictive control (MPC) has become a widely used strategy for controlling permanent magnet synchronous motors (PMSMs), owing to its advantages such as fast dynamic response and excellent steady-state performance [1], [2]. Depending on the implementation approach, MPC can be categorized into finite control set MPC (FCS-MPC) and continuous control set (CCS-MPC). Compared to FCS-MPC, CCS-MPC generates the optimal control inputs directly and exhibits lower current ripple and torque ripple in the steady state. Nevertheless, CCS-MPC may experience a substantial decline in performance if the controller parameters deviate from those of the actual system [3], [4].

Received 24 May 2025; revised 16 August 2025, 9 October 2025, and 19 October 2025; accepted 2 November 2025. Date of publication 10 November 2025; date of current version 19 January 2026. This work was supported by the National Key Research and Development Program of China under Grant 2022YFB4201200. Recommended for publication by Associate Editor P. Karamanakos. *(Corresponding author: Yixiao Luo.)*

Sen Hong, Yixiao Luo, and Kai Yang are with the School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: m202472450@hust.edu.cn; luoyx@hust.edu.cn; yk@hust.edu.cn).

Shuangxia Niu is with the Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: shuangxia.niu@polyu.edu.hk).

Jincheng Yu is with the School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China (e-mail: yujincheng@hit.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPEL.2025.3630682.

Digital Object Identifier 10.1109/TPEL.2025.3630682

A. Literature Review

A wide range of strategies has been investigated in previous studies to address the adverse impact of parameter mismatch in control system design. The first class focuses on parameter identification techniques, which seek to continuously adjust system parameters during operation, thereby improving the control system's resilience to parameter mismatch. [5]. However, due to the nonlinear characteristics of inverters, the accuracy of parameter identification is often difficult to guarantee. Moreover, the complexity of simultaneously identifying multiple parameters poses a significant challenge, limiting the practical applicability of these methods [6]. The second category focuses on observer-based methods, which estimate and compensate for the equivalent disturbances caused by parameter mismatches in real time. Commonly used observers include extended state observers (ESOs) [7], sliding mode observers [8], and Luenberger observers [9]. While such methods offer notable advantages in improving system robustness, they often require intricate parameter tuning. Additionally, both parameter identification and observer-based methods fundamentally rely on mathematical models of the motor, thereby constraining their generalizability and flexibility.

In recent years, the accelerated digitalization of industrial processes has led to a substantial increase in the volume of operational data generated. Leveraging this, data-driven control approaches have emerged as promising alternatives for improving the parameter robustness of MPC systems [10], [11]. These methods replace traditional motor models with data-driven constructs such as look-up table models, ultralocal models, and autoregressive models with exogenous inputs (ARX) [12]. The look-up table-based approach predicts the future current using stored current gradient information. However, its accuracy heavily depends on the precision of the current sampling. Furthermore, the repeated application of an identical voltage vector across consecutive control cycles can lead to stagnation in current gradient updates, ultimately impairing prediction accuracy [13]. The ultralocal model-based approach offers structural simplicity but still requires inductance parameters for gain computation [14]. In the ARX model, the motor dynamics are modeled as a set of discrete-time transfer functions, with the coefficients obtained through input–output decoupling, thus enabling effective control [15].

TABLE I
COMPARISON OF DIFFERENT LEARNING-BASED CONTROL METHODS

Methods / Metrics	Reinforcement learning	Deep learning	Imitation learning
Requirement for model parameters	No	No	No
Computational burden	Low to Moderate	Low to Moderate	Low
Implementation complexity	High	Very high	Moderate to high
Training data requirement	Large	Large	Large
Need for offline training	Yes	Yes	Yes
References	[16], [17], [18]	[19], [20]	[21], [22]

With the increasing adoption of artificial intelligence in motor control and power electronics, learning-based data-driven approaches—including reinforcement learning (RL), deep learning (DL), and imitation learning—have gained considerable attention in the MPC context. RL enables control policy learning through interaction with the environment, where the agent refines its behavior based on received rewards or penalties. For instance, an actor–critic RL strategy was proposed in [16], in which the controller learns to mimic an MPC-based policy without model parameters, ultimately optimizing the switching strategy for power converters. Additionally, deep Q-learning networks have been employed to design PMSM controllers [17]. However, the effectiveness of these methods is strongly influenced by both the adequacy and reliability of the available training data. To mitigate this issue, a meta-RL strategy was introduced in [18], framing the data environment as a partially observable Markov decision process to reduce dependence on large datasets. Nonetheless, overfitting remains a potential concern within this approach that could compromise system robustness. DL-based methods approximate the MPC control law through multilayer neural networks, thereby enhancing robustness to parameter variations. However, DL-based predictive control methods typically require the deployment of multilayer neural networks, and their training process is computationally intensive, which may pose challenges for real-time implementation [19], [20]. In contrast, imitation learning generates extensive training data using expert demonstrations or algorithms and uses this data to train agents that approximate the MPC control law, thereby eliminating dependence on explicit model parameters. Nevertheless, the effectiveness of this approach hinges on whether the training data adequately covers the entire state space; insufficient coverage may lead to inaccurate predictions [21], [22].

Table Isummarizes the comparative analysis of representative learning-based techniques. As can be observed, most learning-based data-driven approaches still heavily rely on high-quality training data and necessitate an offline training phase, thereby significantly increasing implementation complexity and computational overhead.

B. Motivation and Innovation

Inspired by the aforementioned studies, this work investigates a novel learning-based MPC approach that eliminates the need for offline training and extensive reliance on pre-collected datasets—common limitations of traditional learning-based methods. Specifically, a radial basis function neural network (RBFNN) is utilized to approximate the ideal control input, with its parameters updated online via gradient descent. This process can be regarded as a form of supervised learning. Moreover, a robust control term is incorporated to actively compensate for disturbances.

The primary achievements of this research are as follows.

1) *Supervised online learning:* The proposed controller approximates the ideal output through an online, supervised process, without requiring an offline learning phase. Additionally, the only parameter that needs adjustment is the learning rate, significantly reducing the workload.
2) *Robustness to uncertainties:* A robust compensation term is designed to mitigate the effects of neural network approximation errors and external disturbances, enhancing the system's disturbance rejection capability.
3) *Model-free predictive control:* The proposed method avoids reliance on motor parameters by directly generating the control voltage through the neural network, thus improving robustness against parameter mismatches.

II. CONVENTIONAL CCS-MPC METHOD

The voltage equations of an interior PMSM in the dq-axes are given by

$$\begin{cases} u_d = L_d \frac{di_d}{dt} + R_s i_d - \omega_e L_q i_q \\ u_q = L_q \frac{di_q}{dt} + R_s i_q + \omega_e (L_d i_d + \psi_f) \end{cases} \tag{1}$$

Where ω_e is the electrical angular velocity, u_d and u_q are the d- and q-axis voltages of the motor, i_d and i_q are the corresponding currents, R_s is the stator resistance, L_d and L_q refer to the d- and q-axis stator inductances, and ψ_f is the rotor flux linkage.

Applying the forward Euler method, the discrete current predictive model of the IPMSM can be obtained. The cost function governing current control is expressed as

$$g = (i_d^* - i_d(k+1))^2 + (i_q^* - i_q(k+1))^2 \tag{2}$$

where i_d^* and i_q^* denote the reference currents of the d- and q-axis currents, and $i_d(k+1)$ and $i_q(k+1)$ are the predictive currents at the $(k+1)$th period. The optimal voltage inputs u_d^* and u_q^* can be obtained by solving $\nabla g = 0$. Since voltage and current constraints are not considered, the optimization reduces to an unconstrained quadratic program, indicating that the CCS-MPC

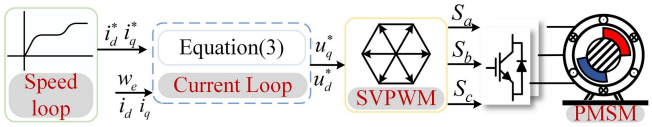

Fig. 1. Schematic diagram of CCS-MPC.

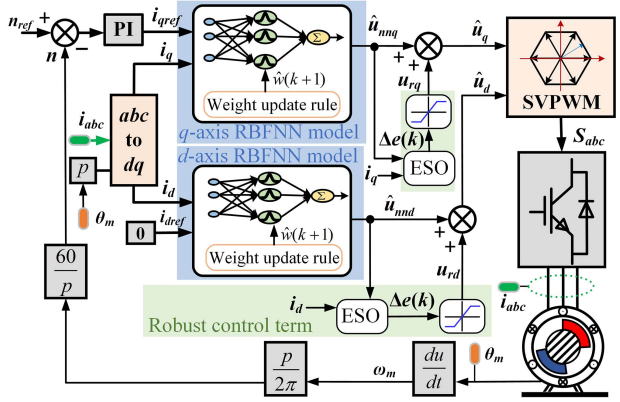

Fig. 2. Schematic diagram of the proposed method.

is an unconstrained MPC approach.

$$\begin{cases} u_d^*(k) = R_s i_d(k) + \frac{L_d}{T_s}(i_d^* - i_d(k)) - \omega_e(k) L_q i_q(k) \\ u_q^*(k) = R_s i_q(k) + \frac{L_q}{T_s}(i_q^* - i_q(k)) + \omega_e(k)(L_d i_d(k) + \psi_f) \end{cases} \tag{3}$$

where T_s is the control period.

Once the optimal voltage inputs are obtained, substitute them into the space vector pulsewidth modulation algorithm to generate the corresponding switching signals, thereby realizing control of the IPMSM. This process is the principle of CCS-MPC, as shown in Fig. 1. While this method offers a straightforward structure and ease of implementation, it is evident that the accuracy of the control input, as expressed in (3), is highly sensitive to the precision of the motor parameters.

III. Proposed SLPC Method

To mitigate the adverse effects caused by motor parameter mismatches, this article proposes a supervised learning predictive control (SLPC) method. The control structure of this method mainly includes three key elements, an RBFNN approximation model, adaptive weight update rules, and a robust control term, as shown in Fig. 2. First, the establishment of the RBFNN approximation model is introduced.

A. Establishment of RBFNN Controller

Taking the d-axis as a representative case, rewrite the d-axis voltage equation as follows:

$$\dot{x} = f(x) + g u_d + d \tag{4}$$

where $x = i_d$, $f(x) = \frac{R_s}{L_d} i_d$, $g = \frac{1}{L_d}$, and d represents the disturbance terms in (1). At time k, by making use of Euler discretization, (4) can be written as

$$x(k+1) = H(k) + G(k) u_d \tag{5}$$

where $G(k) = T_s g(k)$ and $H(k) = x(k) + T_s(f(x(k)) + d(k))$.

