

Permanent Magnet Synchronous Magnet Motor Intelligent Parameter Estimation Using Machine Learning Based Nonlinear Control

Eliyab Yosef Delelew

Jimma University

Kejela Adane Dulecha

Addis Ababa University

Chala Merga Abdissa

`chala.merga@aaau.edu.et`

Addis Ababa University

Article

Keywords:

Posted Date: October 8th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10876667/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Permanent Magnet Synchronous Magnet Motor Intelligent Parameter Estimation Using Machine Learning Based Nonlinear Control.

Eliyab Yosef Delelew¹, Kejela Adane Dulecha², and Chala Merga Abdissa^{3,*}

¹Jimma Institute of Technology, Faculty of Electrical and Computer Engineering, Jimma, 378, Ethiopia

²Jimma Institute of Technology, Faculty of Electrical and Computer Engineering, Jimma, 378, Ethiopia

³Addis Ababa University, School of Electrical and Computer Engineering, Addis Ababa, 1000, Ethiopia

*chala.merga@aau.edu.et

[†]these authors contributed equally to this work

ABSTRACT

Modern industry has been witnessing the wide adoption of Permanent Magnet Synchronous Motors (PMSMs) for their high power density and efficiency in applications such as transportation. However, it is well-known that nonlinearity and the sensitivity to variation of parameters in PMSM dynamics pose challenges to developing a highly efficient controller. Moreover, parameters' variation caused by temperature rise, magnetic saturation, and aging of components is a problem in many control problems. In this paper, three different intelligent parameter estimation approaches that can be implemented alongside the control strategies are suggested and compared: Recurrent Neural Network based Online Parameter Estimation for Model Predictive Control (RNN-MPC), Reinforcement Learning-based Online Parameter Estimation for Model Predictive Control (RL-MPC), and Reinforcement Learning-based Online Parameter Estimation for Sliding Mode Control (RL-SMC). All of the above frameworks suggest estimating the following PMSM parameters: stator resistance (R_s), d -axis inductance (L_d), q -axis inductance (L_q), permanent magnet flux linkage (ψ_f), and rotor inertia (J) under various speed and load conditions. Specifically, the RNN estimator uses Long Short-Term Memory architecture trained offline using operational data, while reinforcement learning estimators implement the Twin Delayed Deep Deterministic Policy Gradient algorithm to adapt parameters online. The simulation results show that all methods provide successful parameter estimation with an error level less than 3.5% for electrical parameters and 0.5% for mechanical ones. RNN based online parameter estimation for MPC demonstrates the smoothest parameter trajectories and minimum torque ripple (1.0 N-m peak-to-peak), whereas RL based online parameter estimation for MPC gives the highest transient performance with rise times equal to 0.12–0.18s and low overshoot. Boundedness of the estimation error and closed-loop system's stability are proven using the Lyapunov-based approach. Based on the findings, it is clear that intelligent parameter estimation improves the performance of PMSM controllers. The RNN based online parameter estimation for MPC shows the best smoothness, RL-MPC shows the fastest adaptation and RL based online parameter estimation for SMC shows robustness but with oscillatory behavior.

Introduction

The Permanent Magnet Synchronous Motor (PMSM) is the most popular electric machine because of its high-power density, high-efficiency, rapid dynamic response, small dimensions, and high torque characteristic. These advantages make it suitable for the use in electric vehicles, robotics, aerospace applications, submarines and servomotors, where high speed and torque are required^{1, 2}. In comparison with induction motor, the PMSM provides a higher efficiency and better power factor since there are no rotor losses of copper in it. The global tendency towards electrification of the transport industry has caused the popularity of PMSM, and by 2040 54% of cars sold are expected to be electric. But the dynamics of PMSM are characterized by nonlinearity and strong parameter dependence which makes control design difficult³. The main difficulties of the PMSM control are parameter variability due to overheating, magnetic saturation, and aging, as well as the requirement of sensorless control for the estimation of rotor position and speed⁴.

Among many other control techniques Model Predictive Control (MPC) can be considered one of the best possible options for PMSM drive control systems due to fast transient response, multivariable optimization capability and taking into account the system constraints⁵. While traditional PI controllers and FOC consider the present moment for control actions, MPC makes predictions about the future state of the motor and finds out the most optimal control actions according to the minimized predefined cost function⁶. In doing so, the MPC controller achieves outstanding dynamic performance and low current ripple at different operating conditions. Due to recent improvements in discrete-time models of PMSMs, the numerical stability and

prediction accuracy of MPC have become higher compared to those using traditional forward Euler discretization method. A systematic review on MPC of PMSMs for electrified transportation applications reveals that integration of artificial intelligence with predictive control constitutes an important future development direction⁷. However, both FCS-MPC and CCS-MPC have several technical challenges, namely, variable switching frequency, high current ripples, difficulties in the weighting factors tuning, and significant computational requirements.

However, despite their strengths the performance of MPC becomes extremely sensitive to the accuracy of the model of the motor. In practice PMSM is exposed to continuous parameter changes caused by such factors as temperature changes, magnetic saturation, aging, manufacturing tolerances and load variations. The key parameters include stator resistance R_s , d-axis inductance L_d , q-axis inductance L_q , permanent magnet flux linkage Ψ_f , and rotor inertia J . Any changes in these parameters decrease the prediction accuracy, increase current distortion, decrease torque accuracy, and even result in the loss of stability of the whole control process^{8,9}. The rise in temperature increases stator resistance but decreases both rotor flux linkage and d-q axis inductances, which results in torque ripples¹⁰. Thus, nowadays, accurate estimation of the parameters of PMSM online is becoming one of the crucial parts of control systems¹¹. Recent systematic reviews have shown that online parameter identification still presents a serious technical challenge especially because of such problems as rank deficiency in the model equations, cross-saturation effects, nonlinearity of the inverter, and parameter identification at the extreme conditions¹². The problem of the rank deficiency where the number of parameters to be estimated exceeds the number of independent machine equations.

Several different approaches for parameter estimation have been reported in literature. Conventional approaches like RLS (Recursive Least Squares), EKF (Extended Kalman Filter), UKF (Unscented Kalman Filter), and adaptive observers exhibit good performance in estimating parameters under certain operating conditions^{12,13}. These approaches normally have limitations like linearizing the models, require tuning of their parameters and have large computation cost or poor robustness due to non-linearity and changes in operating conditions¹⁴. Recent developments have tried to overcome these problems using better formulations such as fixed-time observer based sliding mode control which ensures fixed time convergence¹⁵. The literature on parameter estimation problem has also advanced to tackle the problem of sensor-less control, where the additional error in rotor position estimation results in a difficult estimation of all the parameters¹⁶. Techniques like partially fixed parameters, extra measurements, harmonic extraction, signal injection and variable speed excitation have been suggested to solve the problem of rank deficiency, each with its own pros and cons¹². Recently, a review has been conducted on parameter estimation problem of PMSM and techniques like RLS, Kalman filters, Model reference adaptive system, Adaline Neural Networks, Gradient based approach, Particle swarm optimization and Genetic Algorithms have been included¹¹.

In the past few years the application of Artificial Intelligence (AI) techniques for the purpose of parameter identification has gained a lot of popularity because of their ability to approximate a nonlinear map with just input-output data¹⁷. Within various AI techniques RNNs are extremely suited for dynamic systems since they have a memory capacity. Some researchers have found that RNN estimators have higher estimation accuracy and robustness than conventional neural networks in electrical drive systems¹⁸. In 2021, an RNN based method was proposed by Wu et al. for the purpose of online parameter identification and efficiency optimization of interior PMSM drive systems, which proved its high speed and accuracy¹⁹. In the same year, a survey paper on online parameter estimation methods of PMSMs by Zhu, Liang, and Liu presented the current modeling approaches and discussed the key challenges of these techniques, namely, rank deficiency issue and nonlinearity of the inverter²⁰. Rotor position estimation techniques based on RNN have been recently proposed to build up a nonlinear relationship between stator voltages and rotor position, offering frequency-dependent filtering properties to get rid of harmonic disturbances²⁰. Moreover, a RNN based robust adaptive model predictive speed control strategy has been introduced to estimate the parameter mismatches online and solve an optimization problem to determine control law²¹. However, the performance of RNN mainly relies on the diversity of training data.

Recently, Reinforcement Learning (RL) became a useful technique in adaptive control and optimization. While compared to the methods of supervised learning, RL updates control policy based on continuous interaction with the environment and reward maximization²². Several DRL algorithms such as Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3), and Soft Actor-Critic (SAC) have proven to be efficient tools for the control of continuous processes including electric motor drives^{23,24}. In 2021, Song et al. examined speed control strategies in DRL-based control of PMSM servo systems, considering the speed control problem as the Markov decision process. Thus, RL-based speed control improved robustness to disturbances in comparison with conventional PI control²⁵. In 2021, a policy gradient-based DRL vector controller of PMSM for electric vehicles was designed in order to avoid non-linearity of the machine parameters and to eliminate the decoupling accuracy issues²⁶. Some studies proved that the usage of RL-based control allows achieving a significant decrease in tracking error under variable speed conditions which makes the system highly responsive²⁷. The TD3 algorithm was successfully used in the current loop control in FOC framework. The measured output signals were in accordance with the simulation results. A 2022 study compared the linear and nonlinear PMSM control applying computational intelligence and RL algorithms, and found out that the PMSM-control system with the RL-TD3 agents performed better than PI type classical controllers²⁸. An

RL-based online estimation of PMSM inductances is one of the novel approaches in parameter estimation²⁹. As mentioned before, the use of RL in MPC allows adjusting the model parameters in real time in order to make it more accurate in case of disturbance and uncertainties. According to some recent reviews, DRL-based control techniques for PMSM are facing some problems including algorithm designing and training, parameter sensitivity, and implementation issues³⁰.

Yet another a powerful nonlinear control approach is Sliding Mode Control (SMC) whose strong resistance to model uncertainties and disturbances is widely known³¹. However, conventional SMC can suffer from degraded performance and even chattering issues caused by significant differences between the real motor parameters and their nominal values. There have been improvements in SMC techniques made recently. Thus, designed adaptive sliding mode control with position tracking for PMSM drive systems using a nonlinear extended state observer to estimate unmeasured states and disturbances and to ensure finite-time stability³². In 2020, reaching law-based sliding mode discrete observer has been used to design MPC of PMSM drives and to increase parameter robustness and decrease parameter sensitiveness³³. With SMC used together with reinforcement learning, one can develop adaptive control system that increases parameter estimation without losing SMC robustness³⁴. Another way of SMC development is a use of neural networks that allow designing an adaptive sliding mode control with estimation error-driven learning laws for permanent magnet linear synchronous motors in which leakage terms driven by estimation errors estimate the unknown parameters online and the gain of discontinuous terms is changed to avoid chattering³⁵. In 2021, MPC of surface-mounted PMSMs with parameter robustness has been designed using parameter identification module based on deviation³⁶.

Although many works have been performed for intelligent estimation of PMSM parameters, yet most of the researchers have used only one approach for intelligent estimation or they have estimated only few parameters like stator resistance and/or permanent magnet flux linkage. There are very few works where more than one intelligent estimation approach is compared in order to estimate all the five major parameters R_s , L_d , L_q , Ψ_f , and J at various speeds and loads. There is no work which has compared the performance of RNN-MPC, RL-MPC, and RL-SMC approaches under the same working condition. Recent comprehensive reviews have pointed out that the control techniques of PMSM based on DRL suffer from various limitations in terms of algorithm design, training, parameter sensitivity, and real-time implementation issues^{12, 30}.

In order to overcome these gaps, this paper proposes a comparative study of three different intelligent parameter estimation techniques for PMSM: Recurrent Neural Network-based online parameter estimation for Model Predictive Control (RNN-MPC), Reinforcement Learning-based online parameter estimation for Model Predictive Control (RL-MPC), and Reinforcement Learning-based online parameter estimation Sliding Mode Control (RL-SMC). These techniques estimate the following motor parameters: stator resistance R_s , d-axis inductance L_d , q-axis inductance L_q , permanent magnet flux linkage Ψ_f , and rotor inertia J .

