UDC 621.313.333:681.5.015:004.032.26
DOI: 10.15587/1729-4061.2025.331895

OPTIMIZING PERMANENT MAGNET SYNCHRONOUS MOTOR CONTROL: A COMPARATIVE STUDY OF MPCC-BASED TECHNIQUES

Doan Van Hoa
Master Student*
Huynh Hoang Bao Nghia
College Student *
Le Van Dai
Corresponding author
Doctor of Technical Sciences*
E-mail: levandai@iuh.edu.vn
*Industrial University of Ho Chi Minh City
Nguyen Van Bao, 12, Go Vap District,
Ho Chi Minh, Vietnam, 70000

This study focuses on optimizing the control of permanent magnet synchronous motors (PMSMs) by introducing a new model predictive current control (MPCC) strategy, integrated with the adaptive neuro-fuzzy inference system (ANFIS), referred to as ANFIS-MPCC. The main problem addressed in this research is the challenge of improving the dynamic response of PMSMs under varying operating conditions, particularly under rapid load and speed variations. Traditional control methods like proportional-integral MPCC (PI-MPCC) and artificial neural network MPCC (ANN-MPCC) are compared with the proposed ANFIS-MPCC method to evaluate its effectiveness in solving issues such as overshoot reduction, settling time minimization, and harmonic distortion (THD) suppression. The results show that ANFIS-MPCC significantly outperforms the traditional methods, with overshoot reduced to 0.015%, a settling time of 0.00147 seconds, and THD minimized to 2.0% at rated speed and 2.02% at low speed. These improvements demonstrate that ANFIS-MPCC is highly effective in controlling PMSMs, particularly in systems exposed to rapid load changes and dynamic speed variations. The method's key advantage lies in its integration of fuzzy logic and neural networks, allowing superior handling of nonlinearities and dynamic load conditions. The results suggest that ANFIS-MPCC is especially beneficial for industrial motor control systems and electric vehicles, where fast response, stability, and low harmonic distortion are crucial

Keywords: PMSM, MPCC, ANFIS, ANN, THD, industrial drives, electric vehicles

Received 01.04.2025
Received in revised form 14.05.2025
Accepted date 02.06.2025
Published date 30.06.2025

How to Cite: *Hoa, D. V., Nghia, H. H. B., Dai, L. V. (2025). Optimizing permanent magnet synchronous motor control: a comparative study of MPCC-based techniques. Eastern-European Journal of Enterprise Technologies, 3 (2 (135)), 73–89. https://doi.org/10.15587/1729-4061.2025.331895*

1. Introduction

In modern industrial contexts, optimizing the performance of electric motors is a crucial requirement, particularly for permanent magnet synchronous motors (PMSMs), which are widely used in applications such as electric vehicles, robotics, and industrial automation systems. With the ability to provide high power and superior efficiency, PMSMs have become a popular choice due to their reliability and compact size. However, controlling PMSMs under dynamic load conditions and high-speed demands remains a significant challenge for traditional control methods.

Among the various control methods, model predictive control (MPC) stands out due to its ability to predict and optimize system behavior. However, MPC faces a major issue with computational complexity, especially when real-time calculations are needed. This limits its application in environments where fast response times and computational efficiency are critical.

Recent studies have shown that integrating MPC with machine learning methods, such as artificial neural networks (ANN) and adaptive neuro-fuzzy inference systems (ANFIS), can reduce computational load and improve control performance, especially in applications that require high stability and adaptability, such as PMSMs. However, applying these methods to PMSM control still presents challenges, particularly in optimizing computational efficiency and real-time system implementation.

Therefore, research into advanced PMSM control methods, such as ANFIS-based on model predictive current control (ANFIS-MPCC), is of utmost importance. This approach not only reduces total harmonic distortion (THD) but also improves system stability and provides faster response times to sudden load changes. Furthermore, it opens up opportunities for real-world applications in industries, electric vehicles, and renewable energy systems. Despite significant progress in MPCC and ANFIS research, gaps still remain in computational optimization and practical applications, making further research into ANFIS-MPCC highly necessary.