Based on the above equations and using the principle of the feedback linearization technique, the desired output is obtained as

$$u^*(k) = \frac{(-H(k) + \gamma e(k) + x^*(k+1))}{G(k)} \tag{6}$$

where $e(k) = x*(k)$-$x(k)$ is the system state error, $x*$ is the reference value of the system state, and γ is a constant set by the user.

In real motor operation, parameters such as inductance and magnetic flux linkage are not constants. Therefore, $H(k)$ and $G(k)$ are considered unknown nonlinear functions, which makes it impossible to directly obtain $u^*(k)$ through (6).

To address this issue, this article employs an RBFNN to approximate $u^*(k)$. Studies have shown that RBFNN possesses a strong approximation capability even in systems with significant uncertainties. First, the ideal output of the network is defined as $u_{nn}^*(k)$, which serves as the ideal approximation of $u^*(k)$ (i.e., assuming $u^*(k) = u_{nn}^*(k)$). By incorporating the neural network's equations, this output can be specifically expressed as

$$u_{nn}^*(k) = \boldsymbol{w}^{*\mathrm{T}} \boldsymbol{L}(\boldsymbol{X}) + \xi(k). \tag{7}$$

Let $\xi(k)$ denote the approximation error, which satisfies $\xi(k) < \bar{\xi}$, where $\bar{\xi}$ is the upper bound of the error, and its value is a normal constant. The ideal weight vector is denoted as $w^* = [w_1^* \; w_2^* \; \cdots \; w_m^*]^{\mathrm{T}}$, with m being the total number of neurons. $\boldsymbol{L}(\boldsymbol{X}) = [l_1(\boldsymbol{X}) \, l_2(\boldsymbol{X}) \cdots l_{\mathrm{m}}(\boldsymbol{X})]^{\mathrm{T}}$ is the hidden layer composed of m known activation functions. Here, the neural network input is $\boldsymbol{X} = e(k)$. Equation (7) indicates that there exists an ideal neural network model that can mathematically represent the ideal input $u^*(k)$. Since the weight matrix and approximation error of the ideal neural network model are unknown, $u_{nn}^*(k)$ cannot be directly obtained from (7). Therefore, $\hat{u}_{nn}(k)$ is defined as the estimated value of $u_{nn}^*(k)$, and $\hat{w}$ as the estimate of $\boldsymbol{w}^*$. Combining the mathematical model of the neural network, $\hat{u}_{nn}(k)$ can be expressed as

$$\hat{u}_{nn}(k) = \hat{\boldsymbol{w}}^T \boldsymbol{L}(\boldsymbol{X}). \tag{8}$$

Equation (8) represents that the established model uses $\hat{w}^{\mathrm{T}} \boldsymbol{L}(\boldsymbol{X})$ to approximate $\boldsymbol{w}^{*\mathrm{T}} \boldsymbol{L}(\boldsymbol{X})$ in (7).

B. Adaptive Weight Update Law

To obtain the update rule of the weight matrix, the tracking error is first defined as follows:

$$\begin{aligned} \tilde{u}_{nn}(k) &= \hat{u}_{nn}(k) - u_{nn}^*(k) \\ &= (\hat{\boldsymbol{w}}^T - \boldsymbol{w}^{*T}) \boldsymbol{L}(\boldsymbol{X}) - \xi(k) \\ &= \tilde{\boldsymbol{w}}^T \boldsymbol{L}(\boldsymbol{X}) - \xi(k) \end{aligned} \tag{9}$$

where $\tilde{\boldsymbol{w}}$ represents the error between the actual weight matrix $\hat{\boldsymbol{w}}$ and the ideal weight matrix w^*. To obtain the adaptive update rule of the weight, define a cost function $J = \frac{1}{2} \tilde{u}_{nn}^2$. In accordance with gradient descent, the weight matrix adjustment rule is

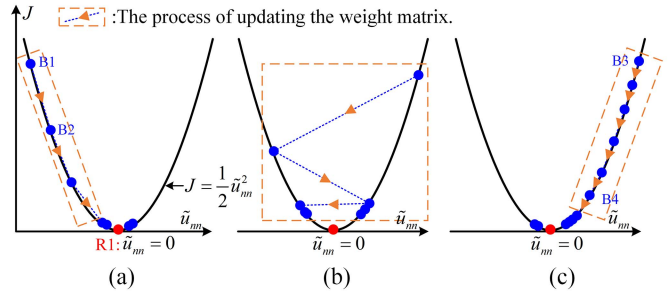

Fig. 3. Gradient descent process under various learning rate settings. (a) Suitable learning rate. (b) Excessive learning rate. (c) Insufficient learning rate.

obtained as

$$\hat{\boldsymbol{w}}(k+1)=\hat{\boldsymbol{w}}(k)-\eta\frac{\partial J}{\partial\hat{\boldsymbol{w}}(k)}$$
$$=\hat{\boldsymbol{w}}(k)-\eta\tilde{u}_{nn}(k)\boldsymbol{L}(\boldsymbol{X}) \tag{10}$$

where η is the learning rate. It is worth noting that, in order to simplify the update rule for the weight matrix, the inherent approximation error in (7) is approximated as zero, and this error will be compensated for in the subsequent robust control term.

Fig. 3(a) intuitively shows the principle of gradient descent. Suppose the error is initially located at point B1, where $\tilde{u}_{nn}<0$. According to (10), the weight matrix should be decreased in the next period, thereby reducing the error, which moves from B1 to B2. This process continues iteratively until the error converges to the reference point R1. Fig. 3(b) shows the scenario with an increased learning rate. As the learning rate grows, the convergence speed improves; however, it also introduces larger oscillations around $\tilde{u}_{nn}=0$, potentially destabilizing the system. In contrast, Fig. 3(c) shows the case of a smaller learning rate. Here, the error decreases from B3 to B4 at a slower pace, resulting in diminished convergence speed and a noticeably weaker dynamic response. In summary, Fig. 3 shows the significant influence of the learning rate on the convergence behavior of the system. To achieve optimal control performance, it is essential to select an appropriately tuned learning rate. The specific value will be determined through experimental validation in the subsequent sections of this study.

As can be seen from (10), at this time, the update law of the weight matrix is still related to the output error, and the system state does not directly participate in the update of the weight matrix. If the system state can be introduced to replace $\tilde{u}_{nn}$ in the weight matrix update rule, online updating of the controller can be achieved. Therefore, this article next deduces the relationship between the output tracking error and the state error. Substituting the optimal input $u_{nn}^*(k)$ into (5) yields the present predicted value based on this input

$$i_{d\_nn}^*(k+1)=\left(1-\frac{R_sT_s}{L_d}\right)i_d(k)+\frac{T_s}{L_d}u_{nn}^*(k)+T_sd(k) \tag{11}$$

where $i_{d\_nn}^*(k+1)=i_d^*(k+1)+\gamma e(k)$.

By substituting the approximate output $\hat{u}_{nn}(k)$ of the neural network into (5), the predicted current value under this input is given by

$$i_d(k+1)=\left(1-\frac{R_sT_s}{L_d}\right)i_d(k)+\frac{T_s}{L_d}\hat{u}_{nn}(k)+T_sd(k). \tag{12}$$

Since the approximate output is the voltage actually applied to the motor, the predicted value from (12) will be the current value at the next step.

Subtracting (12) from (11) yields the relationship between the system state error and the output tracking error, as shown in the following equation:

$$e(k+1)+\gamma e(k)=-\frac{T_s}{L_d}\tilde{u}_{nn}(k). \tag{13}$$

It is assumed that the neural network provides a reliable approximation. Assuming the neural network model provides an effective approximation for the ideal input, the state error at the next time step will be smaller than the state error at time k. At this point, there exists a variable M (where M satisfies $0<M<1$) such that the state error satisfies $e(k+1)=Me(k)$. Thus, (13) can be updated as

$$(M+\gamma)e(k)=-\frac{T_s}{L_d}\tilde{u}_{nn}(k). \tag{14}$$

Inserting (14) into (10) yields the following formulation for the network's weight update rule:

$$\hat{\boldsymbol{w}}(k+1)=\hat{\boldsymbol{w}}(k)+\eta\frac{L_{\mathrm{d}}}{T_s}(M+\gamma)e(k)\boldsymbol{L}(X). \tag{15}$$

Through the derivation of the above formula, the weight matrix update rule is transformed into online learning and update using the system state. For ease of subsequent analysis, the learning rate of the d-axis neural network approximator is defined as η_d, as shown in the following equation:

$$\eta_d=\eta\frac{L_{\mathrm{d}}}{T_s}(M+\gamma). \tag{16}$$

Similarly, the learning rate η_q of the q-axis neural network approximator can be defined by substituting the inductance parameters in (16) with the q-axis inductance parameters. In the experiment, there is no need to calculate η_d using (16) and the inductance parameters. Instead, the corresponding values of η_d and η_q are directly set, thereby achieving MPC without the need for motor parameters.

C. Robust Control Term Output

When deriving the update rule for the neural network weight matrix, the effect of the inherent approximation error was neglected. To address this, a robust control term is introduced in this section to compensate for the error. Experimental results demonstrate that this robust control term not only effectively compensates for the inherent approximation error but also mitigates the influence of external disturbances. Since ESO can accurately estimate the motor state and compensate for external disturbances, it has been widely applied in MFPC of PMSM. Therefore, this article designs an ESO to obtain the data required

for robust control, as shown in the following equation:

$$\begin{cases} e_1 = x - z_1 \\ \dot{z}_1 = az_1 + bu + z_2 - \beta_1 e_1 \\ \dot{z}_2 = z_3 - \beta_2 e_1 \\ \dot{z}_3 = -\beta_3 e_1 \end{cases}. \tag{17}$$

The variables are defined as follows: z_1 is the estimate of state x, z_2 is the observation of d, and z_3 is the observed value of the first derivative of d. β_1, β_2, and β_3 are all coefficients of the observer. $a = \frac{R_s}{L_d}$, $b = -\frac{1}{L_d}$. By discretizing the above equation using the Euler method, we can obtain

$$\begin{cases} e_1(k) = x(k) - z_1(k) \\ z_1(k+1) = (1+aT_s)z_1(k) + T_s(b\hat{u}_{nn}(k) + z_2(k) - \beta_1 e_1(k)) \\ z_2(k+1) = z_2(k) + T_s(z_3(k) - \beta_2 e_1(k)) \\ z_3(k+1) = z_3(k) - T_s\beta_3 e_1(k) \end{cases}. \tag{18}$$

By properly setting the parameters of the observer, $z_1(k+1)$ can be effectively obtained and used in the subsequent process to replace $x(k+1)$ for control. To guarantee observer convergence, the parameter values are set as follows: $\beta_1 = -3w_c - a$, $\beta_2 = -3w_c^2$, and $\beta_3 = -w_c^3$. Among them, w_c is the bandwidth of the observer. In this article, it is designed as $w_c = 2000$.