The main contributions of this paper are summarized as follows: The main contributions of this paper are summarized as follows: The main contributions of this paper are summarized as follows:

- An RNN-based online parameter estimation framework is developed and integrated with model predictive control (MPC) for the estimation key PMSM parameters, namely the stator resistance R_s , d-axis inductance L_d , q-axis inductance L_q , permanent-magnet flux linkage Ψ_f , and rotor moment of inertia J .
- Reinforcement learning (RL)-based parameter estimation framework is developed for MPC, enabling online adaptation of the PMSM prediction model to parameter variations under dynamic operating conditions.
- An RL-based online parameter estimation for sliding mode control (SMC) framework is developed by combining the adaptive learning capability of RL with the inherent robustness of SMC against parameter uncertainties and disturbances.
- A comparative assessment of the proposed RNN-MPC, RL-MPC, and RL-SMC frameworks is conducted under variable-speed operating conditions, providing a consistent basis for evaluating their parameter estimation and closed-loop control performance.
- Lyapunov-based stability analyses are established for the proposed adaptive control frameworks to characterize the influence of bounded parameter-estimation errors on closed-loop stability and tracking performance.

System Modeling and Preliminaries

Figure 1 shows the structure of the surface-mounted permanent magnet synchronous motor (SM-PMSM). It includes the three-phase winding on the stator, the rotor with permanent magnets, (d)-(q) coordinate frame, rotor electrical angle (θ_e), and the three-phase inverter feeding the motor. The (d)-axis aligns with the rotor's permanent magnets' magnetic field, and the (q)-axis is perpendicular to that. Such (dq)-frame of reference makes a good choice to describe the motor dynamics. Thus, the continuous time model is derived based on the dynamics of the PMSM and then transformed into the discrete-time state space model.

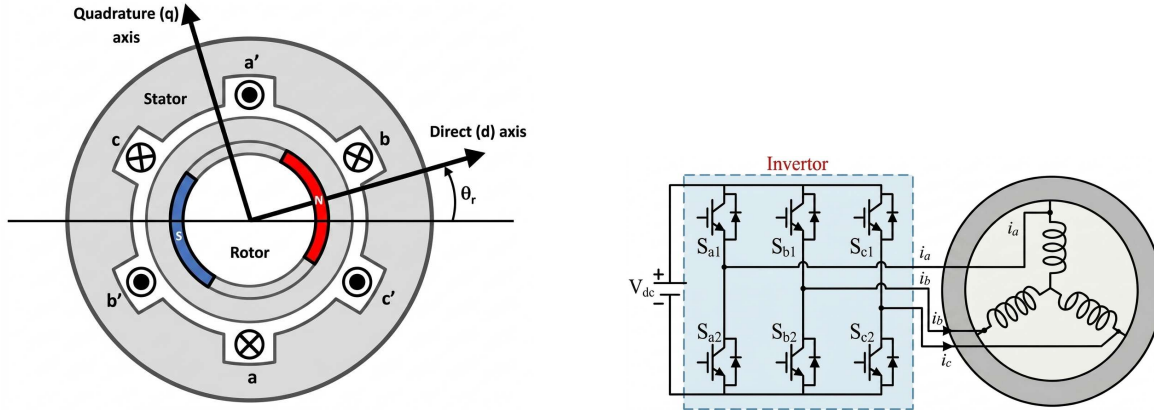


Figure 1. Surface-Mounted PMSM.

PMSM Electrical Model in the dq -Reference Frame

The dynamic behavior of a Surface-Mounted PMSM (SM-PMSM) in the synchronous rotating reference frame is governed by the following stator voltage equations, derived from Faraday's law:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q \quad (1)$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + \omega_e (L_d i_d + \psi_f) \quad (2)$$

where, v_d, v_q are the d- and q-axis stator voltages; i_d, i_q are the d- and q-axis stator currents; R_s is the stator resistance; L_d, L_q are the d- and q-axis inductances, For SM-PMSM, $L_d = L_q = L_s$; ψ_f is the permanent magnet flux linkage; $\omega_e = p\omega_m$ is the electrical angular speed, where ω_m is the mechanical rotor speed and p is the number of pole pairs.

The electromagnetic torque T_e is expressed as the co-energy derivative with respect to the mechanical angle, yielding:

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

For SM-PMSM (where $L_d = L_q$), the reluctance torque term $(L_d - L_q) i_d i_q$ vanishes, leading to the simple linear relationship:

$$T_e = K_t i_q = \frac{3}{2} p \psi_f i_q \quad (4)$$

where K_t is the torque constant.

Mechanical Model

The mechanical subsystem is described by Newton's second law for rotational motion, including the inertia, viscous friction, and load torque:

$$J \frac{d\omega_m}{dt} = T_e - T_L - B\omega_m \quad (5)$$

$$\frac{d\theta_m}{dt} = \omega_m \quad (6)$$

where, J is the rotor moment of inertia; B is the viscous friction coefficient; T_L is the external load torque; θ_m is the rotor mechanical angle.

The (dq)-axis current dynamics of the PMSM can be obtained from the stator voltage equations. The corresponding current equations can be represented as follows:

$$\frac{di_d}{dt} = -\frac{R_s}{L_d}i_d + \frac{L_q}{L_d}\omega_e i_q + \frac{1}{L_d}v_d \quad (7)$$

$$\frac{di_q}{dt} = -\frac{R_s}{L_q}i_q - \frac{L_d}{L_q}\omega_e i_d - \frac{\psi_f}{L_q}\omega_e + \frac{1}{L_q}v_q \quad (8)$$

$$\frac{d\omega_m}{dt} = \frac{3p\psi_f}{2J}i_q - \frac{B}{J}\omega_m - \frac{1}{J}T_L \quad (9)$$

This system can be compactly written in the standard matrix form as $\dot{x} = A(\theta)x + B(\theta)u + D$, where:

The parameter vector affecting the system dynamics is $\theta = [R_s, L_d, L_q, \psi_f, J]^T$.

Discrete-Time PMSM Model

Since the MPC operates in discrete time with a fixed sampling period T_s , the continuous-time PMSM model is discretized using the forward Euler method. For a sufficiently small sampling period, the discrete-time state equation is given by

$$x(k+1) = A_d(\theta)x(k) + B_d(\theta)u(k) + D_d(k), \quad (10)$$

where $x(k) = [i_d(k), i_q(k), \omega_m(k)]^T$ and $u(k) = [v_d(k), v_q(k)]^T$. Using

$$A_d(\theta) = I + A_c(\theta)T_s, \quad B_d(\theta) = B_c(\theta)T_s, \quad (11)$$

the discrete-time system matrix becomes

$$A_d(\theta) = \begin{bmatrix} 1 - \frac{R_s T_s}{L_d} & \frac{L_q}{L_d} p \omega_m T_s & 0 \\ -\frac{L_d}{L_q} p \omega_m T_s & 1 - \frac{R_s T_s}{L_q} & -\frac{p \psi_f T_s}{L_q} \\ 0 & \frac{3p \psi_f T_s}{2J} & 1 - \frac{B T_s}{J} \end{bmatrix}, \quad (12)$$

while the discrete-time input matrix is

$$B_d(\theta) = \begin{bmatrix} \frac{T_s}{L_d} & 0 \\ 0 & \frac{T_s}{L_q} \\ 0 & 0 \end{bmatrix}. \quad (13)$$

The load-torque contribution is discretized as

$$D_d(k) = \begin{bmatrix} 0 \\ 0 \\ -\frac{T_s}{J}T_L(k) \end{bmatrix}. \quad (14)$$

where A_c and B_c denote the continuous-time state and input matrices, respectively, while A_d and B_d denote the corresponding discrete-time state-transition and input matrices, respectively.

Therefore, the complete discrete-time PMSM model used for MPC prediction can be expressed as

$$\begin{bmatrix} i_d(k+1) \\ i_q(k+1) \\ \omega_m(k+1) \end{bmatrix} = A_d(\theta) \begin{bmatrix} i_d(k) \\ i_q(k) \\ \omega_m(k) \end{bmatrix} + B_d(\theta) \begin{bmatrix} v_d(k) \\ v_q(k) \end{bmatrix} + D_d(k). \quad (15)$$

This discrete-time model explicitly shows how the PMSM parameters R_s , L_d , L_q , ψ_f , and J affect the predicted motor states and therefore provides the mathematical basis for the proposed parameter estimation and MPC frameworks.

RNN based Online Parameter Estimation for MPC

The efficacy of the MPC hinges on the accuracy of $A_d(\hat{\theta})$ and $B_d(\hat{\theta})$, which depend directly on the estimated PMSM parameters. As illustrated in Figure 2, the proposed RNN-based MPC architecture integrates an LSTM parameter estimator with the MPC controller to continuously update the prediction model according to the operating condition. The measured PMSM variables, including the d - and q -axis currents (i_d and i_q), d - and q -axis voltages (v_d and v_q), and rotor speed ω_m , are provided as inputs to the LSTM parameter estimator. The LSTM processes the temporal relationships among these measured variables and generates the estimated PMSM parameters, which are then supplied to the MPC controller to update its prediction model. Using the updated model, the MPC solves the optimization problem based on the defined cost function and generates the control inputs for the PMSM. The resulting motor responses are fed back to the estimator and controller, forming a closed-loop parameter estimation and control structure. This integration enables the MPC prediction model to adapt to changes in the PMSM parameters during operation. The following subsection details the LSTM architecture, input data preparation, and training procedure used for online parameter estimation.

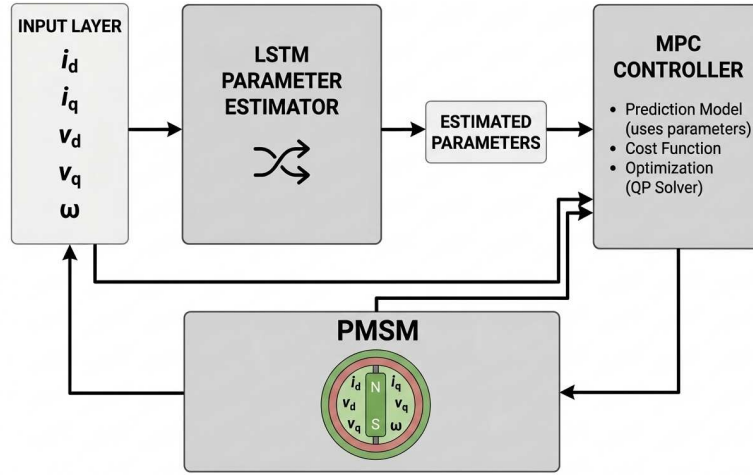


Figure 2. Architecture of RNN based MPC parameter estimation.

RNN Input and Output Formulation

At each discrete time step k , the RNN's input vector is a stacked vector of all measurable and controllable variables:

$$z(k) = [i_d(k), i_q(k), v_d(k), v_q(k), \omega_m(k)]^T \in \mathbb{R}^5 \quad (16)$$

To incorporate temporal context, the RNN takes a sequence of the past N samples as its input:

$$Z_k = [z(k-N+1), z(k-N+2), \dots, z(k)] \in \mathbb{R}^{5 \times N} \quad (17)$$

The RNN maps this input sequence to the estimated parameter vector:

$$\hat{\theta}(k) = \text{RNN}(Z_k) = \hat{R}_s(k) \quad (18)$$

Detailed LSTM Forward Propagation Equations

The internal operation of the LSTM cell consists of a sequence of gating and memory-update operations. Given the current input vector x_k and the previous hidden state h_{k-1} , the forget gate first determines which information from the previous cell state should be retained. It is computed as

$$f_k = \sigma(W_f[h_{k-1}, x_k] + b_f), \quad (19)$$

where f_k is the forget-gate output, W_f is the corresponding trainable weight matrix, b_f is the bias vector, and $[h_{k-1}, x_k]$ denotes the concatenation of the previous hidden state and current input.

Next, the input gate determines which new information should be added to the cell memory. The input-gate output is calculated as

$$i_k = \sigma(W_i[h_{k-1}, x_k] + b_i), \quad (20)$$

where i_k denotes the input-gate output, while W_i and b_i are the corresponding trainable weight matrix and bias vector, respectively.

The candidate cell state represents the new information that can be stored in the LSTM memory. It is obtained using the hyperbolic tangent activation function as

$$\tilde{c}_k = \tanh(W_c[h_{k-1}, x_k] + b_c), \quad (21)$$

where $\tilde{c}_k$ is the candidate cell state, and W_c and b_c are the associated trainable weight matrix and bias vector.

The previous cell state is then updated by combining the retained previous information with the newly selected candidate information. The updated cell state is given by

$$c_k = f_k \odot c_{k-1} + i_k \odot \tilde{c}_k, \quad (22)$$

where c_k and c_{k-1} denote the current and previous cell states, respectively, and $\odot$ represents element-wise multiplication.

After updating the cell state, the output gate determines which information from the current memory is passed to the hidden state. The output gate is calculated as

$$o_k = \sigma(W_o[h_{k-1}, x_k] + b_o), \quad (23)$$

where o_k denotes the output-gate vector, and W_o and b_o are the corresponding trainable weight matrix and bias vector.