2. Literature review and problem statement

The study [1] proposed an MPCC-based approach with an improvement called hybrid flux predictor-based predictive flux control (HFP-PFC). It combines both voltage models and current models to improve the accuracy of predicting the stator flux, enhancing the control performance, especially under varying load conditions or when motor parameters are mismatched. Although the HFP-PFC method offers significant improvements in accuracy and stability for PMSM motor control, it still has limitations in terms of computational complexity, hardware requirements, and dependence on motor parameters.

A control solution for motors using MPCC with extended voltage vectors is introduced in the study [2] to enhance motor control performance, reduce computational complexity, and implement zero-sequence current suppression to improve system stability under varying load and parameter

conditions. However, this method has limitations, such as poor control performance under rapidly changing load conditions, the need for powerful hardware to handle real-time computations, and the potential for over-modulation if the voltage vector amplitude is not properly controlled. It is also not optimal for applications that require fast and precise responses in high-dynamics environments.

The study [3] developed a virtual MPC-based ANN-MPCC for PMSM drives to address the high computational burden of traditional MPC. Its primary focus was to replace online optimization with a lightweight, offline-trained neural network, enabling faster real-time control execution. While effective in maintaining control performance and reducing computational load, the study did not address adaptability to dynamic changes in system parameters or external disturbances, limiting its robustness in practical, highly variable environments.

The three-vector model-free predictive control (TV-MFPC) method is developed in [4] to address the performance degradation issues in predictive control systems caused by parameter mismatches and dead-zone effects of inverters in PMSM. This method reduces reliance on precise motor models while effectively compensating for dead-zone effects and improving the stability of the control system. However, the method has not yet fully addressed the issue of optimization under rapidly changing motor parameters or significant disturbances, and it still requires complex computations in the control process.

In [5], an FCS-MPCC method with an online virtual vector synthesis strategy is proposed to solve the control error caused by voltage vector discretization in multi-phase systems. This method not only improves control performance but also helps reduce harmonic distortion and computational load compared to traditional methods. However, the method still faces challenges in fully covering the voltage vector space and handling higher-order harmonic waves, and it will require further research and improvement in the future to achieve optimal performance in practical applications.

An improved ANFIS model predictive current control approach is proposed in [6], significantly reducing torque ripple and improving dynamic speed response for electric vehicles driven using PMSM. However, the method faces challenges with computational efficiency under different drive cycles.

In [7], the ANFIS controller is developed to improve load variation and speed tracking. However, issues like overshoot still persist, particularly when using PI controllers. ANFIS improves control compared to traditional controllers but still struggles with peak overshoot in certain scenarios.

The use of particle swarm optimization (PSO) combined with fuzzy logic is presented in [8] to solve multi-criteria optimization problems in PMSM drives. While this approach improves optimization performance, it still struggles with real-time adaptability, especially when load conditions change rapidly.

Research [9] presented a multiple-vector-based MPCC approach combined with fuzzy logic for controlling PMSM. This method effectively reduces torque ripple and current harmonics, improving system stability. However, it faces significant issues related to computational complexity, especially in multi-phase systems, where calculating all candidate voltage vectors requires substantial computational resources. Additionally, multi-criteria optimization remains unresolved because it is difficult to simultaneously consider multiple fuzzy parameters under varying conditions.

After analyzing the research studies [1–9], it can be concluded that it is advisable to conduct further research on control methods for PMSM, focusing on reducing computational burden, improving adaptability to changes in load and motor parameters, and optimizing performance in practical applications. This is particularly important in environments with rapid fluctuations in load conditions, motor parameters, and the need for harmonic reduction. The combination of ANFIS and MPCC offers a promising approach and continues to be further developed in this research. In this approach, MPCC utilizes predictive models to calculate and optimize the current and torque in the PMSM motor, while ANFIS enhances the control process of MPCC by integrating the learning capabilities of neural networks and fuzzy logic.