Through the observer, we can obtain $e(k+1)$ and $e(k)$. Currently, we define the filtering error as $\Theta(k) = e(k+1) + \sigma e(k)$, where σ is the filtering coefficient. The output expression of the robust control term is

$$u_{rd}(k) = \tau \mathrm{sat}(\Theta) \tag{19}$$

where $\mathrm{sat}(\Theta(k)) = \begin{cases} \frac{\Theta(k)}{|\Theta(k)|} \text{ if} |\Theta(k)| > \delta \\ \frac{\Theta(k)}{\delta} \text{ if} |\Theta(k)| \le \delta \end{cases}$, and τ and δ are constant coefficients designed by the user.

The compensatory effect of the robust control term can be understood as follows: when the observer predicts that the next-step current is greater than the reference value (this error arises from neglecting inherent errors during the derivation of the network matrix update), it indicates that the input calculated in (8) is overly large. Therefore, the robust control term applies a negative voltage to compensate for this deviation.

The output of the d-axis is finally defined as

$$\hat{u}_d(k) = \hat{u}_{nn}(k) + u_{rd}(k). \tag{20}$$

Using the same method, the control voltage input $\hat{u}_q(k)$ on the q-axis can also be obtained.

D. Stability Analysis

The first step in the analysis is to determine whether $u^*(k)$ obtained through the feedback linearization technique can stabilize the system. Define the Lyapunov function as $V_1 = \frac{1}{2}e^2$. Taking its derivative yields

$$\dot{V}_1 = \frac{-\lambda - 1}{T_s} e(k)^2. \tag{21}$$

When the value of λ is greater than -1, the system satisfies the Lyapunov stability condition. Therefore, $u^*(k)$ can make the system tend to be stable.

Next, it is proved that the neural network output defined can effectively approximate $u^*(k)$. Taking the derivative of J mentioned earlier, we can obtain

$$\dot{J} = -\frac{\eta}{T_s}\tilde{u}_{nn}^2(k)\boldsymbol{L}^T(\boldsymbol{X})\boldsymbol{L}(\boldsymbol{X}) \le 0. \tag{22}$$

From (22), it can be seen that the weight update rule obtained using the gradient descent method satisfies Lyapunov stability.

Finally, the convergence of the designed observer is demonstrated. The system (4) is rewritten as follows:

$$\begin{cases} \dot{x} = ax + bu + x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = \ddot{\mathrm{d}} \end{cases}. \tag{23}$$

By subtracting (17) from (23) and supposing $e_2 = x\text{-}z_2$ and $e_3 = x\text{-}z_3$, the state-space equation of estimation error is obtained as

$$\begin{cases} \dot{e}_1 = (a + \beta_1)e_1 + e_2 \\ \dot{e}_2 = \beta_2 e_1 + e_3 \\ \dot{e}_3 = \beta_3 e_1 + \ddot{\mathrm{d}} \end{cases}. \tag{24}$$

Therefore, the characteristic equation of (24) is given by

$$s^3 - (a + \beta_1)s^2 - \beta_2 s - \beta_3 = 0. \tag{25}$$

By substituting the previously designed observer parameter values, it can be shown that all three roots of (25) are $-w_c$. Thus, it can be concluded that the designed observer is also convergent.

In the preceding section, the observer bandwidth ω_c was set to 2000. This choice was based on two primary considerations. First, the observer gains β_1, β_2, and β_3 are functions of ω_c; thus, increasing the bandwidth enhances the convergence speed, resulting in a faster response. Second, at this bandwidth, the observer demonstrates a favorable stability margin. As indicated in (25), the convergence of the observer depends on ω_c and the parameter a. In practical systems, parameter variations can shift the poles of the transfer function, potentially affecting stability. In the theoretical derivation, parameter a in (25) is defined as $-Rs/L_d$. However, in experimental implementation, the theoretical value was not directly adopted; instead, a was set to -100 to ensure robust convergence in practice.

To evaluate the stability margin at the chosen bandwidth, a was tested at its theoretical value $-Rs/L_d$, and then adjusted to four times and one-fourth of that value for testing. The resulting pole locations of the observer's transfer function are shown in Fig. 4. The pole trajectory diagram shows that even with variations in the observer parameters, the system maintains convergence.

The experimental results under varying observer parameters are shown in Fig. 5, where i_{dp} and i_{qp} are the predicted current values at the next time step. The results indicate that even in the presence of parameter mismatches, the proposed observer maintains convergence, thereby enabling accurate estimation of the system states.

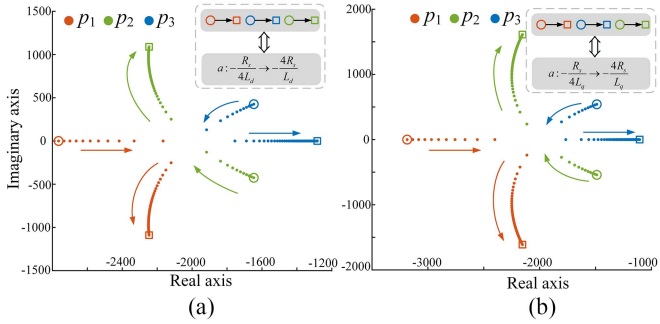

Fig. 4. Pole position trajectory plot. (a) d-axis and (b) q-axis.

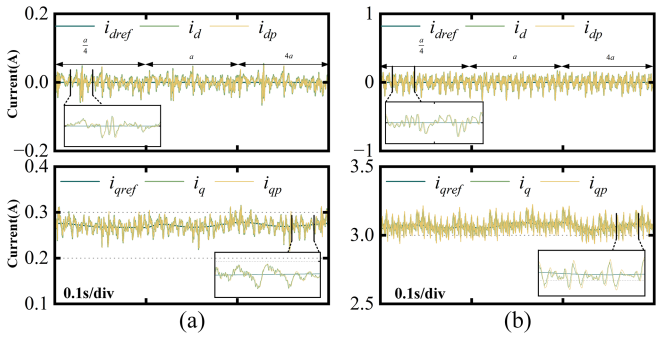

Fig. 5. Analysis of observer convergence under parameter mismatch. (a) Without load. (b) With 2 N·m load.

Remark 1: The robust control terms modify the system output without compromising the system's overall stability.

IV. Experimental Results Analysis

In this section, experiments have been conducted to evaluate the effectiveness of the proposed SLPC strategy. The experiment platform is shown in Fig. 4, where an IPMSM serves as the driving motor. Its specific parameters are detailed in Table I. An induction motor is used as the load motor, driven by a variable frequency drive. The control period is configured as 0.0001 s. To verify the superiority of the proposed method, experiments were carried out to compare four methods: the conventional method, the proposed method, the ultralocal-based method from [14] (which uses an ESO as the disturbance observer, added as Method-I), and the RL-based predictive control method from [24] (which adopts a critic-actor structure, denoted as Method-II). It is worth noting that the implementation of Method-II requires the neural network predictor proposed in [25].

A. Steady Performance Comparison

Experimental evaluations are conducted in this section to verify the steady-state performance of the conventional method, Method-I, Method-II, and the proposed method.

Fig. 7 shows the steady-state performance of the four methods at 200 r/min without load (the subscript "ref" represents the reference value). The current total harmonic distortion (THD) values for the four methods are 5.09%, 4.81%, 4.46%, and 4.04%, respectively.

Fig. 8 further shows the control performance at 1000 r/min under 5 N·m load. As shown in Fig. 8, the current THD values of the four methods are 4.54%, 3.53%, 2.59%, and 2.50%, respectively. Compared to the conventional CCS-MPC approach, the other three methods exhibit improved steady-state performance, with reduced current ripple in the *dq*-axis, indicating better control stability.

The experimental results under both low-speed/no-load and high-speed/rated-load conditions validate that the proposed method exhibits superior steady-state control performance.

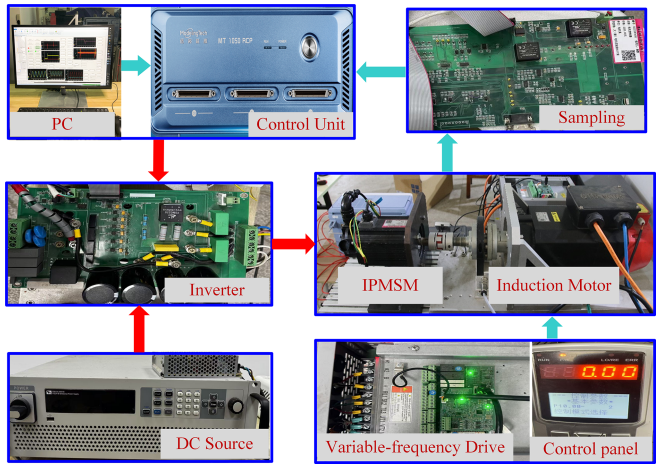

Fig. 6. Experimental platform.