Finally, the hidden state is generated by filtering the updated cell state through the output gate. It is expressed as

$$h_k = o_k \odot \tanh(c_k), \quad (24)$$

where h_k is the current hidden state and represents the temporal information extracted by the LSTM at sampling step k .

The sigmoid activation function used in the forget, input, and output gates is defined as

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \quad (25)$$

whereas $\tanh(\cdot)$ denotes the hyperbolic tangent activation function. After processing the sequential input data, the hidden state h_k is passed through a fully connected output layer to generate the estimated PMSM parameter vector:

$$\hat{\theta}(k) = W_\theta h_k + b_\theta, \quad (26)$$

where W_θ and b_θ are the trainable weight matrix and bias vector of the fully connected output layer, respectively, and $\hat{\theta}(k)$ denotes the estimated parameter.

RNN Training Objective and Normalization

Before training, all parameters in the dataset are normalized to ensure stable and efficient gradient descent. For any parameter $\Theta \in \theta$, the normalized value $\bar{\Theta}$ is:

$$\bar{\Theta} = \frac{\Theta - \mu_\Theta}{\sigma_\Theta} \quad (27)$$

where μ_Θ is the mean and σ_Θ is the standard deviation calculated over the entire training dataset.

The RNN is trained offline using a large, diverse dataset generated from a high-fidelity PMSM simulator covering various speeds, loads, and parameter drift scenarios. The training loss function is the Mean Squared Error (MSE):

$$\mathcal{L}_{RNN} = \frac{1}{N_{\text{batch}}} \sum_{k=1}^{N_{\text{batch}}} \|\bar{\Theta}(k) - \hat{\Theta}(k)\|_2^2 \quad (28)$$

where $\mathcal{L}_{RNN}$ denotes the RNN training loss, N_{batch} is the number of samples contained in each mini-batch, $\bar{\Theta}(k)$ represents the normalized target PMSM parameter vector at sample k , and $\hat{\Theta}(k)$ denotes the corresponding normalized parameter vector predicted by the RNN. The individual estimation accuracy of each PMSM parameter is evaluated using the Root Mean Square Error (RMSE), given by

$$RMSE_{\theta_j} = \sqrt{\frac{1}{N} \sum_{k=1}^N (\theta_j(k) - \hat{\theta}_j(k))^2} \quad (29)$$

MPC Formulation with RNN Estimates

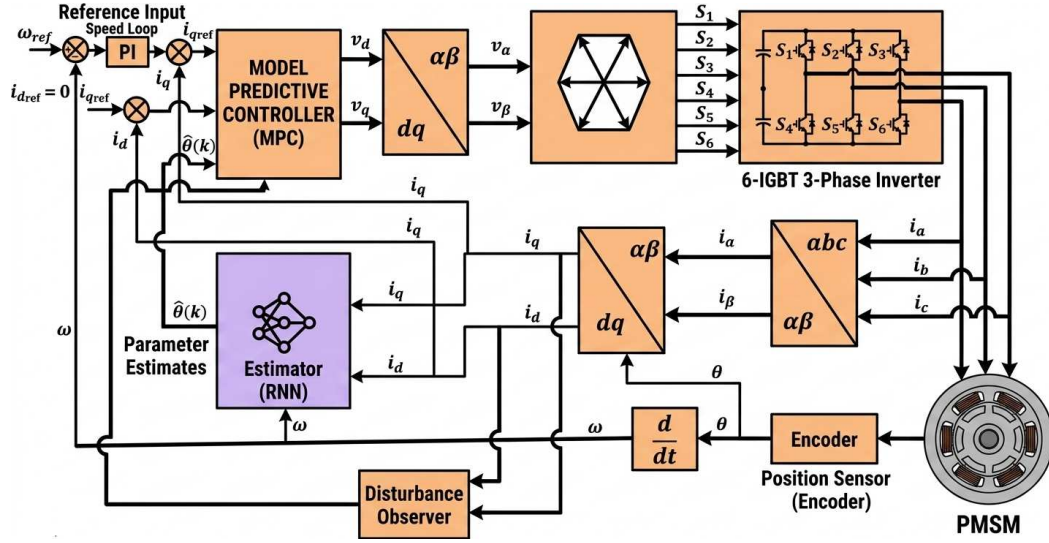


Figure 3. Block diagram of RNN based online parameter estimation for MPC .

In Figure 3 the block diagram of RNN based online parameter estimation for MPC is shown. At each control cycle k , the MPC receives the latest RNN estimates $\hat{\theta}(k)$. The discrete-time model for prediction is updated, resulting in a time-varying Linear Time-Invariant (LTI) model valid over the short prediction horizon N_p :

$$x_{k+j+1} = A_d(\hat{\theta}_k)x_{k+j} + B_d(\hat{\theta}_k)u_{k+j} + \hat{d}(k) \quad (30)$$

where $\hat{d}(k)$ is an estimate of the disturbance term, typically updated using a disturbance observer or by filtering the previous step's prediction error.

The MPC cost function J_{MPC} to be minimized is:

$$J_{MPC} = \sum_{j=1}^{N_p} \|x_{k+j} - x_{ref}\|_Q^2 + \sum_{j=0}^{N_c-1} \|u_{k+j}\|_R^2 + \sum_{j=1}^{N_c-1} \|\Delta u_{k+j}\|_S^2 \quad (31)$$

Where, N_p is the prediction horizon; N_c is the control horizon; $Q \in \mathbb{R}^{3 \times 3}$, $R \in \mathbb{R}^{2 \times 2}$, and $S \in \mathbb{R}^{2 \times 2}$ are positive definite weighting matrices for states, inputs, and input rates, respectively; x_{ref} is the reference state trajectory ($i_{d,ref} = 0$, $\omega_{m,ref}$ from the speed loop); $\Delta u_{k+j} = u_{k+j} - u_{k+j-1}$ is the input increment, used to penalize high-frequency control activity.

Subject to physical constraints:

$$u_{\min} \leq u_{k+j} \leq u_{\max} \quad \forall j \in [0, N_c - 1] \quad (32)$$

$$\Delta u_{\min} \leq \Delta u_{k+j} \leq \Delta u_{\max} \quad \forall j \in [1, N_c - 1] \quad (33)$$

The optimal control sequence is found by solving the constrained quadratic program (QP):

$$U_k^* = \arg \min_U J_{MPC} \quad (34)$$

RL based Online Parameter Estimation for MPC

In addition to the RNN-based parameter estimation technique, there is another mechanism used for parameter estimation based on the RL technique. It offers an adaptive mechanism capable of learning from changes in operating conditions by interacting with the motor environment. As shown in Figure 4, the RL agent takes the observed states of the PMSM, namely, the d -axis and q -axis current (i_d, i_q), the d -axis and q -axis voltages (v_d, v_q), the rotor speed (ω), and electromagnetic torque (T_e). Using these state variables, the RL agent computes the parameter estimates ($\hat{R}_s, \hat{\psi}_f, \hat{L}_d, \hat{L}_q, \hat{J}$), which are then provided to the MPC in order to adjust the prediction model and find the optimal control input u_k^* . The output of the PMSM is then given back to the RL agent as a feedback via state observations and reward signal r_k . The aforementioned interaction enables the RL agent to continually optimize the estimation policy using the reward function, thus, creating an adaptive parameter estimation mechanism for the MPC controller.

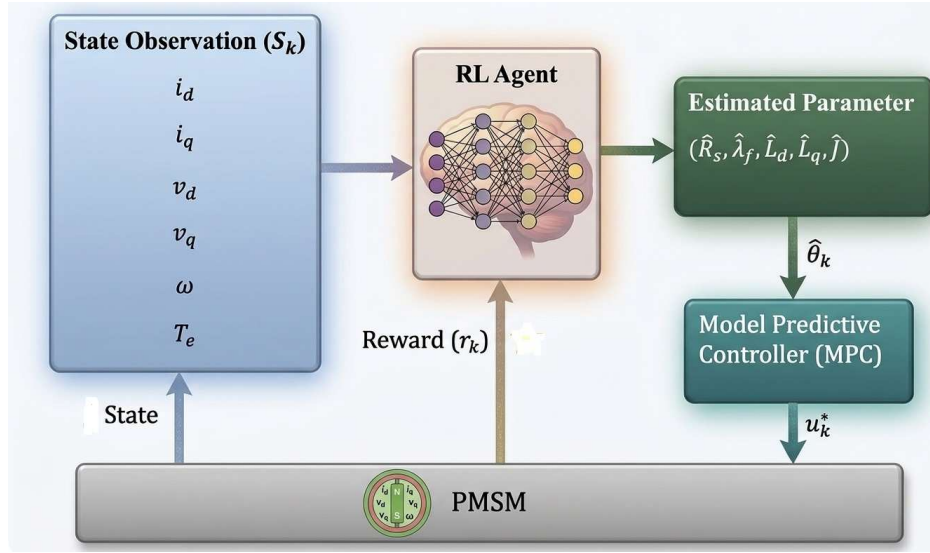


Figure 4. Architecture of RL based MPC parameter estimation.

RL Formulation for Parameter Estimation

The RL-based parameter estimation problem is formulated as a Markov Decision Process (MDP). The agent directly estimates the motor parameters by interacting with the system.

State Space

The RL state is designed to capture the relevant system information for parameter estimation:

$$s_k = [e_\omega, i_d, i_q, \omega_m, T_e, T_L]^T \quad (35)$$

where, $e_\omega = \omega_{ref} - \omega_m$ is the speed tracking error; i_d, i_q are the d- and q-axis currents; ω_m is the mechanical speed; T_e is the electromagnetic torque; T_L is the load torque.

Action Space

The action represents the parameter estimates. The agent can estimate single parameters. In this work, we consider two configurations:

$$a_k = \hat{R}_s \quad (36)$$

The reduced set estimates the one parameters most susceptible to temperature variation, which is computationally efficient while still capturing the dominant parameter uncertainties.

Transition Dynamics

The environment evolves according to the PMSM dynamics described in Section II: The actual d- and q-axis stator voltages are obtained from the PMSM electrical model as

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q, \quad (37)$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + \omega_e (L_d i_d + \psi_f). \quad (38)$$

Since the RL agent estimates one parameter the stator resistance R_s , the estimated d- and q-axis voltages are calculated by replacing the actual resistance with its estimated value $\hat{R}_s$:

$$\hat{v}_d = \hat{R}_s i_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q, \quad (39)$$

$$\hat{v}_q = \hat{R}_s i_q + L_q \frac{di_q}{dt} + \omega_e (L_d i_d + \psi_f). \quad (40)$$

$$s_{k+1} = F(s_k, a_k) \quad (41)$$

where F represents the PMSM state transition function incorporating the estimated parameters.

RL Objective and Policy

The agent aims to find an optimal policy $\pi^*(a|s)$ that maximizes the expected discounted cumulative reward:

$$\pi^*(a|s) = \arg \max_{\pi} \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k r_k \right] \quad (42)$$

where $\gamma \in [0, 1]$ is the discount factor that determines the weight given to future rewards.

Reward Function Design

The reward function is designed to guide the agent towards accurate parameter estimation while ensuring smoothness and stability. The immediate reward is defined as:

$$r_k = -[Q_1(v_d - \hat{v}_d)^2 + Q_2(v_q - \hat{v}_q)^2 + Q_3(\theta - \hat{\theta})^2 + R(\hat{\theta}_k - \hat{\theta}_{k-1})^2] \quad (43)$$

where, $(v_d - \hat{v}_d)^2$ and $(v_q - \hat{v}_q)^2$ are the voltage errors, penalizing inaccurate parameter estimates; $(\theta - \hat{\theta})^2$ penalizes the direct parameter estimation error; $(\hat{\theta}_k - \hat{\theta}_{k-1})^2$ is a smoothness term that penalizes abrupt changes in the estimates, ensuring the RL agent produces physically plausible, slowly-varying parameters; Q_1, Q_2, Q_3, R are positive weighting factors that balance the different objectives.

Twin Delayed Deep Deterministic Policy Gradient (TD3) for Parameter Estimation

We employ the TD3 algorithm to solve the parameter estimation MDP. TD3 is an off-policy, model-free actor-critic method designed for continuous action spaces.

Network Architecture

The TD3 architecture consists of, Actor Network: $a = \pi_{\phi}(s)$ maps the state to an action (parameter estimates). It has parameters ϕ ; Critic Networks: $Q_{\theta_1}(s, a)$ and $Q_{\theta_2}(s, a)$ are two networks that estimate the expected return. They have parameters θ_1 and θ_2 ; Target Networks: $\pi_{\phi'}, Q_{\theta'_1}, Q_{\theta'_2}$ are delayed copies used for stable learning.