3. The aim and objectives of the study

The aim of the study is to optimizing the control of PMSM using the ANFIS-MPCC method. This will make it possible to application ANFIS-MPCC method in industrial systems and electric vehicles, where rapid response and minimizing harmonic distortion are crucial. To achieve this aim, the following objectives are accomplished:

– to develop the ANFIS-MPCC control strategy applying the speed controller;

– to evaluate the dynamic response of PMSM under various operating conditions;

– to analyze the THD of the stator current of PMSM when operating with the rated load at both low and rated speeds.

4. Materials and methods of research

4. 1. The object and hypothesis of the study

This research focuses on the PMSM control system employing the MPCC method, as shown in Fig. 1. The system consists of a speed controller and a current controller, combined with PI, ANN, and ANFIS methods in the MPCC approach.

The detailed control process for PMSM includes a speed controller utilizing PI, ANN, and ANFIS techniques to adjust speed, a current controller that generates control signals for d-axis and q-axis currents, and a two-level voltage source inverter (2LVSI) that converts DC voltage to AC voltage for PMSM operation.

Based on the expectation that the ANFIS-MPCC method will outperform the PI-MPCC and ANN-MPCC methods in controlling PMSM under varying load and speed conditions, the following hypotheses are proposed:

1. Hypothesis 1: the ANFIS-MPCC method will significantly reduce overshoot and settling time when PMSM operates under strong load and speed changes, thanks to the ability to handle nonlinear relationships and the flexibility of the ANFIS system.

2. Hypothesis 2*:* the ANFIS-MPCC will minimize THD in the stator current of PMSM, thereby maintaining stable efficiency and minimizing energy efficiency issues under sudden load changes.

3. Hypothesis 3: when PMSM faces sudden speed changes or reversals, the ANFIS-MPCC method will perform better than the PI-MPCC and ANN-MPCC, with higher stability and more accurate control.

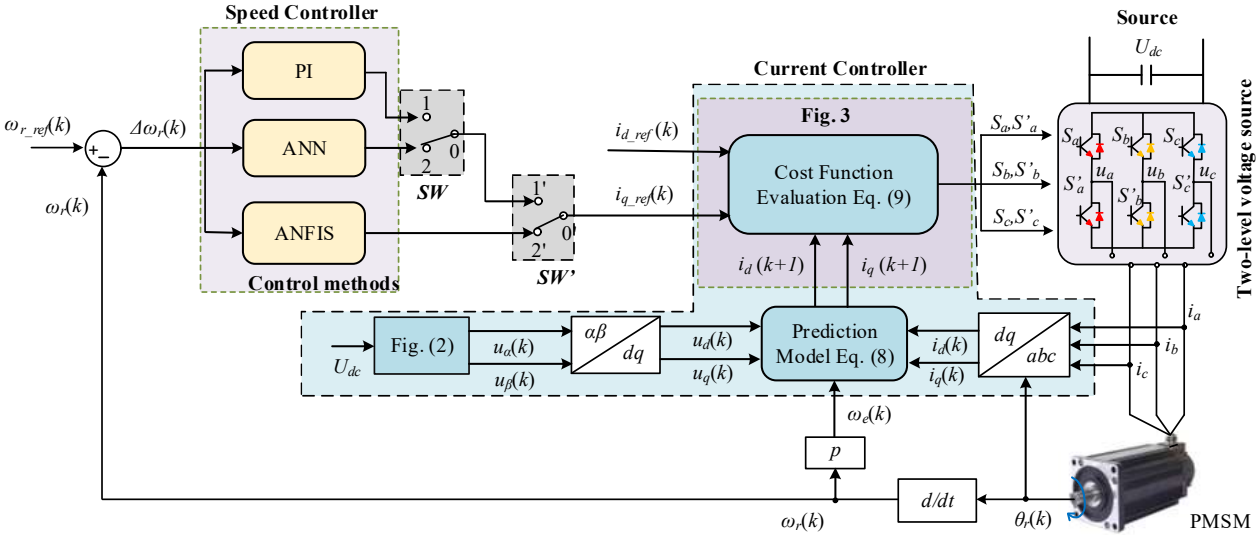


Fig. 1. The control system of the permanent magnet synchronous motor

4. 2. Theoretical methods

4. 2. 1. Permanent magnet synchronous motor

The mathematical model of PMSM can be expressed under the *d-q* rotating reference frame as follows [10]