TABLE II
Parameters of the Experimental Setup

Symbol	Parameters	Value
R_s	Phase resistance	0.75 Ω
L_d	*d*-axis inductance	3.5×10^{-3} H
L_q	*q*-axis inductance	9.8×10^{-3} H
ψ_f	Rotor flux linkage	0.142 Wb
Np	Pole pairs	3
J	Rotational inertia	0.0174 kg·m^2
n_N	Rated Speed	1500 r/min
T_e	Rated torque	5 N·m
f_s	Sampling Frequency	10 kHz
f_w	Carrier Frequency	10 kHz

B. Dynamic Performance Comparison

To assess dynamic response characteristics, experiments involving speed transitions are conducted. Fig. 9 shows the performance during a speed ramp from 500 to 1000 r/min. All four methods are capable of effectively tracking the *dq*-axis currents. However, the *q*-axis current peak of the conventional method reaches as high as 9.57 A, while the other three methods exhibit significantly lower peak values of 7.43 A, 8.46 A, and 8.71 A, respectively—indicating a substantial reduction in current overshoot. Furthermore, the proposed method demonstrates superior

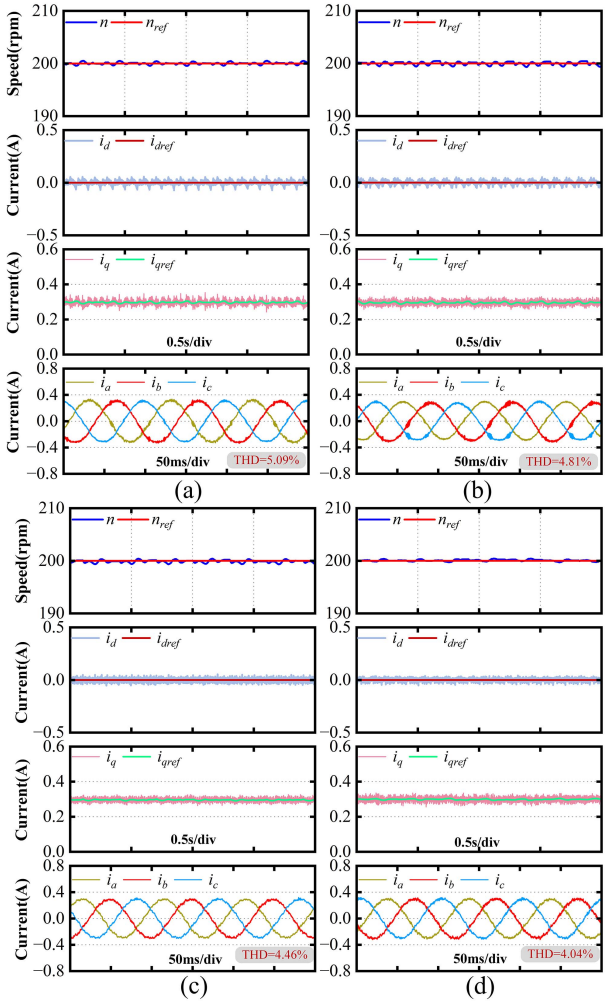

Fig. 7. Experimental results when PMSM operates at 200 r/min without load. (a) CCS-MPC. (b) Method-I. (c) Method-II. (d) Proposed method.

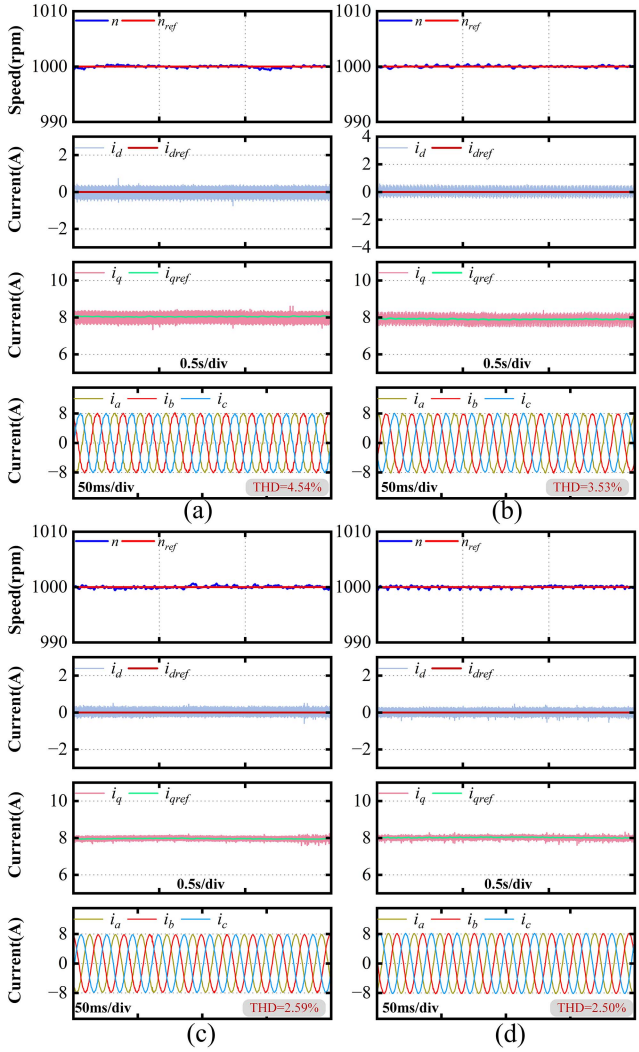

Fig. 8. Experimental results when PMSM operates at 1000 r/min with 5 N·m load. (a) CCS-MPC. (b) Method-I. (c) Method-II. (d) Proposed method.

dynamic performance, with a settling time of 0.669 s, compared to 0.784 s for the conventional method.

The effectiveness of the designed robust term in compensating for external disturbances is shown in Fig. 10. A disturbance of the form $\rho(t) = A\ (1+x)\sin(B\mathrm{t})$, with amplitude $A = 0.06$ and frequency $B = 200\ \pi$, is injected into the phase A current sampling loop. The motor operates at 500 r/min, and the load is raised from 2 to 5 N·m. Under a 2 N·m load condition, the THD values for the four methods are 6.38%, 4.95%, 3.86%, and 3.81%, respectively. When the load increases to 5 N·m, the corresponding THD values are 6.50%, 4.83%, 4.06%, and 3.77%. Owing to their disturbance compensation capabilities, the proposed method and Method-II exhibit superior steady-state performance under external interference compared to the CCS-MPC and Method-I.

C. Control Performance Under Parameter Mismatch

Fig. 11 shows the control performance of the CCS-MPC under parameter mismatches. The IPMSM runs at 500 r/min under a 2 N·m load. To quantitatively assess the control performance, two metrics—current ripple r_z and current fluctuation p_z are defined. These metrics provide a more intuitive reflection of control performance, enabling a clearer comparison between the four methods.

$$r_z = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(i_z^*(t) - i_z(t)\right)^2}\, z \in (d,q) \tag{26}$$

$$p_z = \max(i_z(1), \cdots\cdots, i_z(N)) - \min(i_z(1), \cdots\cdots, i_z(N))\ z \in (d,q) \tag{27}$$

where N represents the sample size.

The performance metrics under parameter mismatch are summarized in Table III. As shown in the table, resistance mismatch has a minimal impact on the conventional method. However, as shown in Fig. 11(b), inductance mismatch results in a significant deterioration in the control effectiveness of the CCS-MPC method. Specifically, under $0.5L_d$ and $0.5\ L_q$, the current ripple r_d increases to 0.193 A, while under $2Ld$ and $2Lq$, the current fluctuation p_d rises to 2.13 A, leading to noticeable current fluctuations. Additionally, while flux linkage mismatch has little effect on the d-axis current performance of the conventional method, it significantly amplifies the q-axis current ripple, as

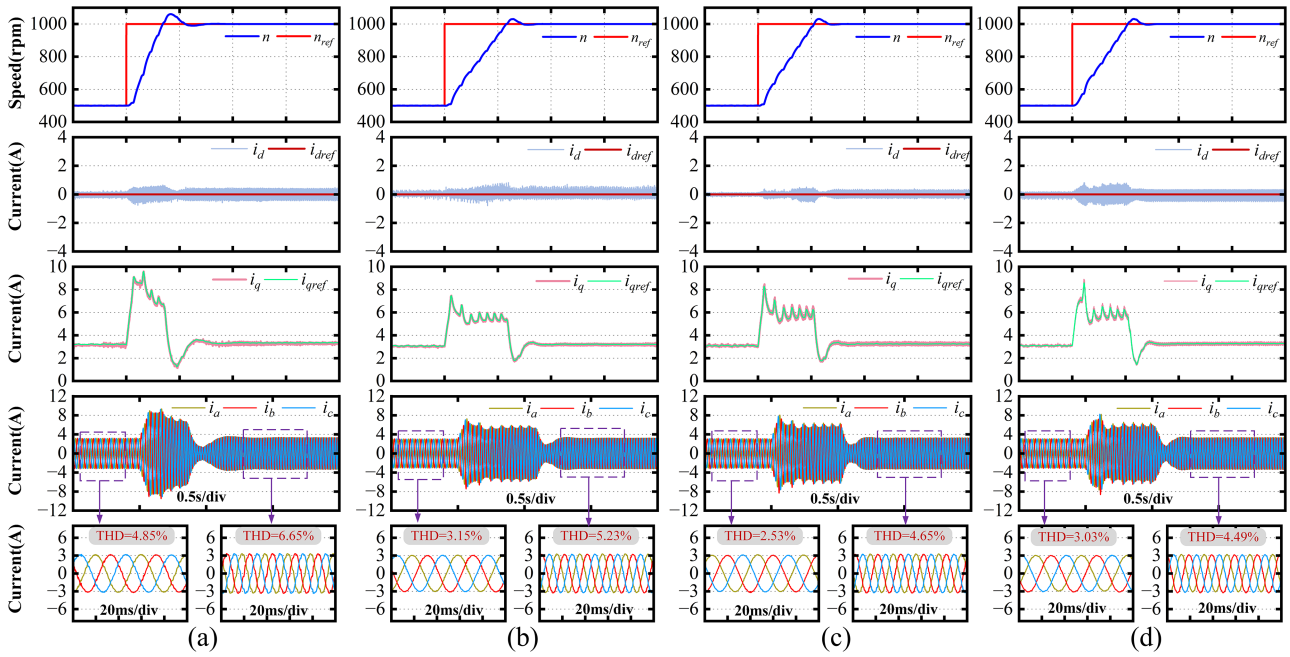

Fig. 9. Control performance when PMSM operates speed increase from 500 to 1000 r/min with 2 N·m load. (a) CCS-MPC. (b) Method-I. (c) Method-II. (d) Proposed method.