Target Calculation

The target value y is computed using the target actor and critic networks:

$$y = r + \gamma \min_{l=1,2} Q_{\theta'_l}(s', \pi_{\phi'}(s')) + \varepsilon \quad (44)$$

where $\varepsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c)$ is a small noise added to the target policy for regularization (smoothing). This target smoothing prevents the network from overfitting to narrow peaks in the Q -function.

Critic Update

Both critic networks are updated by minimizing the Mean Squared Bellman Error (MSBE):

$$L(\theta_i) = \frac{1}{N_{\text{batch}}} \sum (y - Q_{\theta_i}(s, a))^2, \quad i = 1, 2 \quad (45)$$

Actor Update

The actor is updated every d (e.g., $d = 2$) iterations using the deterministic policy gradient:

$$\nabla_{\phi} J \approx \frac{1}{N_{\text{batch}}} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_{\phi}(s)} \nabla_{\phi} \pi_{\phi}(s) \quad (46)$$

Target Update

The target networks are updated via Polyak averaging:

$$\theta'_i \leftarrow \tau \theta_i + (1 - \tau) \theta'_i \quad (47)$$

$$\phi' \leftarrow \tau \phi + (1 - \tau) \phi' \quad (48)$$

with a slow update rate $\tau \ll 1$.

Parameter Constraints for Physical Feasibility

To ensure the RL agent outputs physically meaningful estimates, we implement hard constraints on the action space. The agent's output is clamped as follows:

$$R_s^{\min} \leq \hat{R}_s \leq R_s^{\max} \quad (49)$$

$$L_d^{\min} \leq \hat{L}_d \leq L_d^{\max} \quad (50)$$

$$L_q^{\min} \leq \hat{L}_q \leq L_q^{\max} \quad (51)$$

$$\psi_f^{\min} \leq \hat{\psi}_f \leq \psi_f^{\max} \quad (52)$$

$$J^{\min} \leq \hat{J} \leq J^{\max} \quad (53)$$

These constraints prevent the agent from exploring physically impossible parameter regions, which is crucial for maintaining system stability during learning and ensuring practical applicability.

Integration with MPC

The RL-estimated parameters $\hat{\theta}_{RL}(k)$ are used to update the MPC's internal model. As shown in Figure 5 the block diagram represent RL online parameter estimator for MPC.

$$x_{k+j+1} = A_d(\hat{\theta}_{RL,k})x_{k+j} + B_d(\hat{\theta}_{RL,k})u_{k+j} + \hat{d}(k) \quad (54)$$

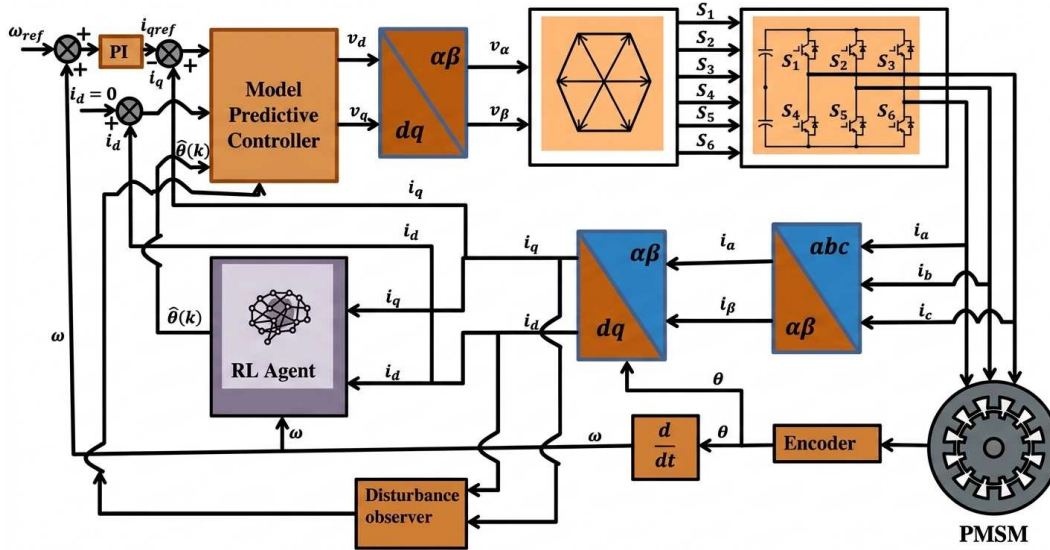


Figure 5. Block diagram of RL based online parameter estimation for MPC .

The MPC then solves the same optimization problem as in (31)-(34) using the RL-estimated parameters. This provides a purely data-driven alternative to the RNN-based estimation, with the advantage of being able to adapt online through trial-and-error learning.

RL based Online Parameter Estimation for SMC

In this section, the design of RL with the SMC framework for the online estimation of parameters is demonstrated. As shown in Figure 6, the RL agent obtains the measurement values of PMSM states, which include d -axis and q -axis currents (i_d and i_q), d -axis and q -axis voltages (v_d and v_q), and rotor speed ω . In order to estimate the motor parameter using the measured states, the RL agent computes the estimated motor parameter. The computed value is then used by the SMC as the parameter value for control. On the other hand, the SMC computes the input for the PMSM control, where the feedback of the motor state and the reward signal is used by the RL agent.

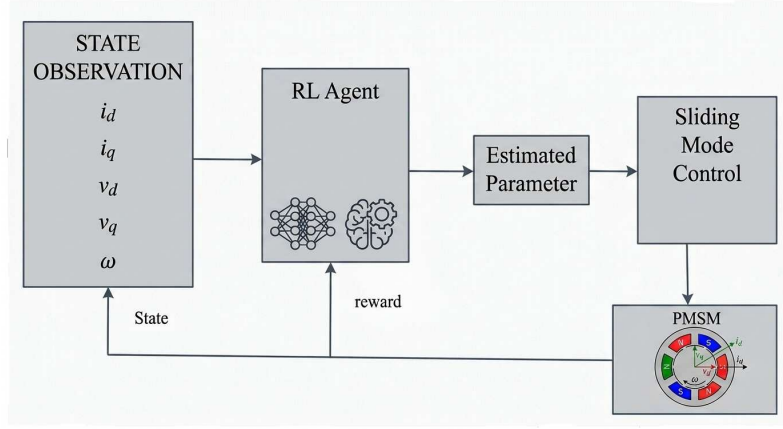


Figure 6. Architecture of RL based SMC parameter estimation.

Sliding Surface Design

The primary control objective is speed tracking. The speed tracking error is defined as:

$$e_\omega = \omega_{ref} - \omega_m \quad (55)$$

To ensure convergence with prescribed dynamics, we define a sliding surface s . A standard first-order surface is:

$$s = \dot{e}_\omega + \lambda e_\omega \quad (56)$$

where $\lambda > 0$ is a design parameter that determines the convergence rate on the sliding surface.

To enhance robustness against constant disturbances (like load torque), which cause steady-state errors in the sliding phase, we modify the surface to include an integral action:

$$s = \dot{e}_\omega + \lambda_1 e_\omega + \lambda_2 \int_0^t e_\omega(\tau) d\tau \quad (57)$$

where $\lambda_1 > 0, \lambda_2 > 0$. The integral term ensures zero steady-state error even under persistent disturbances.

SMC Control Law Derivation

The SMC control law is structured as the sum of an equivalent control and a switching control:

$$u = u_{eq} + u_{sw} \quad (58)$$

Equivalent Control (u_{eq})

The equivalent control is the ideal control that maintains $\dot{s} = 0$ in the absence of uncertainties and disturbances. To derive it, we substitute (5) and (8) into $\dot{s} = 0$. From (8), the derivative of i_q depends on v_q . The mechanical dynamics give $\dot{\omega}_m$, and hence $\dot{e}_\omega$.

The time derivative of the sliding surface from (56) is:

$$\dot{s} = \ddot{e}_\omega + \lambda \dot{e}_\omega \quad (59)$$

Substituting the mechanical and electrical dynamics yields the equivalent control:

$$u_{eq} = \frac{L_q}{K_t} \left(\ddot{\omega}_{ref} + \lambda \dot{e}_\omega + \frac{B}{J} \omega_m + \frac{T_L}{J} \right) + R_s i_q + \omega_e L_d i_d + \omega_e \psi_f \quad (60)$$

For the integral sliding surface in (57), the equivalent control becomes:

$$u_{eq} = \frac{L_q}{K_t} \left(\ddot{\omega}_{ref} + \lambda_1 \dot{e}_\omega + \lambda_2 e_\omega + \frac{B}{J} \omega_m + \frac{T_L}{J} \right) + R_s i_q + \omega_e L_d i_d + \omega_e \psi_f \quad (61)$$

In practice, (60) and (61) are computed using the RL-estimated parameters $\hat{R}_s$ to make the equivalent control parameter changes.

Switching Control (u_{sw})

To handle uncertainties (bounds on $\Delta R_s, \Delta \psi_f$), a discontinuous or high-gain term is added. To mitigate chattering, a saturation function is used instead of the discontinuous sign function:

$$u_{sw} = -K_{smc} \cdot \text{sat}\left(\frac{s}{\phi}\right) \quad (62)$$

where: $\text{sat}(y) = y$ if $|y| \leq 1$, and $\text{sat}(y) = \text{sign}(y)$ otherwise; $\phi > 0$ is the boundary layer thickness; K_{smc} is the switching gain, which is adapted online by the RL agent.

The saturation function replaces the ideal sign function to eliminate high-frequency chattering:

$$\text{sign}(s) \rightarrow \text{sat}\left(\frac{s}{\phi}\right) \quad (63)$$

Inside the boundary layer ($|s| \leq \phi$), the control is continuous, effectively smoothing the switching action.

RL Formulation for SMC Parameter Estimation

The RL-based SMC parameter estimation is formulated as a Markov Decision Process (MDP) where the agent learns to estimate the optimal SMC parameters.

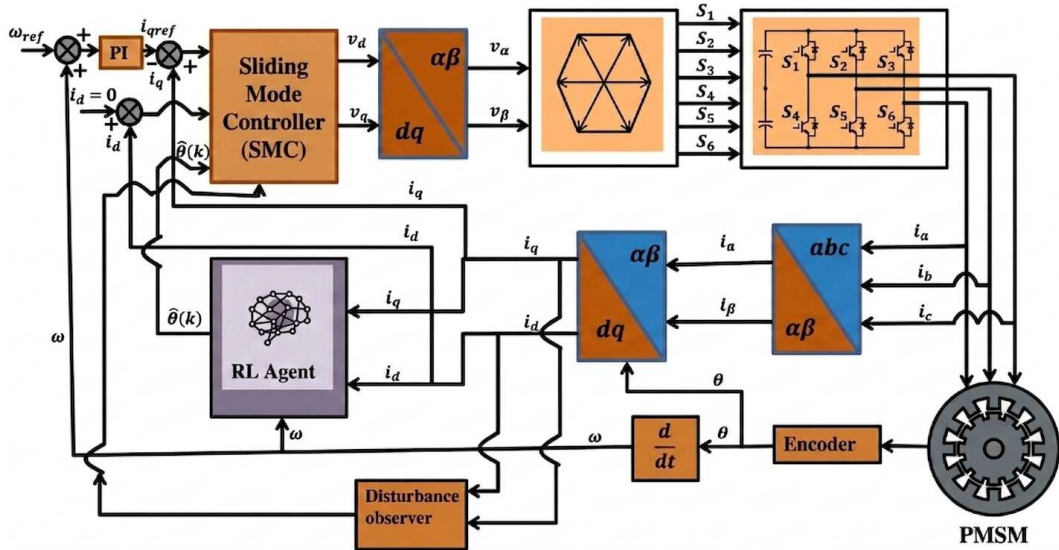


Figure 7. Block diagram of RL based SMC parameter estimation.

State Space for SMC

The state for the RL agent in the SMC parameter estimation is defined as:

$$s_k = [e_\omega(k), \dot{e}_\omega(k), s(k), i_d(k), i_q(k), \omega_m(k)]^T \quad (64)$$

This state captures the tracking error, its derivative, the sliding surface value, and the current states, providing full information for parameter adaptation.

Action Space for SMC

The action vector represents the SMC parameters to be estimated:

$$a_k = \hat{R}_s(k) \quad (65)$$

where, $\hat{R}_s(k)$ is the estimated stator resistance.