$$\begin{bmatrix} \frac{di_d(t)}{dt} \\ \frac{di_q(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & \frac{\omega_e(t)\cdot L_q}{L_d} \\ \frac{\omega_e(t)\cdot L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} i_d(t) \\ i_q(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} u_d(t) \\ u_q(t) - \omega_e(t)\cdot\varphi_f \end{bmatrix}, \quad (1)$$

where i_d and i_q – the d and q axis currents, respectively; L_d and L_q – the d and q axis inductances, respectively; R_s – the stator resistance; ω_e – the rotor mechanical angular velocity; φ_f – the rotor flux linkage; u_d and u_q – the d and q axis voltages, respectively.

The dynamic model of the rotating system is expressed as

$$T_e = T_L + J\frac{d\omega_m(t)}{dt} + B\cdot\omega_m(t), \quad (2)$$

in which T_L – the load torque; J – the inertia torque; B – the viscous friction coefficient; ω_m – the rotor's angular speed; T_e – the electromagnetic torque, and can be determined as follows

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

where p – the number of pole pairs.

4. 2. 2. Two-level voltage source inverter

This study employs the 2LVSI as a power converter to convert DC to AC voltage and power the PMSM. Each phase leg of the inverter features two controllable switches, generating eight possible switching states. Among these, six states produce active voltage vectors, including V_1–V_6, while two states, including V_0 and V_7, are null vectors with no voltage. The voltage and current of the three-phase system are typically represented in a three-axis coordinate system, as depicted in the switching table. Fig. 2 illustrates all the possible switching states and voltage vectors generated by the 2LVSI, including six active vectors and two zero vectors

$$\begin{cases} u_{abc} = \begin{bmatrix} u_a & u_b & u_c \end{bmatrix}^T; \\ i_{abc} = \begin{bmatrix} i_a & i_b & i_c \end{bmatrix}^T, \end{cases} \quad (4)$$

where the stator voltages in phases a, b, c are denoted as u_a, u_b, u_c, respectively, and the stator current in these phases is represented by i_a, i_b, i_c, respectively.

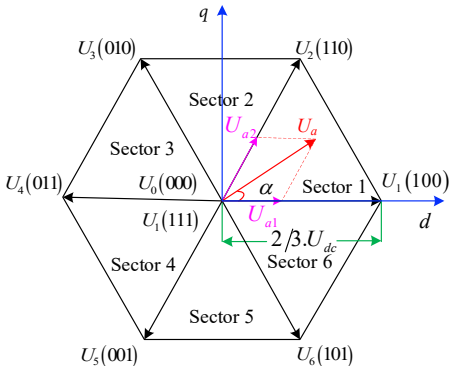


Fig. 2. Schematic of the voltage vectors of the two-level voltage source inverter

To simplify the analysis and control of the PMSM, this study uses the $\alpha\beta$ and dq coordinate systems to describe the stator voltage in a two-dimensional space. The stator voltage in the $\alpha\beta$ and dq coordinate systems is defined, respectively, as follows

$$\begin{cases} \begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \end{bmatrix} \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix}; \\ \begin{bmatrix} u_d \\ u_q \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix}. \end{cases} \quad (5)$$

The applied voltage to the PMSM from the 2LVSI can be expressed as follows

$$u_{abc} = \begin{bmatrix} S_a & S_b & S_c \end{bmatrix}^T \cdot \frac{U_{dc}}{2}, \quad (6)$$

where the notation with a superscript T in (4) and (6) indicates the transpose of the vector or matrix involved; U_{dc} – the DC-link voltage; S_a, S_b, S_c – the switching functions.