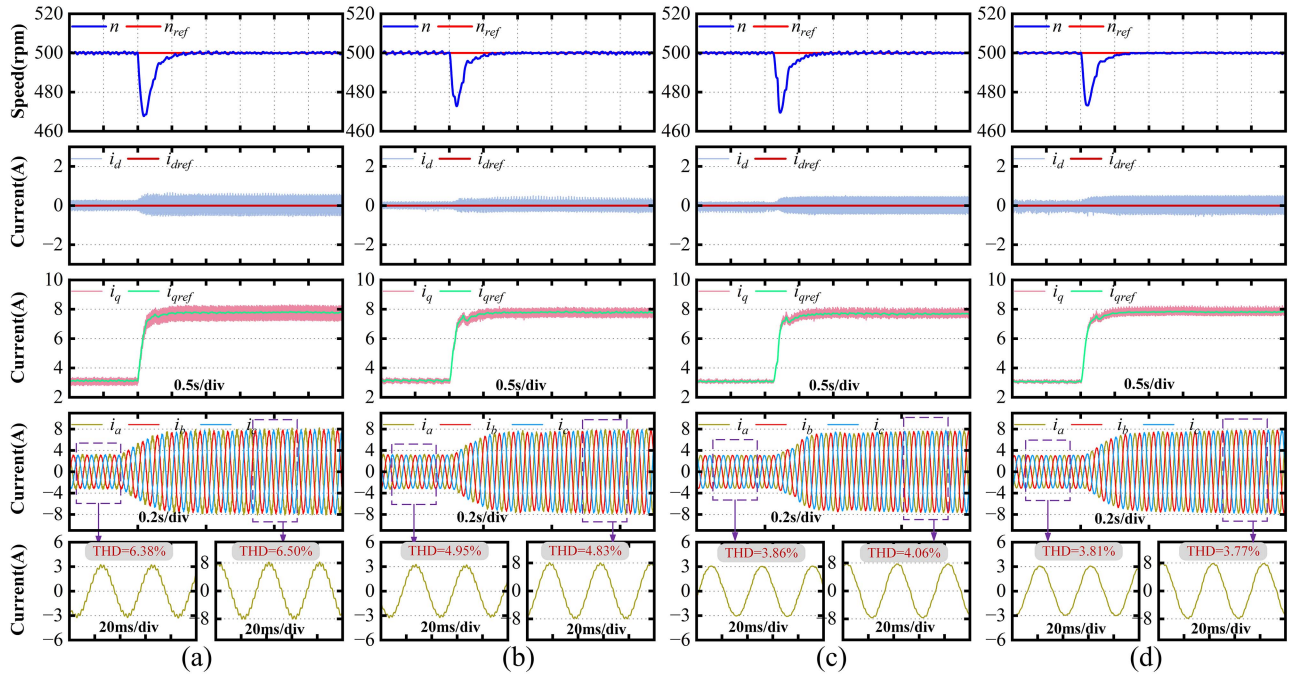

Fig. 10. Control performance when PMSM operates at 500 r/min under sudden load change (2 to 5 N·m) and disturbance. (a) CCS-MPC. (b) Method-I. (c) Method-II. (d) Proposed method.

shown in Fig. 11(c), which reaches as high as 0.617 A, substantially increasing the fluctuation. In addition, Table III evaluates the performance metrics of Method-I and Method-II. Method-I and Method-II are model-free control methods, and their algorithm implementations do not involve motor model parameters, thus eliminating the need for experiments involving changes in motor parameters. Only the corresponding performance metrics are provided under the same operating conditions. The method proposed in this article involves the inductance parameter in the learning rate term, but this parameter is not used in the actual implementation. To ensure fairness in comparison, we simulated the parameter mismatch by multiplying this parameter by the

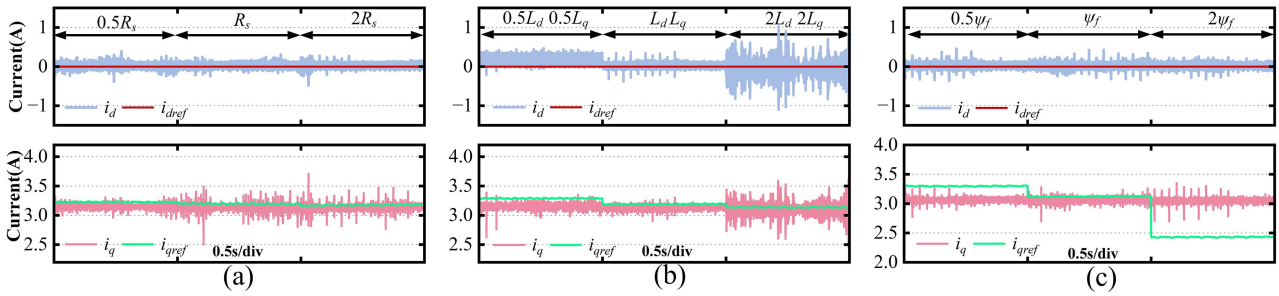

Fig. 11. Control performance under parameter mismatch of conventional method. (a) Under resistance mismatch. (b) Under inductance mismatch. (c) Under flux linkage mismatch.

TABLE III
PERFORMANCE METRICS UNDER PARAMETER MISMATCH

Method	Parameter	r_d(A)	r_q(A)	p_d(A)	p_q(A)
Conventional Method	nominal	0.072	0.083	0.541	1.001
	0.5R_s	0.068	0.076	0.545	0.989
	2 R_s	0.070	0.054	0.060	0.939
	0.5ψ_f	0.071	0.237	0.818	0.734
	2ψ_f	0.070	0.617	0.581	0.752
	0.5L_d 0.5L_q	0.193	0.139	0.681	0.760
	2 L_d 2 L_q	0.190	0.092	2.130	1.156
Method-I	nominal	0.064	0.082	0.443	0.672
Method-II	nominal	0.077	0.042	0.401	0.302
Proposed method	nominal	0.068	0.043	0.394	0.308
	0.5η_d 0.5η_q	0.067	0.043	0.409	0.305
	2η_d 2η_q	0.067	0.044	0.417	0.372

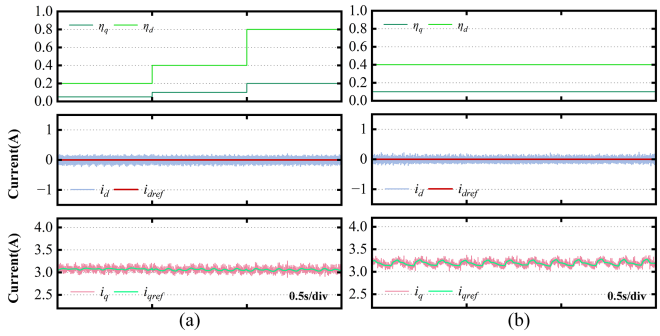

Fig. 12. Verification of the proposed method under inductance parameter mismatch. (a) Control performance under varying learning rate without external inductance. (b) Control performance with external inductance.

corresponding coefficient. The relevant performance metrics are shown in Table III, and the control performance before and after modifying the learning rate is shown in Fig. 12(a). The comparison results clearly demonstrate that the steady-state performance of the proposed method remains unaffected by changes in the learning rate.

To further validate the parameter robustness of the proposed method, the physical parameters of the motor system were modified on the experimental platform, and two sets of experiments were conducted. First, a 3 mH inductor was connected in series between the motor and the inverter to simulate variations in the equivalent inductance. The experimental waveforms under this

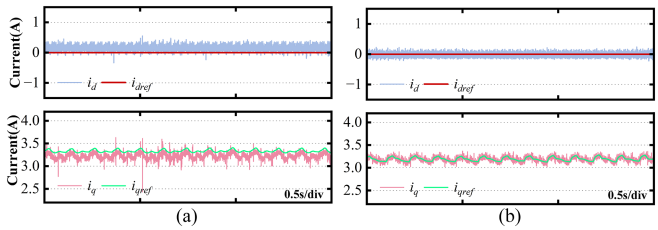

Fig. 13. Control performance with external inductance. (a) CCS-MPC. (b) Proposed method.

condition are shown in Fig. 12, where the controller parameters remained unchanged. The results indicate that the dq-axis current ripples did not increase significantly, with ripple factors of 0.065 and 0.041, compared to 0.068 and 0.043 without the inductor, which are nearly identical.

In addition, the control performance of the conventional method and the proposed method was compared under the condition of an external inductor. In this comparison, the current loop of the conventional method was designed based on the nominal parameters of the motor, and the PI regulator parameters were kept identical to those of the proposed method. As shown in Fig. 13, when the external inductor is connected, the dq-axis currents of the conventional method fail to achieve effective tracking. Specifically, the ripple coefficients of the d-axis and q-axis currents are 0.182 and 0.112, respectively. In contrast, the proposed method achieves significantly lower ripple coefficients of 0.065 and 0.041 for the d- and q-axis currents, respectively, demonstrating superior control performance.

Second, since the flux linkage of the permanent magnets in the test motor cannot be directly adjusted, it was indirectly varied by raising the motor operating temperature. The control performance before and after the temperature change is shown in Fig. 14. The experimental waveforms demonstrate that the d-axis current tracking is not affected by the temperature variation, as the d-axis voltage equation does not involve the flux linkage parameter. In contrast, the q-axis current increases because the flux linkage decreases with rising temperature, requiring a higher q-axis current to balance the electromagnetic torque with the load torque. Importantly, even though the current magnitude increases, the proposed method still ensures effective current tracking. Specifically, the ripple factors of the d-axis current

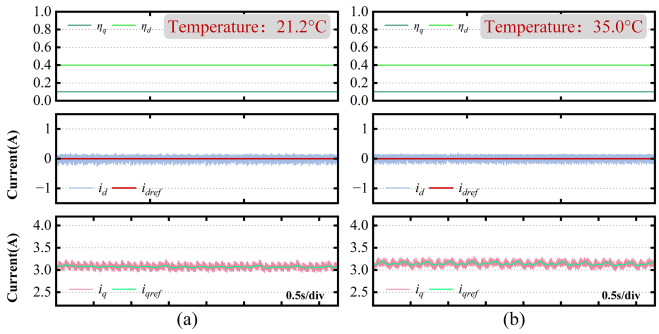

Fig. 14. Current tracking performance with temperature variation. (a) 21.2 °C. (b) 35 °C.

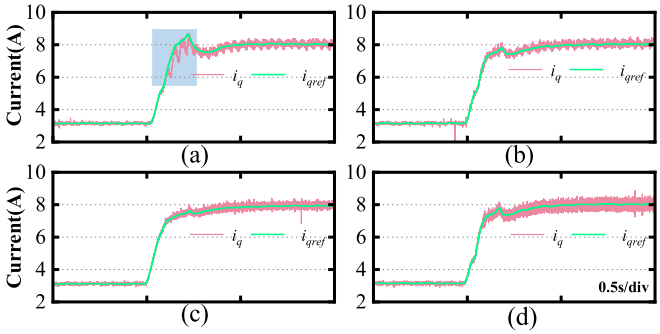

Fig. 15. Dynamic response results under different q-axis learning rates (load from 2 to 5 N·m). (a) $\eta_q = 0.001$. (b) $\eta_q = 0.01$. (c) $\eta_q = 0.1$. (d) $\eta_q = 1$.

before and after the temperature change were 0.065 and 0.066, respectively, while those of the q-axis current were 0.043 and 0.044, showing almost no difference.