Lyapunov Stability Analysis

To establish the stability of the proposed PMSM control and parameter estimation frameworks, a Lyapunov-based analysis is conducted for the RL parameter estimator for SMC, RL parameter estimator for MPC, and RNN parameter estimator for MPC approaches. Since the parameter estimators are data-driven methods, the analysis does not assume asymptotic convergence of the learning algorithm itself. Instead, the closed-loop stability is established under the practical assumption that the parameter-estimation errors remain bounded. This condition is consistent with the objective of the proposed online estimation schemes, where the estimated parameters are continuously updated according to the measured PMSM states.

The parameter-estimation error is defined as

$$\tilde{\theta} = \theta - \hat{\theta}, \quad (66)$$

where θ denotes the actual PMSM parameter vector and $\hat{\theta}$ represents its estimated value. For the proposed RNN parameter estimator for MPC approach,

$$\theta = [R_s \quad L_d \quad L_q \quad \psi_f \quad J]^T, \quad (67)$$

while

$$\hat{\theta} = [\hat{R}_s \quad \hat{L}_d \quad \hat{L}_q \quad \hat{\psi}_f \quad \hat{J}]^T. \quad (68)$$

For the RL-based estimators considered in this work, the same formulation can be applied to the estimated parameter. In particular, when only the stator resistance is estimated,

$$\tilde{R}_s = R_s - \hat{R}_s. \quad (69)$$

The analysis assumes that the estimation error is bounded, i.e.,

$$\|\tilde{\theta}\| \leq \bar{\theta}, \quad (70)$$

where $\bar{\theta} > 0$ is a finite upper bound.

Lyapunov Analysis of RL parameter estimator for SMC

For the RL parameter estimator of SMC structure, the objective is to ensure that the PMSM speed tracking error converges toward zero despite bounded parameter-estimation errors and external disturbances. The mechanical dynamics of the PMSM can be expressed as

$$J\dot{\omega}_m = T_e - T_L - B\omega_m. \quad (71)$$

Let the speed tracking error be defined as

$$e_\omega = \omega_m - \omega_m^*, \quad (72)$$

where ω_m^* is the reference mechanical speed. A sliding surface is selected as

$$s = e_\omega + \lambda \int_0^t e_\omega(\tau) d\tau, \quad (73)$$

where $\lambda > 0$ is the sliding-surface coefficient. Differentiating (56) gives

$$\dot{s} = \dot{e}_\omega + \lambda e_\omega. \quad (74)$$

A Lyapunov candidate function is selected as

$$V_{SMC} = \frac{1}{2}s^2. \quad (75)$$

Since $V_{SMC} > 0$ for $s \neq 0$ and $V_{SMC} = 0$ when $s = 0$, it is positive definite. Its time derivative is

$$\dot{V}_{SMC} = s\dot{s}. \quad (76)$$

The SMC control law is designed such that

$$\dot{s} = -k_s \operatorname{sgn}(s) + d_s, \quad (77)$$

where $k_s > 0$ is the switching gain and d_s represents the lumped effect of parameter-estimation errors, load disturbances, and model uncertainties. Assuming that the uncertainty is bounded according to

$$|d_s| \leq \bar{d}_s, \quad (78)$$

and selecting

$$k_s > \bar{d}_s, \quad (79)$$

the Lyapunov derivative satisfies

$$\dot{V}_{SMC} = s [-k_s \operatorname{sgn}(s) + d_s] \quad (80)$$

$$\leq -k_s |s| + |s| \bar{d}_s \quad (81)$$

$$= -(k_s - \bar{d}_s) |s| < 0, \quad s \neq 0. \quad (82)$$

Therefore, the sliding variable converges toward zero, and the speed tracking error is ultimately bounded and converges to the desired trajectory. The bounded parameter-estimation error generated by the RL agent is incorporated into $\bar{d}_s$ and is compensated by the robust switching action of the SMC.

Lyapunov Analysis of RL parameter estimator for MPC

For the RL parameter estimator of MPC framework, the estimated PMSM parameters are used to construct the discrete-time prediction model. The actual and estimated models can be expressed as

$$x_{k+1} = A_d(\theta)x_k + B_d(\theta)u_k + D_k, \quad (83)$$

and

$$\hat{x}_{k+1} = A_d(\hat{\theta})x_k + B_d(\hat{\theta})u_k + D_k. \quad (84)$$

The prediction-model mismatch resulting from parameter estimation is therefore represented by

$$\Delta_k = [A_d(\theta) - A_d(\hat{\theta})]x_k + [B_d(\theta) - B_d(\hat{\theta})]u_k. \quad (85)$$

Because the PMSM model matrices depend continuously on the motor parameters, bounded parameter-estimation error implies a bounded model mismatch. Thus,

$$\|\Delta_k\| \leq \bar{\Delta}, \quad (86)$$

where $\bar{\Delta}$ is a finite upper bound on the prediction-model uncertainty.

Let the tracking error be

$$e_k = x_k - x_k^*, \quad (87)$$

where x_k^* is the desired state. Consider the quadratic Lyapunov function

$$V_{MPC}(k) = e_k^T P e_k, \quad (88)$$

where $P = P^T > 0$ is a positive-definite matrix.

For the nominal closed-loop system, the MPC feedback law is assumed to produce a stable error dynamics of the form

$$e_{k+1} = A_{cl}e_k + \Delta_k, \quad (89)$$

where A_{cl} denotes the nominal closed-loop system matrix and Δ_k represents the bounded prediction-model mismatch caused by parameter-estimation errors.

The difference in the Lyapunov function is

$$\Delta V_{MPC} = V_{MPC}(k+1) - V_{MPC}(k). \quad (90)$$

Substituting (89) gives

$$\begin{aligned}\Delta V_{MPC} &= e_k^T (A_{cl}^T P A_{cl} - P) e_k \\ &\quad + 2e_k^T A_{cl}^T P \Delta_k + \Delta_k^T P \Delta_k.\end{aligned}\quad (91)$$

If the MPC closed-loop matrix satisfies

$$A_{cl}^T P A_{cl} - P = -Q, \quad Q = Q^T > 0, \quad (92)$$

then

$$\Delta V_{MPC} = -e_k^T Q e_k + 2e_k^T A_{cl}^T P \Delta_k + \Delta_k^T P \Delta_k. \quad (93)$$

Using the boundedness of Δ_k , the effect of parameter-estimation error remains bounded. Consequently, for sufficiently small estimation error, the negative-definite nominal term dominates the uncertainty terms, resulting in a stable or practically stable closed-loop response. Therefore, the RL-based parameter estimator can continuously update the MPC prediction model while preserving closed-loop stability provided that the resulting model uncertainty remains within the admissible stability region of the MPC controller.

Lyapunov Analysis of RNN parameter estimator for MPC

For the RNN parameter estimator of MPC framework, the LSTM network provides the estimated parameter vector $\hat{\theta}_k$ from the measured PMSM operating variables. The estimation error is defined as

$$\tilde{\theta}_k = \theta - \hat{\theta}_k. \quad (94)$$

The corresponding discrete-time PMSM model is

$$x_{k+1} = A_d(\hat{\theta}_k)x_k + B_d(\hat{\theta}_k)u_k + D_k + \Delta_k, \quad (95)$$

where Δ_k represents the modeling error caused by imperfect parameter estimation and unmodeled dynamics. Assuming that the trained RNN provides a bounded parameter-estimation error,

$$\|\tilde{\theta}_k\| \leq \bar{\theta}_{RNN}, \quad (96)$$

the resulting model uncertainty is also bounded:

$$\|\Delta_k\| \leq \bar{\Delta}_{RNN}. \quad (97)$$

For the closed-loop tracking error

$$e_k = x_k - x_k^*, \quad (98)$$

consider the Lyapunov candidate

$$V_{RNN}(k) = e_k^T P e_k, \quad P = P^T > 0. \quad (99)$$

Under the nominal MPC controller, the tracking error can be represented as

$$e_{k+1} = A_{cl} e_k + \Delta_k. \quad (100)$$

The corresponding Lyapunov difference is

$$\Delta V_{RNN} = -e_k^T Q e_k + 2e_k^T A_{cl}^T P \Delta_k + \Delta_k^T P \Delta_k, \quad (101)$$

where $Q = Q^T > 0$ satisfies

$$A_{cl}^T P A_{cl} - P = -Q. \quad (102)$$

Because the RNN estimation error is bounded, the uncertainty term Δ_k is bounded. Therefore, the negative-definite term $-e_k^T Q e_k$ dominates the bounded uncertainty terms when the parameter estimation error is sufficiently small. Consequently, the RNN-based MPC closed-loop system is stable in the nominal case and practically stable under bounded parameter-estimation errors. Improved RNN estimation accuracy reduces $\bar{\Delta}_{RNN}$ and consequently reduces the deviation of the actual closed-loop behavior from the nominal MPC trajectory.

Results and Discussion

This section presents comprehensive simulation results validating the proposed hierarchical adaptive control framework for PMSM drives. The experiments were conducted using MATLAB/Simulink with the PMSM parameters listed in Table 1. The parameter estimation frameworks RNN parameter estimator for MPC, RL parameter estimator for MPC and RL parameter estimator for SMC. The performance is evaluated in terms of speed tracking accuracy, torque response, online parameter estimation capabilities, RNN training performance, and RL agent convergence under speed variation .

Table 1. PMSM System Parameters

Parameter	Symbol	Value
Rated Power	P_{rated}	7.797 kW
DC Voltage	V_{dc}	300 V
Rated Current	I_{rated}	28.88 A
Rated Speed	n_{rated}	2200 rpm
Rated Torque	T_{rated}	33.9 N.m
Pole Pairs	p	4
Stator Resistance	R_s	0.129 Ω
d-axis Inductance	L_d	1.53 mH
q-axis Inductance	L_q	1.53 mH
PM Flux Linkage	ψ_f	0.1821 Wb
Moment of Inertia	J	0.003334 kg.m ²

A. RNN Training and Regression Performance

Figure 8 presents the regression analysis results for the RNN-based parameter estimator. The regression plot shows the relationship between the target (actual) parameter values and the RNN's predicted outputs.

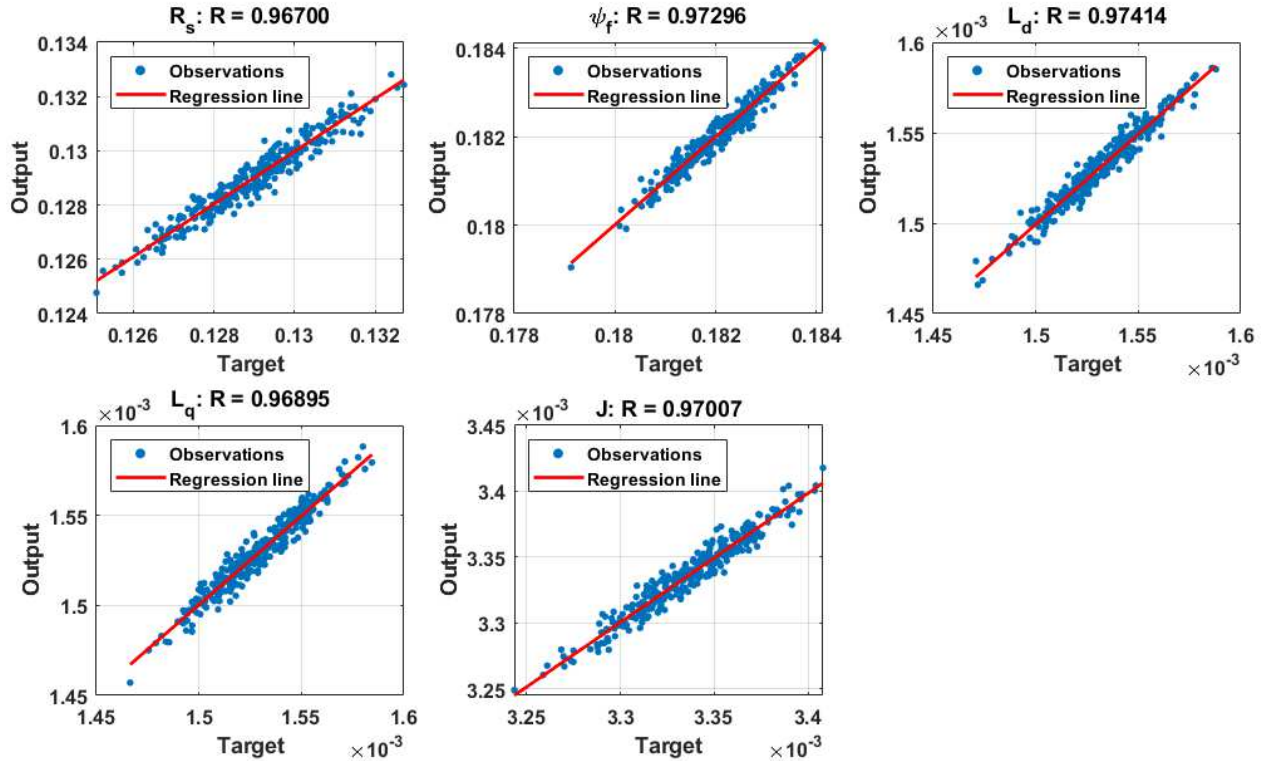


Figure 8. Regression performance of RNN-based parameter estimation.