4. 2. 3. Current control

Based on the MPCC approach, the diagram in Fig. 1 illustrates how the control signals from the current controller are optimized to maintain high performance for the PMSM. This method uses prediction models and a cost function to calculate the current values for the next time step, from which the optimal control signals are determined. This process helps improve the system's response under varying load conditions while ensuring stable and efficient motor operation. The Euler method is applied to discretize the continuous state equation of the motor, allowing the calculation of optimal voltage vectors and adjusting the current accordingly to match the actual operating conditions.

Considering the sampling interval as T_s, at sampling time k, the future predicted values of the dq-axis stator current are determined as follows [11]

$$\begin{bmatrix} i_d(k+1) \\ i_q(k+1) \end{bmatrix} = \begin{bmatrix} i_d(k) \\ i_q(k) \end{bmatrix} + T_s \begin{bmatrix} \frac{di_d(t)}{dt} \\ \frac{di_q(t)}{dt} \end{bmatrix}, \quad (7)$$

where T_s – the sample period; $i_{dq}(k)$ and $i_{dq}(k+1)$ – the dq-axis current state variables at the k^{th} and $(k+1)^{\text{th}}$, respectively.

By substituting (1) into (7), the future predicted values of the dq-axis stator current are re-determined as follows

$$\begin{bmatrix} i_d(k+1) \\ i_q(k+1) \end{bmatrix} = \begin{bmatrix} i_d(k) \\ i_q(k) \end{bmatrix} + T_s \begin{bmatrix} -\frac{R_s(k)}{L_d} & \frac{\omega_e(k)\cdot L_q}{L_d} \\ \frac{\omega_e(k)\cdot L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} i_d(k) \\ i_q(k) \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} u_d(k) \\ u_q(k) - \omega_e(k)\cdot\varphi_f \end{bmatrix}. \quad (8)$$

The cost function, which defines the desired behavior of the system, is highly flexible and allows for the inclusion of system constraints. It can be expressed as [12]

$$const = \left(i_{d\_ref}(k) - i_d(k+1)\right)^2 + \left(i_{q\_ref}(k) - i_q(k+1)\right)^2. \quad (9)$$

Based on the discrete motor mathematical model, the predicted values of the current for all possible voltage vectors are calculated. The optimal voltage vector is then applied using the cost function for the next control cycle. The MPCC algorithm is designed to improve steady-state performance, as shown in Fig. 3, in which N_V represents the count of voltage vectors for both real and virtual, corresponding to each region.

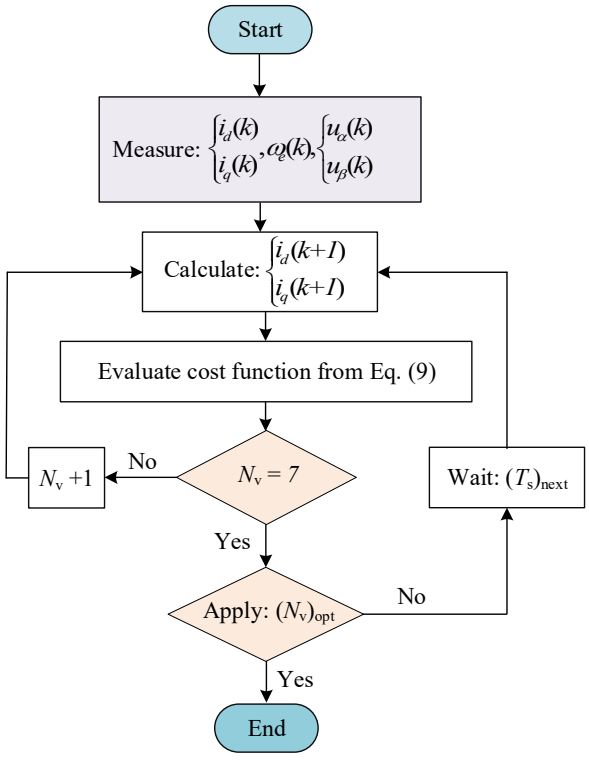


Fig. 3. Flow diagram of the model predictive current control model

The MPCC algorithm follows a structured calculation process, as illustrated in the flow diagram. Below are the key steps involved:

1. Measurement: key system parameters, including currents $i_d(k)$, $i_q(k)$, speed $\omega_e(k)$, and voltage components $u_\alpha(k)$, $u_\beta(k)$, are measured.

2. Current prediction: using the motor's mathematical model, the algorithm predicts the current values $i_d(k+1)$ and $i_q(k+1)$ for all possible voltage vectors.

3. Cost function evaluation*:* the algorithm evaluates the cost function (from eq. (9)) to determine how well the predicted currents meet the control objectives.

4. Voltage vector selection*:* the number of voltage vectors N_v corresponding to real and virtual regions is checked. N_v=7, the optimal voltage vector $(N_v)_{\text{opt}}$ is selected.

5. Application*:* the optimal voltage vector is applied in the next control cycle to adjust the motor's current and speed.

4. 2. 4. Speed control

The control strategy employs a speed controller that includes PI, ANN, and ANFIS techniques. In this paper, ANFIS is developed and presented in Subsection 5.1, while PI and ANN are used as references for comparison. The specific details are presented as follows:

1. Recall the proportional-integral method: From Fig. 1, the q-axis current reference at the k^{th} cycle is determined based on the PI controller as follows [13]

$$\begin{cases} i_{q\_ref}(k) = k_P \cdot \Delta\omega_r(k) + k_I \sum_{i=0}^{k} \Delta\omega_r(i); \\ \Delta\omega_r(k) = \omega_{r\_ref}(k) - \omega_r(k), \end{cases} \quad (10)$$

where $\Delta\omega_r(k)$ – the speed error at cycle k^{th}; k_P – the proportional gain; k_I – the integral gain.

In this paper, the k_P and k_I are determined based on the Ziegler-Nichols method [13]. Their values are 40 and 3, respectively.

2. Recall the artificial neural network method: in this paper, the ANN's structure for speed control is illustrated in Fig. 4 [14]. It is used to calculate the optimal q-axis current reference based on the reference speed ω_{r_ref}, the error between the reference and the actual speeds, namely $\Delta\omega_r(k) = \omega_{r\_ref}(k) - \omega_r(k)$. This structure consists of the following components:

1. Input layer: for this structure, the input layer of the neural network does not employ an activation function. This is because the primary role of the input layer is to receive and transmit the input signals to the subsequent layers without modifying or processing them. As a result, the input layer consists of two components, the error signal $\Delta\omega_r(k)$ and the bias b. The input layer provides these signals to the network for further processing.

2. Hidden layer: the network includes one hidden layer with three neurons, denoted as n_{11}, n_{12}, and n_{13}. Each neuron receives inputs from the input layer and computes a weighted sum of these inputs, followed by an activation function to produce the outputs a_{11} and a_{13}. The hidden layer performs a nonlinear input transformation, enabling the network to approximate complex relationships between inputs and outputs. The output of the i^{th} neuron in the hidden layer is given by

$$n_i(k) = \sum_{j=1}^{m} w_{ij}(k) \cdot x_j(k) + b_i, \tag{11}$$

where $x_j(k)$ represents the input from the input layer, $w_{ij}(k)$ represents the weight between the j^{th} input and the i^{th} hidden neuron, b_i is the bias for i^{th} hidden neuron, the output of each neuron is computed via the activation function $a_i = f_i(n_i)$, where f_i is typically a "*tanh*" or "*sigmoid*" function.