The above experimental results demonstrate that the proposed method can still achieve effective current tracking under variations in both controller parameters and motor physical parameters, thereby exhibiting strong parameter robustness.

D. Control Performance Under Different Learning Rates and Different Numbers of Neurons

The previous experiments demonstrated the effectiveness of the SLPC method in steady-state performance, dynamic response, disturbance rejection, and parameter mismatch. However, an in-depth analysis of the proposed method's internal structural characteristics is still lacking. The following experiments focus on evaluating the impact of two key parameters—learning rate and the number of neurons—on motor control performance.

As shown in Fig. 15, with the learning rate of the d-axis neural network fixed, increasing the learning rate of the q-axis network reduces the overall dynamic response time. This result aligns with previously obtained findings. Specifically, the dynamic response times for four different learning rates are 0.546 s, 0.525 s, 0.489 s, and 0.479 s, respectively. At lower learning rates, the motor's dynamic response slows, and there are instances where the current fails to effectively track the desired values, leading to poorer control performance, as shown in the blue-shaded area of Fig. 15(a). As the learning rate increases, the dynamic response time improves.

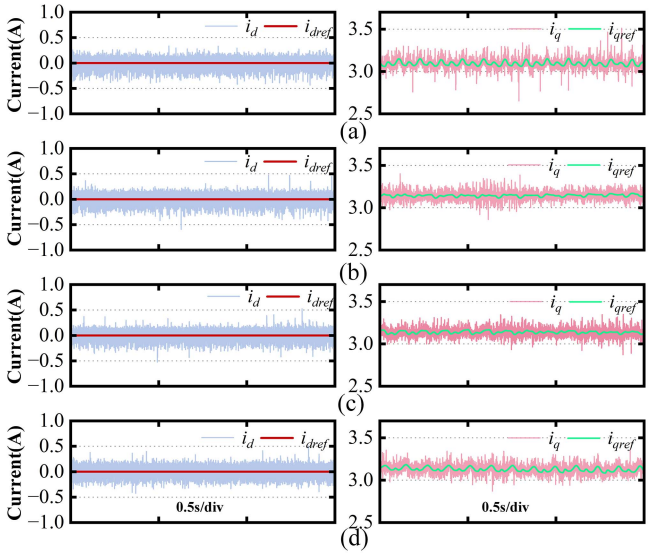

Fig. 16. Current tracking performance of the dq-axes under different numbers of neurons. (a) $m = 5$. (b) $m = 10$. (c) $m = 15$. (d) $m = 20$.

However, the waveforms shown in Fig. 15(c) and (d) reveal that while increasing the learning rate accelerates the dynamic response, it also induces greater fluctuation in the q-axis current. Specifically, the r_q values are 0.048 and 0.122 for the two cases. Based on these observations, the optimal learning rate can be determined experimentally. As shown in the analysis, only a few experiments are necessary to accurately determine this parameter, demonstrating the simplicity and efficiency of the proposed method in terms of parameter tuning.

When using neural networks for approximation, the number of neurons is typically selected to be between 5 and 20 [23]. To determine the optimal number of neurons, experiments are conducted to evaluate the current tracking performance under different neuron counts. Fig. 16 shows the experimental results. For the d-axis current tracking, no significant difference in tracking performance is observed as the number of neurons m increases. To reduce computational complexity, the number of neurons in the d-axis neural network should be set to 5. When the number of q-axis neurons increases from 5 to 10, the current ripple indices r_q and p_q decrease from 0.049 and 0.855 to 0.043 and 0.490, respectively. Further increasing the number of neurons to 15 results in $r_q = 0.044$ and $p_q = 0.476$. These results indicate that, compared to the case with 10 neurons, the performance improvement becomes marginal when the number of neurons is further increased. Therefore, for the experiments, the number of neurons in the q-axis neural network should be set to 10. Hence, for the experimental setup, the q-axis neural network is configured with 10 neurons.

The experimental results (as shown in Fig. 16) indicate that the control performance of the proposed method shows no significant correlation with the number of neurons. The main reasons

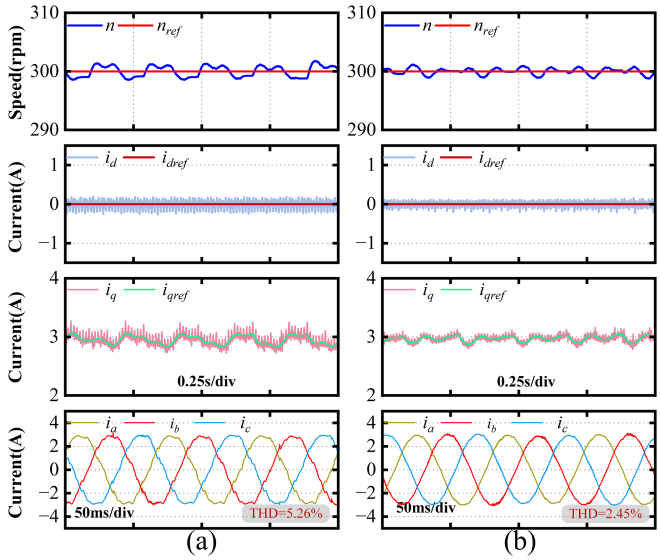

Fig. 17. Comparison of the impact of the robust control term on control performance. (a) Without robust control term. (b) With robust control term.

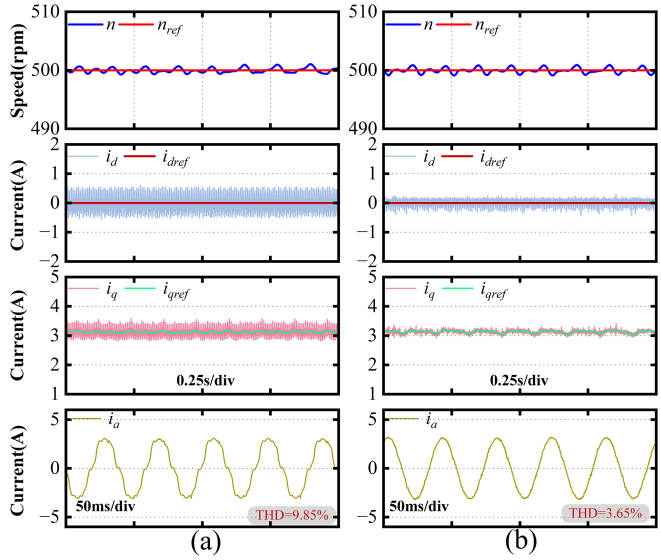

Fig. 18. Comparison of the impact of the robust control term on control performance under disturbance injection. (a) Without robust control term. (b) With robust control term.

are as follows: first, the proposed method is based on online learning, where the network is updated solely using sampling information from the current time step. Due to the limited amount of available data, increasing the number of neurons can easily lead to parameter redundancy and a higher computational burden, making it difficult to achieve further performance improvement. Second, for the sake of implementation simplicity, a fixed learning rate is adopted in this study. As the network size increases, the lack of adaptive adjustment in the learning rate may restrict the optimization process, preventing the additional neurons from fully exploiting their potential.

Based on our experimental observations, a practical guideline is to start with a small learning rate (e.g., 0.0001) to ensure stability, then gradually increase it while monitoring dynamic performance, stopping the adjustment when the optimal trade-off between response speed and steady-state performance is achieved. For the number of neurons, start with 5 and incrementally increase until further improvements in steady-state performance become insignificant.

E. Control Performance Without Robust Control Term

To assess the role of the robust control term relative to the neural network approximator, this article analyzes the control performance without the robust control term. As shown in Fig. 17, after removing the robust control term, the current THD increases from 2.45% to 5.26%. As shown in Fig. 18, under external disturbance injection, after removing the robust term, the A-phase current THD increases from 3.65% to 9.55%. These results demonstrate that the robust control term can effectively compensate for both the inherent approximation errors in the neural network and the external disturbances.

TABLE IV
COMPUTATIONAL TIME COMPARISON OF FOUR METHODS

Method	CCS +MPC	Method I	Method II	Proposed method
Computational time	20.93 μs	22.77 μs	40.29 μs	26.55 μs

F. Comparison of the Four Methods

This section presents a comparative analysis of the four methods from multiple perspectives. Table IV details the computation time of each method. The conventional method, owing to its simple structure, has the shortest execution time. The RL-based Method-II, due to the incorporation of multiple neural networks, has the longest computation time. The proposed method has a computation time of 26.55 μs, placing it in the midrange among the four methods.

In terms of parameter robustness, the proposed method, Method-I, and Method-II all demonstrate strong robustness, since they do not rely on motor model parameters. However, due to the lack of dedicated disturbance rejection mechanisms, both the conventional method and Method-I exhibit weaker disturbance rejection capabilities.

A detailed performance comparison of the four methods is presented in Fig. 19. Overall, the proposed method performs excellently in improving parameter robustness and disturbance rejection. Meanwhile, its structure remains simple, with a weight matrix update process straightforward to derive and implement.

G. Scalability of the Proposed Method

In addition, to verify the scalability of the proposed method, leading angle flux-weakening control experiments and tests under higher load torque conditions were added in this section.

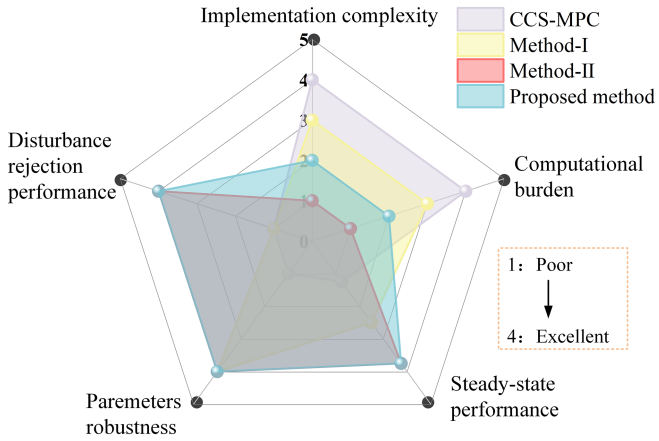

Fig. 19. Comparison of the four methods.