The RNN regression results reveal high levels of prediction accuracy, generalization ability, and parameter estimation

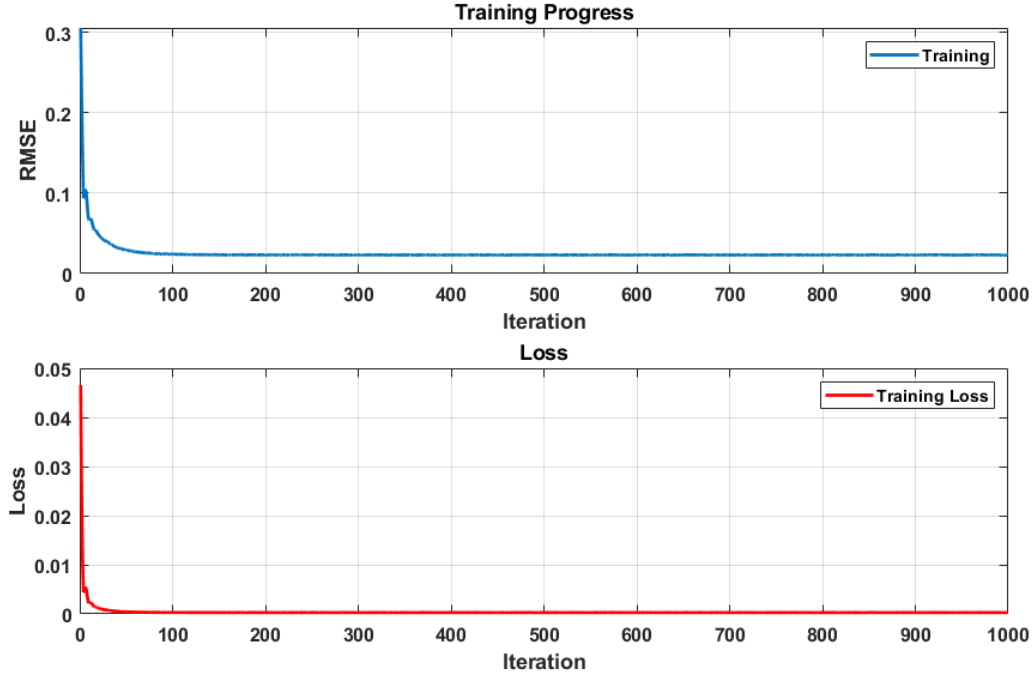


Figure 9. Evolution training loss and RMSE during RNN training.

reliability within the PMSM drive operating range. High correlations between the predicted and target values are visualized by regression plots where data points cluster around the ideal regression line ($y = x$) along with coefficients of determination (R^2) being over 0.95 for all estimated parameters, hence proving efficiency of the proposed RNN model. Moreover, the network has good generalization properties over the whole operating range which implies that the network has successfully learned the relationships between the inputs and parameters rather than overfitting to particular operating regimes. Out of the estimated parameters, R_s and ψ_f are the closest to the ideal regression line showing the highest level of accuracy and learning of the parameter changes, while L_d and L_q have somewhat bigger dispersion caused by the nonlinearity arising from magnetic saturation; however, correlation is still very high in these cases. Estimation of J is almost unaffected by any changes, and its predicted values cluster around the target one because of rather constant physical nature of the moment of inertia. Besides, the prediction errors are almost equally distributed over the zero which proves the absence of systematic bias in RNN parameter estimation and hence shows its high modeling ability in capturing motor dynamics. These findings also prove the effectiveness of the chosen RNN architecture, Adam optimizer and regularization technique in ensuring reliable parameter estimation. Overall, high regression results prove the RNN efficiency as an online parameter estimator providing reliable estimates of PMSM parameters.

B. RNN Training Loss and RMSE Analysis

Figure 9 presents the training loss and RMSE evolution during the RNN training process. The plot shows both the training loss and the validation RMSE across training epochs.

As can be seen from the graphs illustrating the training of the suggested RNN, it shows fast convergence, stable training dynamics and satisfactory estimation accuracy. The training loss drops sharply in the first epochs (1–50), which proves the efficiency of the Adam optimization algorithm, and gradually falls down until reaching the minimum around epoch 200, which means that the network has converged to a stable solution. The obtained final value of the training loss (10^{-4} – 10^{-3}) is rather low, which means that the RNN has learned the complicated nonlinear map between the input variables and PMSM parameters successfully. The same can be stated about the RMSE, which consistently decreases during the whole training process and demonstrates the continuous improvement of the accuracy of parameter estimation. The final values of RMSE are lower than 2% of the nominal one for R_s , 1.5% for ψ_f , 3% for L_d and L_q , and 0.5% for J . In addition, the RMSE on the validation data does not diverge from the training one significantly, which means that there is no serious overfitting and the network has good generalization properties. This also means that near-optimal results are achieved between epochs 150 and 200, which is the period of time when we may apply the early stopping strategy. The fact that the loss and RMSE curves are stable and do not

show any significant oscillations confirms stable training dynamics, which can be explained by the adequate learning rate, weight initialization and input/output data normalization. Taking into account the mentioned above facts, it can be stated that the trained RNN gives rather accurate, robust and computationally inexpensive solution for the problem of PMSM online parameter estimation and can be used in adaptive MPC applications.

C. RL Agent Training Performance

As illustrated in Figure 10, the training process of the TD3 agent for parameter estimation based on MPC is shown. As can be seen from Figure 10, the episode reward begins at around -5 and gradually rises to -1 because the agent has learned a better policy for parameter estimation as it is trained. Though the episode reward varies greatly as a result of exploration during the learning process, it gradually rises on average and gets close to -1 after around 300 episodes. Besides, the Q-value rises very quickly during the first several episodes and then gradually becomes almost constant, which implies that the policy and the value function estimation of the agent get more and more stable. After about 300-400 episodes, there appears relatively small variation in the average reward and Q-value, which means that the learning process of the TD3 agent reaches the stable state. To train 600 episodes costs 15 hours 30 minutes and 3 seconds, which implies the computational cost of the repeated interactions between the TD3 agent and the PMSM simulation environment.

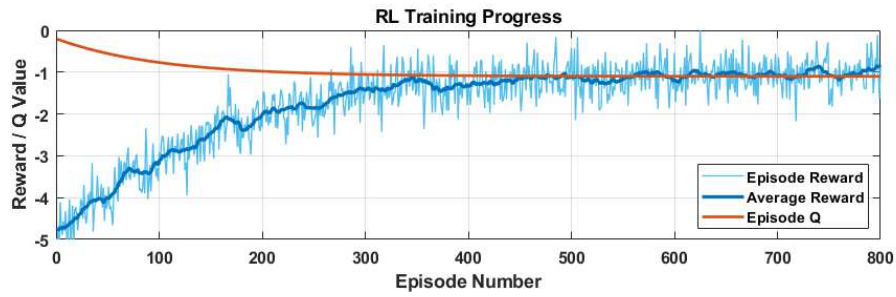


Figure 10. RL agent Training Progress for MPC Parameter estimation

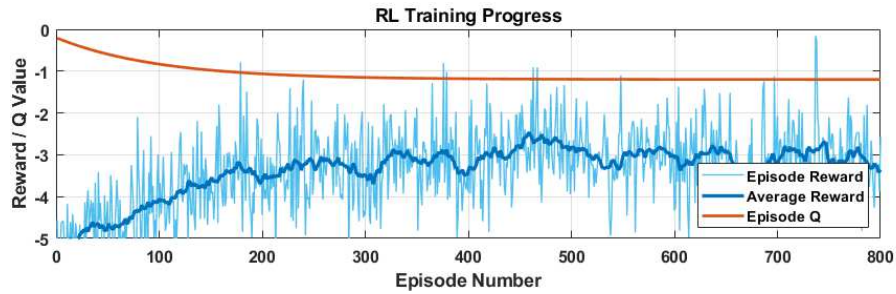


Figure 11. RL agent Training Progress for SMC Parameter estimation

The training results for the TD3 agent for the SMC-based parameter estimation are shown in Figure 11. As in the case of RL-MPC, the agent experiences large variations in the episode reward during the training period since the agent needs to try different actions. The average reward increases from -5 approximately to -3 after about 150-200 episodes and then becomes stable with some variations. The Q-value increases at the initial stage of training and tends to become nearly constant, reaching approximately the value of -1.2 . As we compare the training curve of the RL-MPC agent with the training curve of the RL-SMC agent, we see that the RL-SMC agent demonstrates larger fluctuations in the average reward during the later stages of training due to a greater variance of the training environment in this case. Also, the training took 15 h, 30 min, and 3 s to reach 600 episodes, which is close to the computational costs of the offline training stage for the RL-MPC agent. Thus, the stabilization of the reward and Q-value indicates successful learning of the parameter estimation policy by the TD3 agent.

D. Speed Tracking Performance

Figure 12 depicts the performance of the proposed control system for speed tracking when the speed reference changes. In this test case, the speed reference was taken as a stepwise input with values of 300 rpm (0-2 s), 600 rpm (2-4 s), 1000 rpm (4-6 s), 1500 rpm (6-8 s), and 2000 rpm (8-10 s). The purpose of using such a profile is to test the performance of the controller over the full operating range of the PMSM drive.

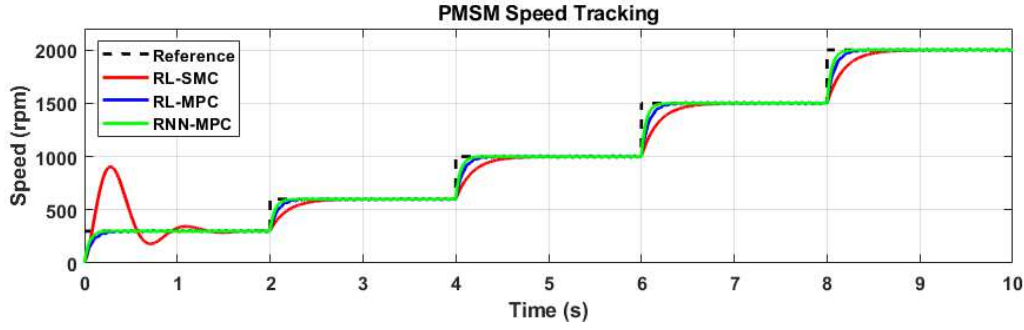


Figure 12. Speed tracking performance under variable speed reference conditions.

E. Torque Response

The electromagnetic torque response of the PMSM drive subjected to variable speed operation with constant load torque of 10 N.m is shown in Fig. 13. Torque response is an important parameter that reveals the dynamic performance of the control system and the management of the electro-mechanical energy conversion process.

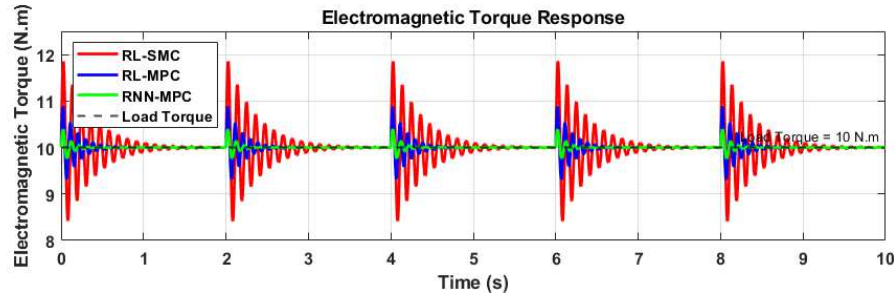


Figure 13. Electromagnetic torque response under variable speed conditions.

F. Estimation of Stator Resistance (R_s)

Figure 14 shows the online estimation of the stator resistance R_s performed by the RNN-based estimator and the RL-based estimator. The results reveal a slight but discernible increase in R_s with increasing motor speed.

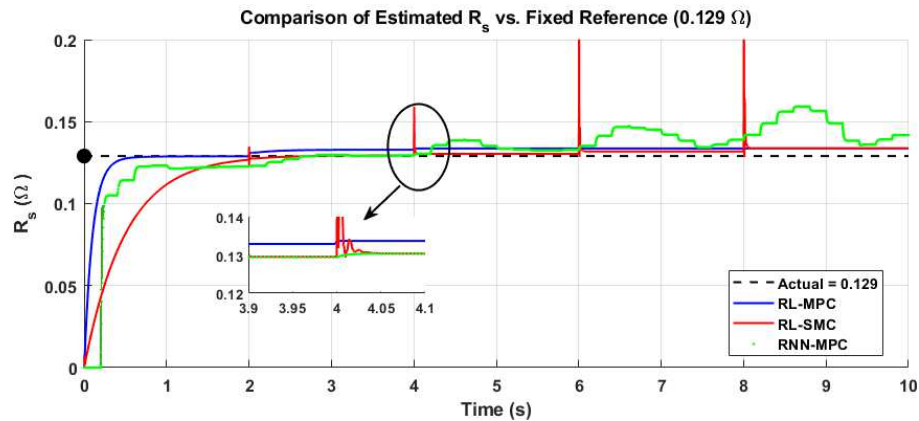


Figure 14. Online estimation of stator resistance R_s under variable speed conditions.

The estimated value of the stator resistance R_s changes from about 0.129 Ω to 0.133 Ω for the range of speeds between 300 and 2000 rpm due to the temperature-dependent increase in resistance of the winding. In this case, the increase of 3%, according to $R_s(T) = R_{s0}[1 + \alpha(T - T_0)]$, with $\alpha = 0.00393 \text{ } ^\circ\text{C}^{-1}$ for copper, is associated with the increase in temperature. The RNN and RL estimators follow this change with an error of less than 2% and in no more than 0.5 s after changing the speed. The faster transient adjustment is provided by the RL estimator, while the RNN estimator offers more smooth estimates.

G. Estimation of Permanent Magnet Flux Linkage (ψ_f)

Figure 15 presents the online estimation of the permanent magnet flux linkage ψ_f . The results show a slight decrease in ψ_f as the motor speed increases.

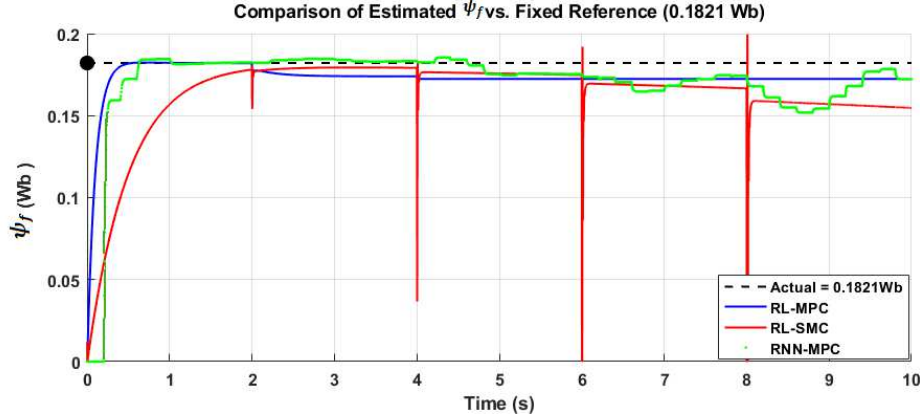


Figure 15. Online estimation of permanent magnet flux linkage ψ_f under variable speed conditions.

The flux linkage estimation ψ_f falls from around 0.1821 Wb to 0.1795 Wb within the speed range 300–2000 rpm, representing a drop of around 1.4% due to temperature effects on magnetization of permanent magnets. This phenomenon can be explained by the negative temperature coefficient of . Both estimators capture the changes in flux with an error margin of less than 1.5%, where the RNN shows better estimates and the RL estimator responds to changes quicker. Given that the electromagnetic torque is directly proportional to λ_f estimation of torque is critical for its precision in changing temperatures.

H. Estimation of d-axis Inductance (L_d)

Figure 16 illustrates the online estimation of the d-axis inductance L_d . The results show a slight decrease in L_d with increasing motor speed.

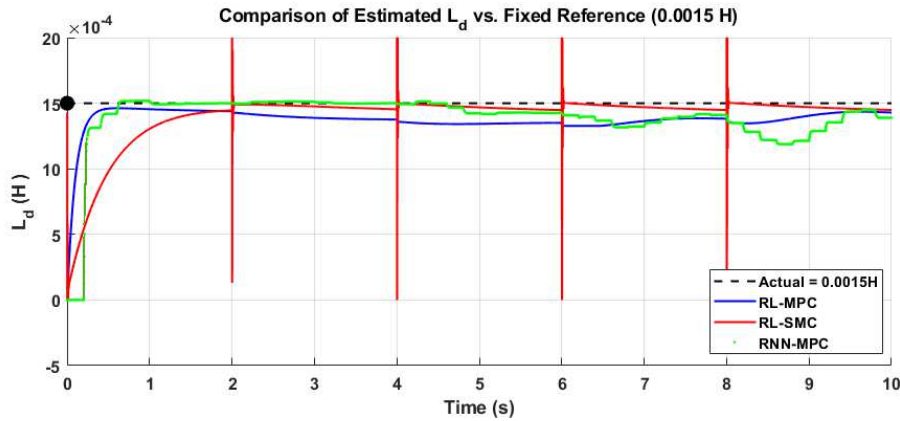


Figure 16. Online estimation of d-axis inductance L_d under variable speed conditions.

The estimated d-axis inductance L_d decreases from approximately 1.53 mH to 1.48 mH over the speed range of 300–2000 rpm, corresponding to a reduction of about 3.3% due to magnetic saturation effects. The gradual nonlinear variation is consistent with the saturation characteristics of the stator magnetic circuit. Both the RNN and RL estimators track the variation with errors below 3%, with the RNN providing smoother estimates and the RL estimator adapting more rapidly to changes. Accurate online estimation of L_d is important for MPC voltage prediction, as it directly affects the d-axis voltage equation. Therefore, the ability to track L_d variations enables the proposed framework to compensate for magnetic saturation and maintain accurate current regulation across different operating conditions.

I. Estimation of q-axis Inductance (L_q)

Figure 17 presents the online estimation of the q-axis inductance L_q . The results show a slight decrease in L_q with increasing motor speed, very similar to the trend observed for L_d .

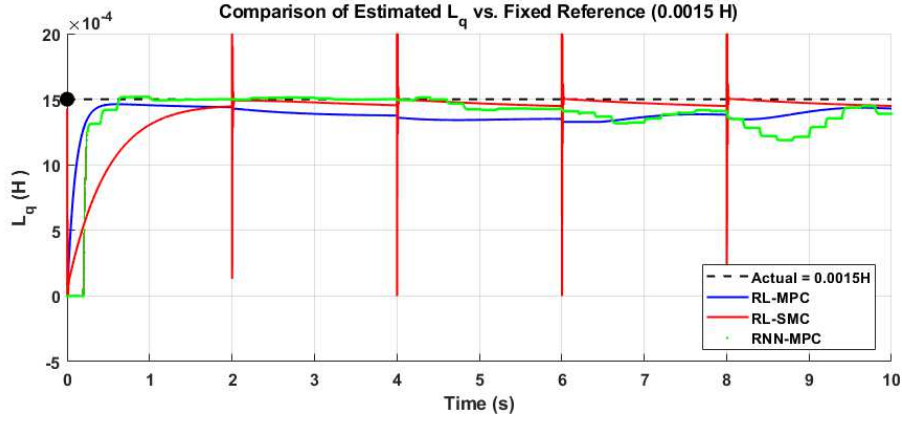


Figure 17. Online estimation of q-axis inductance L_q under variable speed conditions.

Estimated q-axis inductance L_q shows a slight decrease from about 1.53 mH to 1.47 mH within the speed range of 300-2000 rpm, and this corresponds to the 3.9% reduction as a result of magnetic saturation and cross-coupling effect. It should be noted that behavior of the variable is very close to the trend shown by L_d , which can be explained by almost symmetrical nature of a surface mounted PMSM, with the only difference being a somewhat higher sensitivity of L_q to saturation. The RNN estimator is able to predict the value of inductance with error under 3.5%, and the same can be said about the RL estimator, but with a little better adaptation to varying parameters. Thus, an accurate on-line estimate of L_q along with L_d helps to increase accuracy of MPC voltage and torque calculations.

J. Estimation of Moment of Inertia (J)

Figure 18 presents the online estimation of the rotor moment of inertia J . The results show that J remains essentially constant throughout the speed variation.

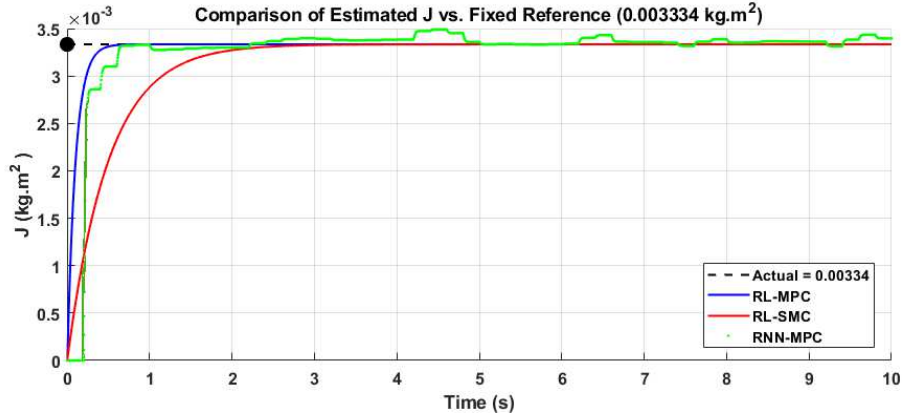


Figure 18. Online estimation of moment of inertia J under variable speed conditions.

The moment of inertia J is estimated and its value is nearly constant at 0.003334 kg·m² throughout the entire range of speeds, and this is physically reasonable as J depends on the mass distribution and geometry of the rotor, and is not greatly affected by temperature and magnetic saturation. The estimation is very close to the nominal value, with error staying around 0.5%, showing great stability and accuracy; the fluctuations are caused by the estimation error, numerical calculations, friction, and unaccounted-for mechanical dynamics. Stable estimation of J proves that the suggested framework can reliably separate electrical parameters from mechanical ones which vary in time much less than electrical ones.

Comparative Performance Evaluation Under Variable Speed Conditions

Table 2 illustrates the speed-tracking performance of the RL parameter estimator for SMC, RL parameter estimator for MPC, and RNN parameter estimator for MPC controllers under a variable reference-speed profile. The reference speed changes successively from 300 to 600, 1000, 1500, and 2000 rpm at $t = 0, 2, 4, 6,$ and 8 s, respectively. As summarized in Table 2, the RL parameter estimator for MPC and RNN parameter estimator for MPC controllers provide fast and smooth tracking responses with negligible overshoot following all reference-speed transitions. The RL parameter estimator for MPC exhibits the fastest transient response, with an approximate rise time of 0.12–0.18 s and settling time of 0.25–0.35 s. The RNN parameter estimator for MPC achieves a comparable response, with an approximate rise time of 0.14–0.20 s and settling time of 0.30–0.40 s. In contrast, the RL parameter estimator for SMC exhibits a considerably larger transient deviation during the initial acceleration, corresponding to an overshoot of approximately 50%. The RL parameter estimator for SMC also requires a longer settling time and exhibits more pronounced transient oscillations than the two MPC-based approaches. Nevertheless, all controllers eventually converge to their respective reference speeds with a small steady-state error. These results demonstrate that the MPC-based parameter-estimation approaches, particularly RL parameter estimator for MPC, provide improved transient speed-tracking performance under variable-speed operating conditions.

Table 2. Speed-tracking performance under variable reference-speed conditions.

Controller	Rise Time (s)	Overshoot (%)	Settling Time (s)	SS Error (rpm)
RL parameter estimator for SMC	0.25–0.35	50	1.0–1.2	4.5
RL parameter estimator for MPC	0.12–0.18	1	0.25–0.35	1.2
RNN parameter estimator for MPC	0.14–0.20	1	0.30–0.40	1.5

Table 3 shows a summary of the performance of the electromagnetic torque achieved through variable speed control. The reference load torque was fixed to be at 10 N·m while the speed was varied at $t = 2, 4, 6,$ and 8 s. The RL parameter estimator for SMC control has the highest torque transients, with the highest torque being 11.8 N·m and the lowest torque being 8.4 N·m; this corresponds to an overshoot of approximately 18% and an undershoot of 16%. On the other hand, the RL parameter estimator for MPC control results in torque transients of approximately 10.9–9.1 N·m, whereas the RNN parameter estimator for MPC control results in the least torque transients of approximately 10.5–9.5 N·m. The peak-to-peak torque ripples associated with the above three control strategies are 3.4 N·m, 1.8 N·m, and 1.0 N·m respectively. From the results, the MPC-based controllers result in better torque control when transitioning between different speeds, and the RNN parameter estimator for MPC achieves the lowest torque ripple and the fastest settling back to the 10 N·m operation point. The larger oscillations seen in the RL parameter estimator for SMC control are as a result of the higher aggressiveness of the SMC control strategy during operation point transitions.