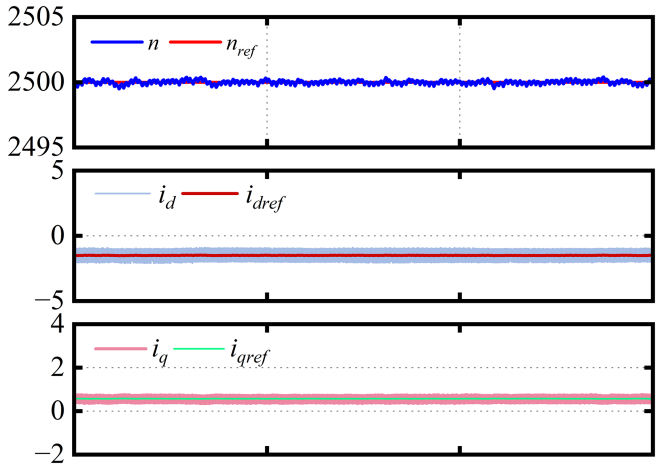

Fig. 20. Experimental results at 2500 r/min under leading angle flux-weakening control.

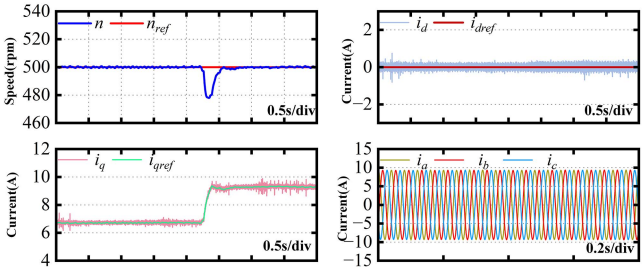

Fig. 21. Overload test performance of the proposed method.

Flux-weakening experiment results are shown in Fig. 20. After applying the flux-weakening algorithm, the motor is able to operate stably at 2500 r/min, with both current and speed achieving excellent tracking performance.

The results of the overload test are shown in Fig. 21, where the motor load torque was increased to 6 N·m, exceeding the rated load by 20%. As observed from the experimental waveforms, the

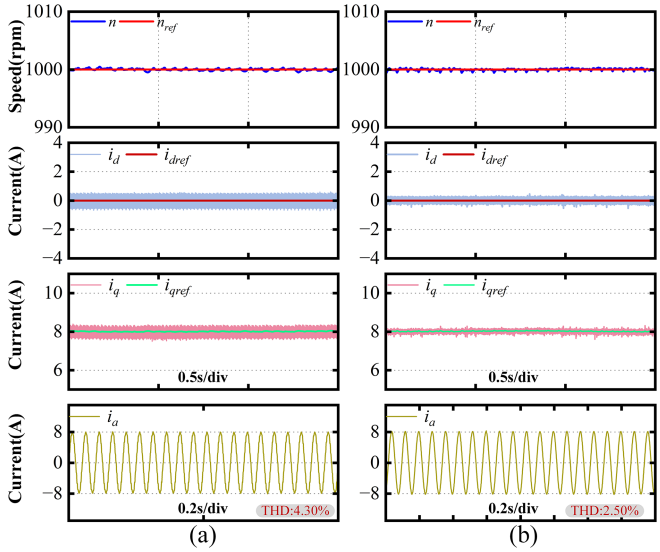

Fig. 22. Experimental results when PMSM operates at 1000 r/min with 5 N·m load. (a) FOC. (b) Proposed method.

proposed method still achieves effective tracking of both speed and current.

H. Performance Comparison With FOC

The practical significance of the proposed method is further enhanced by comparing it with the widely implemented field-oriented control (FOC) technique in industrial applications.

Under the same experimental platform with a 1000 r/min rated load, the steady-state performance of both methods is shown in Fig. 22. The experimental results indicate that the proposed method exhibits better steady-state performance and lower current ripple. The THD of the current for FOC is 4.30%, which is close to that of CCS-MPC, while the THD of the proposed method is only 2.50%.

The speed step test for both methods is shown in Fig. 23 . When the motor speed increases from 500 to 1000 r/min, the *dq*-axis current of the FOC method tracks effectively, but due to the limitation of the current inner loop bandwidth, its response time is 1.07 s, while the proposed method has a faster response time of 0.669 s.

A disturbance of the form $\rho(t) = A\ (1+x)\ \sin\ (B\mathrm{t})$, with amplitude $A = 0.06$ and frequency $B = 200\pi$, is injected into the phase A current sampling loop. When the external load increases from 2 to 5 N·m, both methods can adapt to the load change. However, with external disturbance signals injected, the proposed method shows better control performance, with the current THD before and after the load change being 3.81% and 3.77%, respectively. In contrast, the FOC method, lacking a compensatory loop, shows poorer control performance, with current THD values before and after the load change being 6.39% and 6.45%, respectively.

Additionally, a comparison of computational load and implementation difficulty is also necessary. The proposed method involves the design of a neural network and corresponding

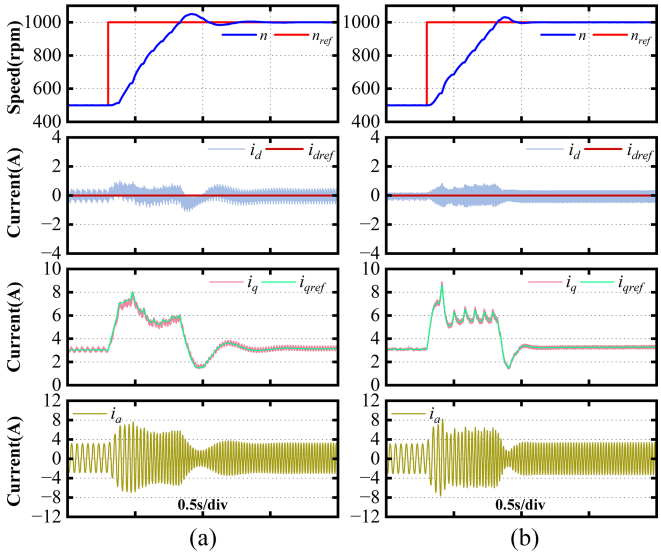

Fig. 23. Control performance when PMSM operates speed increase from 500 to 1000 r/min with 2 N·m load. (a) FOC. (b) Proposed method.

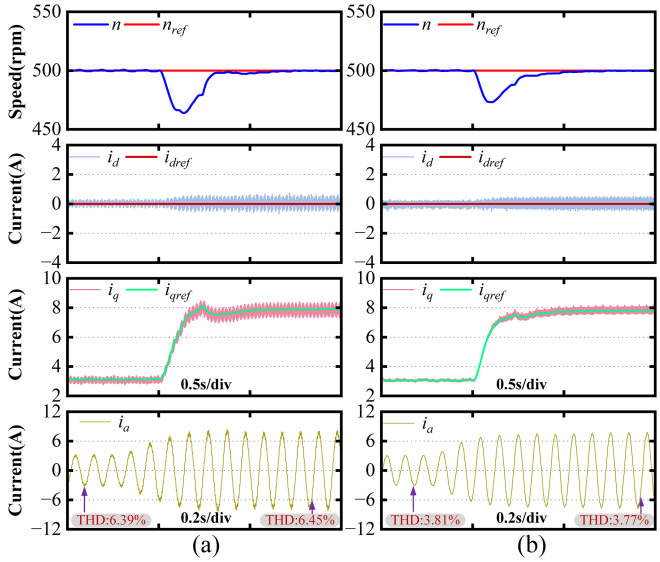

Fig. 24. Control performance when PMSM operates at 500 r/min under sudden load change (2–5 N·m) and disturbance. (a) FOC. (b) Proposed method.

compensation terms, resulting in a computation time of 26.55 μs, while the FOC method only requires a PI regulator, with a computation time of 20.12 μs, making it simpler in structure. Although the proposed method has a slightly increased calculation time and a more complex control structure, the significant improvements in steady-state performance, dynamic response speed, and anti-interference performance make this trade-off worthwhile.

V. Conclusion

A supervised learning-based CCS-MPC approach is proposed in this article for controlling PMSM. The proposed method systematically derives the mapping relationship between tracking error and state error, incorporating the state error into the weight update law to replace the traditional tracking error term. This approach enables the online adaptive updating of the RBFNN based on current sampling data. To address the inherent error in the neural network approximation process and the impact of external disturbances, a robust compensation term is further designed to enhance the system's control performance. Experimental results indicate that the SLPC approach exhibits enhanced control performance in the presence of parameter mismatches and external disturbances.

Appendix

This appendix section provides a stability proof for the system combining neural networks, robust control term, and the dynamic characteristics of the motor. Under the composite voltage action of the neural network and robust control term, the current prediction equation of the PMSM can be expressed as

$$i_d(k+1) = \left(1 - \frac{R_s T_s}{L_d}\right) i_d(k) + \frac{T_s}{L_d}\left(\hat{u}_{nn}(k) + u_{rd}(k)\right) + T_s d(k). \tag{28}$$

Subtracting (11) from (28) yields

$$e(k+1) + \gamma e(k) = \frac{T_s}{L_d}(u^*_{nn}(k) - \hat{u}_{nn}(k) - u_{rd}(k)). \tag{29}$$

Substituting (14) and (19) into (29) yields a simplified expression given as follows:

$$e(k+1) = \frac{(M - \frac{T_s}{L_d}\frac{\tau}{\delta}\sigma)e(k)}{(1 + \frac{T_s}{L_d}\frac{\tau}{\delta})}. \tag{30}$$

To prove the stability of the entire system, we first define the Lyapunov function of the composite system as

$$V = \frac{1}{2}e^2 + \frac{1}{2}\tilde{u}^2_{nn}. \tag{31}$$

Taking the derivative of (32) yields the following expression:

$$\dot{V} = \frac{e(k+1) - e(k)}{T_s} e(k) + \tilde{u}_{nn}(k) \frac{\partial(\hat{u}_{nn}(k) - u^*_{nn}(k))}{\partial \hat{\boldsymbol{w}}} \frac{\partial \hat{\boldsymbol{w}}}{\partial T_s}. \tag{32}$$

Substituting (22) and (30) into (32) yields the simplified expression

$$\begin{aligned}\dot{V} &= \frac{1}{T_s}\left(\frac{(M - \frac{T_s}{L_d}\frac{\tau}{\delta}\sigma)}{(1 + \frac{T_s}{L_d}\frac{\tau}{\delta})} - 1\right) e^2(k) - \frac{\eta}{T_s}\tilde{u}^2_{nn}(k)\boldsymbol{L}^T(\boldsymbol{X})\boldsymbol{L}(\boldsymbol{X}) \\ &< \frac{1}{T_s}\left(\frac{(1 - \frac{T_s}{L_d}\frac{\tau}{\delta})}{(1 + \frac{T_s}{L_d}\frac{\tau}{\delta})} - 1\right) e^2(k) - \frac{\eta}{T_s}\tilde{u}^2_{nn}(k)\boldsymbol{L}^T(\boldsymbol{X})\boldsymbol{L}(\boldsymbol{X}) \\ &= \frac{1}{T_s}\frac{-2\frac{T_s}{L_d}\frac{\tau}{\delta}}{(1 + \frac{T_s}{L_d}\frac{\tau}{\delta})} e^2(k) - \frac{\eta}{T_s}\tilde{u}^2_{nn}(k)\boldsymbol{L}^T(\boldsymbol{X})\boldsymbol{L}(\boldsymbol{X}) < 0.\end{aligned} \tag{33}$$

In (33), the d-axis inductance L_d, the interruption period T_s, the learning rate η, and the parameters δ and τ of the robust control term are all positive. Therefore, it follows that $\cdot V < 0$. Furthermore, since $V > 0$, according to the Lyapunov stability

criterion, the overall motor system can achieve bounded stability under the combined action of the neural network input and the robust control component.

REFERENCES

[1] Y. Luo, K. Yang, and Y. Zheng, "Luenberger observer-based model predictive control for six-phase PMSM motor with localization error compensation," *IEEE Trans. Ind. Electron.*, vol. 70, no. 11, pp. 10800–10810, Nov. 2023.

[2] J. Rodriguez et al., "Latest advances of model predictive control in electrical drives—Part II: Applications and benchmarking with classical control methods," *IEEE Trans. Power Electron.*, vol. 37, no. 5, pp. 5047–5061, May 2022.

[3] J. Rodriguez et al., "Latest advances of model predictive control in electrical drives—Part I: Basic concepts and advanced strategies," *IEEE Trans. Power Electron.*, vol. 37, no. 4, pp. 3927–3942, Apr. 2022.

[4] X. Liu, L. Qiu, Y. Fang, K. Wang, Y. Li, and J. Rodríguez, "Finite control-set learning predictive control for power converters," *IEEE Trans. Ind. Electron.*, vol. 71, no. 7, pp. 8190–8196, Jul. 2024.

[5] L. Wang, S. Zhang, C. Zhang, and Y. Zhou, "An improved deadbeat predictive current control based on parameter identification for PMSM," *IEEE Trans. Transp. Electrific.*, vol. 10, no. 2, pp. 2740–2753, Jun. 2024.

[6] I. Omrane, E. Etien, O. Bachelier, and W. Dib, "A simplified least squares identification of permanent magnet synchronous motor parameters at standstill," in *Proc. 39th Annu. Conf. IEEE Ind. Electron. Soc.*, 2013, pp. 2578–2583.

[7] M. Yang et al., "Nonparametric predictive current control for SPMSM with adaptive cascade extended noise state observer," *IEEE Trans. Power Electron.*, vol. 40, no. 1, pp. 1717–1727, Jan. 2025.

[8] T. Li, X. Sun, M. Yao, D. Guo, and Y. Sun, "Improved finite control set model predictive current control for permanent magnet synchronous motor with sliding mode observer," *IEEE Trans. Transp. Electrific.*, vol. 10, no. 1, pp. 699–710, Mar. 2024.

[9] N. Yang, S. Zhang, X. Li, and X. Li, "A new model-free deadbeat predictive current control for PMSM using parameter-free Luenberger disturbance observer," *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 11, no. 1, pp. 407–417, Feb. 2023.

[10] A. Brosch, S. Hanke, O. Wallscheid, and J. Böcker, "Data-driven recursive least squares estimation for model predictive current control of permanent magnet synchronous motors," *IEEE Trans. Power Electron.*, vol. 36, no. 2, pp. 2179–2190, Feb. 2021.

[11] X. Zhang, X. Yu, and G. Zhang, "Adaptive model predictive current control for PMSM drives based on Bayesian inference," *IEEE Trans. Power Electron.*, vol. 40, no. 6, pp. 8490–8502, Jun. 2025.

[12] F. Wang, Y. Wei, H. Young, D. Ke, D. Huang, and J. Rodrıguez, "Continuous-control-set model-free predictive control using time-series subspace for PMSM drives," *IEEE Trans. Ind. Electron.*, vol. 71, no. 7, pp. 6656–6666, Jul. 2024.

[13] B. Luo, X. Yang, and Y. Zhou, "Model-free predictive current control of permanent magnet synchronous motor based on estimation of current variations," *IEEE Trans. Ind. Electron.*, vol. 71, no. 8, pp. 8395–8405, Aug. 2024.

[14] Y. Zhang, J. Jin, and L. Huang, "Model-free predictive current control of PMSM drives based on extended state observer using ultralocal model," *IEEE Trans. Ind. Electron.*, vol. 68, no. 2, pp. 993–1003, Feb. 2021.

[15] Y. Wei, H. Young, F. Wang, and J. Rodríguez, "Generalized data-driven model-free predictive control for electrical drive systems," *IEEE Trans. Ind. Electron.*, vol. 70, no. 8, pp. 7642–7652, Aug. 2023.

[16] Y. Wan, Q. Xu, and T. Dragičević, "Reinforcement learning-based predictive control for power electronic converters," *IEEE Trans. Ind. Electron.*, vol. 72, no. 5, pp. 5353–5364, May 2025.

[17] M. Schenke, W. Kirchgässner, and O. Wallscheid, "Controller design for electrical drives by deep reinforcement learning: A proof of concept," *IEEE Trans. Ind. Inform.*, vol. 16, no. 7, pp. 4650–4658, Jul. 2020.

[18] N. Farah, G. Lei, J. Zhu, and Y. Guo, "Robust model-free reinforcement learning based current control of PMSM drives," *IEEE Trans. Transp. Electrific.*, vol. 11, no. 1, pp. 1061–1076, Feb. 2025.

[19] M. Abu-Ali, F. Berkel, M. Manderla, S. Reimann, R. Kennel, and M. Abdelrahem, "Deep learning-based long-horizon MPC: Robust, high performing, and computationally efficient control for PMSM drives," *IEEE Trans. Power Electron.*, vol. 37, no. 10, pp. 12486–12501, Oct. 2022.

[20] W. Mei, X. Wang, Y. Lu, K. Yu, and S. Li, "Learning and current prediction of PMSM drive via differential neural networks," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 72, no. 3, pp. 489–493, Mar. 2025.

[21] M. Novak and T. Dragicevic, "Supervised imitation learning of finite-set model predictive control systems for power electronics," *IEEE Trans. Ind. Electron.*, vol. 68, no. 2, pp. 1717–1723, Feb. 2021.

[22] I. S. Mohamed, S. Rovetta, T. D. Do, T. Dragicević, and A. A. Z. Diab, "A neural-network-based model predictive control of three-phase inverter with an output LC filter," *IEEE Access*, vol. 7, pp. 124737–124749, 2019.

[23] H. Chaoui, M. Khayamy, and O. Okoye, "Adaptive RBF network based direct voltage control for interior PMSM based vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 5740–5749, Jul. 2018.

[24] X. Liu, L. Qiu, Y. Fang, K. Wang, Y. Li, and J. Rodríguez, "Event-driven based reinforcement learning predictive controller design for three-phase NPC converters using online approximators," *IEEE Trans. Power Electron.*, vol. 40, no. 4, pp. 4914–4926, Apr. 2025.

[25] X. Liu, L. Qiu, Y. Fang, and J. Rodríguez, "Predictor-based data-driven model-free adaptive predictive control of power converters using machine learning," *IEEE Trans. Ind. Electron.*, vol. 70, no. 8, pp. 7591–7603, Aug. 2023.

Sen Hong was born in Chongqing, China, in 2002. He received the B.S. degree in electrical engineering from Sichuan University, Chengdu, China, in 2024. He is currently working toward the M.S. degree in electrical engineering with Huazhong University of Science and Technology, Wuhan, China.

His research interests include permanent magnet motor drives and power electronics.

Yixiao Luo (Member, IEEE) was born in Hubei, China, in 1991. He received the B.Eng. degree in electrical engineering from Wuhan University, Wuhan, China, in 2013, the M.Eng. degree in electrical engineering from Hanyang University, Seoul, South Korea, in 2015, and the Ph.D. degree in electrical engineering from the City University of Hong Kong, Hong Kong, in 2019.

Since 2022, he has been an Associate Professor with the School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan, China. His research interests include permanent magnet motor drives, renewable energy sources, and microgrids.

Kai Yang (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1998, 2000, and 2003, respectively.

In 2005, he was with the Department of Electrical Machine and Control Engineering, School of Electrical and Electronic Engineering, HUST. He is currently a Professor with the School of Electrical and Electronic Engineering, HUST. He has authored and coauthored more than 100 technical papers on IEEE conferences and journals, and has 30 authorized Chinese patents. His research interests include functional-material actuators, design, and control of ac motors.

Dr. Yang has been a Fellow of the IET since 2023. He is a technical reviewer for IEEE TRANSACTIONS ON POWER ELECTRONICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS. He was the Secretary General for ICEMS 2008 and the Topic Chair for ICEMS 2016/2017/2018/2019.

Shuangxia Niu (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Tianjin University, Tianjin, China, in 2002 and 2005, respectively, and the Ph.D. degree in electrical engineering from The University of Hong Kong, Hong Kong, in 2009.

She is currently working as a Professor with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong, SAR. Her research interests include electric vehicle technologies, renewable energy systems, machines and drives, power electronics technology, applied electromagnetic, numerical methods, and optimization.

Jincheng Yu (Member, IEEE) received the B.E. degree in electrical engineering from Beijing Jiaotong University, Beijing, China, in 2016, and the Ph.D. degree in electrical and electronic engineering from the School of Energy and Environment, City University of Hong Kong, Hong Kong, in 2020.

She is currently an Assistant Professor with the Department of Electrical Engineering, Harbin Institute of Technology, Shenzhen, China. Her research interests include machine design and control and sustainable energy conversion technologies.