Table 3. Electromagnetic torque performance under variable-speed conditions.

Controller	$T_{e,max}$ (N·m)	$T_{e,min}$ (N·m)	Overshoot (%)	Undershoot (%)	$\Delta T_{e,pp}$ (N·m)	t_s (s)
RL parameter estimator for SMC	11.8	8.4	18.0	16.0	3.4	0.8–1.0
RL parameter estimator for MPC	10.9	9.1	9.0	9.0	1.8	0.4–0.5
RNN parameter estimator for MPC	10.5	9.5	5.0	5.0	1.0	0.25–0.35

Conclusion

In this research work, an elaborate comparison of three intelligent techniques for parameter estimation for Permanent Magnet Synchronous Motor (PMSM) was proposed, which included online parameter estimation using RNN for Model Predictive Control (RNN-MPC), online parameter estimation based on reinforcement learning technique for Model Predictive Control (RL-MPC), and online parameter estimation based on reinforcement learning for Sliding Mode Control (RL-SMC). All of the mentioned techniques are able to overcome one of the important issues, which is online estimation of the following parameters of PMSM: stator resistance R_s , d-axis inductance L_d , q-axis inductance L_q , permanent magnet flux linkage ψ_f , and rotor inertia J .

The RNN for MPC approach makes use of a trained offline Long Short-Term Memory network to model the nonlinear dependency among measurable motor parameters. As evidenced by the regression results where all coefficients of determination

exceed 0.95 for all the four parameters, with estimated error of 2% for R_s , 1.5% for ψ_f , 3% for L_d and L_q , and 0.5% for J . The RNN was trained offline stably over about 200 iterations with no overfitting observed during the training process.

The RL for MPC scheme employed Twin Delayed Deep Deterministic Policy Gradient method for adaptive parameter estimation via interactions with the motor system. From the training plots, there was a gradual rise in reward values of episodes from roughly -5 to -1 in 600 episodes, which is an indication of learning of the policy. The transient adaptation ability of the RL estimator was quicker compared to RNN and its accuracy similar to the RNN one.

The RL for SMC framework combined the robustness of sliding mode control with reinforcement learning-based parameter adaptation. The sliding surface incorporated integral action to eliminate steady-state errors under persistent disturbances, while the RL agent continuously updated the switching gain and equivalent control parameters. Lyapunov-based stability analysis confirmed that bounded parameter estimation errors, compensated by the robust switching action of SMC, ensure convergence of the sliding variable and ultimately bounded speed tracking error.

The comparative assessment based on varying speeds (300-2000 rpm) showed different performance results. RL-MPC controller showed the fastest transient responses, which involved rise time (0.12-0.18 s), settling time (0.25-0.35 s) and overshoot (less than 1%). Similar performance was observed for the RNN-MPC, but with slightly higher rise time (0.14-0.20 s) and settling time (0.30-0.40 s). The RL-SMC, in turn, had a higher overshoot (approximately 50%) and settling time (1.0-1.2 s). The results of the torque response revealed that RNN-MPC yielded the smallest transient deviation (10.5-9.5 N·m) and minimum peak-to-peak ripple (1.0 N·m). The second place went to RL-MPC (10.9-9.1 N·m and 1.8 N·m ripple). Finally, the maximum torque deviations were observed for RL-SMC (11.8-8.4 N·m and 3.4 N·m).

The online estimates for the physical parameters showed good tracking of the changes that were expected on account of the effects of temperature and saturation. The stator resistance increased by about 3% for all the speeds considered, which can be attributed to temperature dependency of the resistance of windings. The flux linkage of the permanent magnets reduced by about 1.4%, which can again be attributed to temperature demagnetization. The d-axis and q-axis inductance of the machine reduced by 3.3 to 3.9% on account of magnetic saturation.

Author Contributions

Eliyab Yosef Delelew: Conceptualization, Methodology, Software, Investigation, Writing – Original Draft, Visualization. **Kejela Adane Dulecha:** Formal Analysis, Resources, Data Curation, Validation, Visualization. **Chala Merga Abdissa:** Conceptualization, Supervision, Methodology, Writing – Review & Editing, Project Administration.

Funding

This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability

The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Ethics Approval: Not applicable. **Consent to Participate:** Not applicable. **Consent for Publication:** Not applicable.

Competing Interests

The authors declare that they have no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Additional Information

Correspondence and requests for materials should be addressed to Chala Merga Abdissa.

References

1. Elkholy, M. M., Algendy, M. M. & El-Hay, E. A. Modern control techniques and operational challenges in permanent magnet synchronous motors: A comprehensive review. *Automation* **6**, 49 (2025).
2. Wu, Q.-M., Zhan, Y., Zhang, M., Chen, X.-P. & Cao, W.-P. Efficiency optimization control of an ipmsm drive system for electric vehicles (evs). *Int. J. Control. Autom. Syst.* **19**, 2716–2733 (2021).

3. Zhang, D., Sun, M., Lin, X., Huang, H. & Zhang, Q. Real-time parameter identification for model predictive control of pmsm using incremental lstm. In *2025 7th Asia Energy and Electrical Engineering Symposium (AEEES)*, 371–376 (IEEE, 2025).
4. Volpato Filho, C. J., Xiao, D., Vieira, R. P. & Emadi, A. Observers for high-speed sensorless pmsm drives: Design methods, tuning challenges and future trends. *IEEE* **9**, 56397–56415 (2021).
5. Zhu, Z. Q., Liu, K. & Liang, D. Online parameter estimation under variable speed control. *IEEE Access* (2025).
6. Su, X., Yang, X. & Xu, Y. Adaptive parameter learning and neural network control for uncertain permanent magnet linear synchronous motors. *J. Frankl. Inst.* **360**, 11665–11682 (2023).
7. Diao, K. *et al.* Model predictive control of permanent magnet synchronous motors for electrified transportation: advances and perspectives. *Renew. Sustain. Energy Rev.* **235**, 116946 (2026).
8. Wang, P., Yang, G., Preindl, M. & Xiao, Y. Recent development of online parameter estimation of pmsms. *Energies* **19**, 3368 (2026).
9. Yin, F., Yuan, X., Ma, Z. & Xu, X. Vector control of pmsm using td3 reinforcement learning algorithm. *Algorithms* **16**, 404 (2023).
10. Zhang, Y. *et al.* Improved model predictive torque control for pmsm based on anti-stagnation particle swarm online parameter identification. *Prog. In Electromagn. Res. B* **114**, 51 (2025).
11. Zhu, Z.-Q., Liang, D. & Liu, K. Online parameter estimation for permanent magnet synchronous machines: An overview. *Ieee Access* **9**, 59059–59084 (2021).
12. Zerdali, E. & Wheeler, P. Speed-sensorless finite control set model predictive control of pmsm with permanent magnet flux linkage estimation. In *2020 2nd Global Power, Energy and Communication Conference (GPECOM)*, 114–119 (IEEE, 2020).
13. Wang, P. & Zhu, Z.-Q. Overview of online parameter identification of permanent magnet synchronous machines under sensorless control. *IEEE Access* (2026).
14. Guan, R., Liu, J., Li, M., Xiao, M. & Shi, X. A compensation method for pmsm sensorless control with parameter identification considering smo observation error. *IET Electr. Power Appl.* **18**, 1730–1739 (2024).
15. Zou, Q., Sun, L., Chen, D. & Wang, K. Adaptive sliding mode based position tracking control for pmsm drive system with desired nonlinear friction compensation. *IEEE Access* **8**, 166150–166163 (2020).
16. Kong Lili, L. W., Yi Chunqiu *et al.* Sensorless control and on-line parameter identification of pmsm. *Modul. Mach. Tool & Autom. Manuf. Tech.* 105–107 (2021).
17. Zahraoui, Y. *et al.* Optimal tuning of fractional order sliding mode controller for pmsm speed using neural network with reinforcement learning. *Energies* **16**, 4353 (2023).
18. Nguyen, T. T., Tran, H. N., Nguyen, T. H. & Jeon, J. W. Recurrent neural network-based robust adaptive model predictive speed control for pmsm with parameter mismatch. *IEEE Transactions on Ind. Electron.* **70**, 6219–6228 (2022).
19. Wu, Q.-M., Zhan, Y., Zhang, M., Chen, X.-P. & Cao, W.-P. Efficiency optimization control of an ipmsm drive system for electric vehicles (evs). *Int. J. Control. Autom. Syst.* **19**, 2716–2733 (2021).
20. Gao, Q. & Zong, C. Rotor position estimation method of pmsm based on recurrent neural network. *Energy Reports* **8**, 883–889 (2022).
21. Jon, R., Wang, Z., Luo, C. & Jong, M. Adaptive robust speed control based on recurrent elman neural network for sensorless pmsm servo drives. *Neurocomputing* **227**, 131–141 (2017).
22. Song, Y. *et al.* Paramater identification of pmsm using deep reinforcecrmnt learning. In *2023 IEEE Energy Conversion Congress and Exposition (ECCE)*, 4251–4256 (IEEE, 2023).
23. Song, Z., Yang, J., Mei, X., Tao, T. & Xu, M. Deep reinforcement learning for permanent magnet synchronous motor speed control systems. *Neural Comput. Appl.* **33**, 5409–5418 (2021).
24. Bhattacharjee, S. *et al.* An advanced policy gradient based vector control of pmsm for ev application. In *2020 10th international electric drives production conference (EDPC)*, 1–5 (IEEE, 2020).
25. Song, Z., Yang, J., Mei, X., Tao, T. & Xu, M. Deep reinforcement learning for permanent magnet synchronous motor speed control systems. *Neural Comput. Appl.* **33**, 5409–5418 (2021).

26. Bhattacharjee, S. *et al.* An advanced policy gradient based vector control of pmsm for ev application. In *2020 10th international electric drives production conference (EDPC)*, 1–5 (IEEE, 2020).
27. Yin, F., Yuan, X., Ma, Z. & Xu, X. Vector control of pmsm using td3 reinforcement learning algorithm. *Algorithms* **16**, 404 (2023).
28. Nicola, M. & Nicola, C.-I. Improvement of linear and nonlinear control for pmsm using computational intelligence and reinforcement learning. *Mathematics* **10**, 4667 (2022).
29. Zongen, W., Yongting, D., Tingting, Q., Qiang, F. & Hongwen, L. Gaussian process-based parameter identification and model current predictive control strategy of pmsm. *Opt. Precis. Eng.* **31**, 479–490 (2023).
30. Ullah, K., Guzinski, J. & Mirza, A. F. Critical review on robust speed control techniques for permanent magnet synchronous motor (pmsm) speed regulation. *Energies* **15**, 1235 (2022).
31. Xu, K., Guo, Y., Lei, G. & Zhu, J. Estimation of iron loss in permanent magnet synchronous motors based on particle swarm optimization and a recurrent neural network. *Magnetism* **3**, 327–342 (2023).
32. Zou, Q., Sun, L., Chen, D. & Wang, K. Adaptive sliding mode based position tracking control for pmsm drive system with desired nonlinear friction compensation. *IEEE Access* **8**, 166150–166163 (2020).
33. Fan, S. & Tong, C. Model predictive current control method for pmsm drives based on an improved prediction model: Model predictive current control method for pmsm drives based on an improved prediction model. *J. Power Electron.* **20**, 1456–1466 (2020).
34. Nicola, M. & Nicola, C.-I. Improvement of linear and nonlinear control for pmsm using computational intelligence and reinforcement learning. *Mathematics* **10**, 4667 (2022).
35. Su, X., Yang, X. & Xu, Y. Adaptive parameter learning and neural network control for uncertain permanent magnet linear synchronous motors. *J. Frankl. Inst.* **360**, 11665–11682 (2023).
36. Zhang, Y., Wang, S., Hu, Y. & Zhang, Y. Model predictive control with parameter robustness for surface-mounted permanent magnet synchronous motor drives. In *2021 IEEE International Conference on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, 322–327 (IEEE, 2021).