3. Output layer: the output layer consists of a single neuron that generates the final control signal $i_{q\_ref}(k)$. The neuron in the output layer computes a weighted sum of the outputs from the hidden layer neurons, followed by an activation function. The output of the network is thus computed as

$$n_{\text{out}}(k) = \sum_{i=1}^{n} w_{\text{out}}(k) \cdot a_i(k) + b_{\text{out}}, \tag{12}$$

and the final output is then given by

$$y(k) = f_{\text{out}}\left(n_{\text{out}}(k)\right), \tag{13}$$

where f_{out} is often the identity function, so that $y(k) = n_{out}(k)$, representing the control signal $i_{q\_ref}(k)$.

4. Training and weight update mechanism: the training of the neural network is achieved through Backpropagation and Gradient Descent algorithms. The network is trained to minimize the error between the predicted control signal and the target value.

The error at each time step k is computed as the difference between the predicted output $y(k)$ and the target value $y_{\text{target}}(k)$:

$$E(k) = \frac{1}{2} \cdot \left(y(k) - y_{\text{target}}(k)\right)^2. \tag{14}$$

The gradient of the error function with respect to the network weights is computed using the chain rule in Backpropagation. The error term for the i^{th} neuron in the hidden layer is given by

$$\delta_i(k) = \frac{\partial E(k)}{\partial a_i(k)} \cdot f_i'\left(n_i(k)\right), \tag{15}$$

where $f_i'\left(n_i(k)\right)$ – the derivative of the activation function f_i with respect to its input.

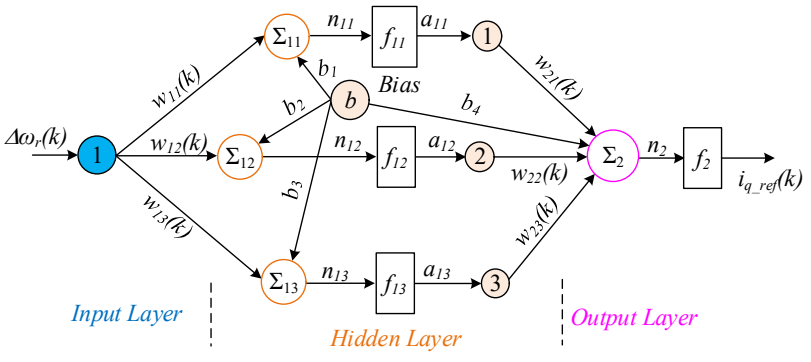


Fig. 4. Artificial neural network control architecture

Once the gradients are computed, the weights and biases are updated using the Gradient Descent algorithm

$$\begin{cases} w_{ij}(k+1) = w_{ij}(k) - \alpha \cdot \dfrac{\partial E(k)}{\partial w_{ij}(k)}; \\ b_i(k+1) = b_i(k) - \alpha \cdot \dfrac{\partial E(k)}{\partial b_i(k)}, \end{cases} \tag{16}$$

where α – the learning rate.

The training results and performance evaluation of the ANN are shown in Fig. 5.

The minimum value of mean squared error to reach the convergence condition is 0.080371 after training 50 epochs, and the parameter changes of the ANN training process can be seen in Fig. 5, *a*. As observed from Fig. 5, *b*, this is a scatter plot comparing the actual values with the predicted values. The regression fit closely follows the diagonal, indicating that the model has made accurate predictions. The correlation coefficient, $R = 1$, indicates perfect accuracy, meaning the predicted values match the target values exactly. Meanwhile, Fig. 5, *c* shows that the gradient decreases significantly across epochs, indicating that the optimization process is stabilizing and the mu value remains stable at 10^{-5}, suggesting a steady learning rate throughout training, contributing to the stability of the model. For validation failures, no validation failures are observed during training, which indicates that the model is not overfitting and is generalizing well to unseen data. Therefore, it can be concluded that the artificial neural network model has been successfully trained with high accuracy ($R = 1$), achieving low MSE and without overfitting. The training process is stable, with the gradient decreasing gradually and remaining constant. Overall, the model is performing well.

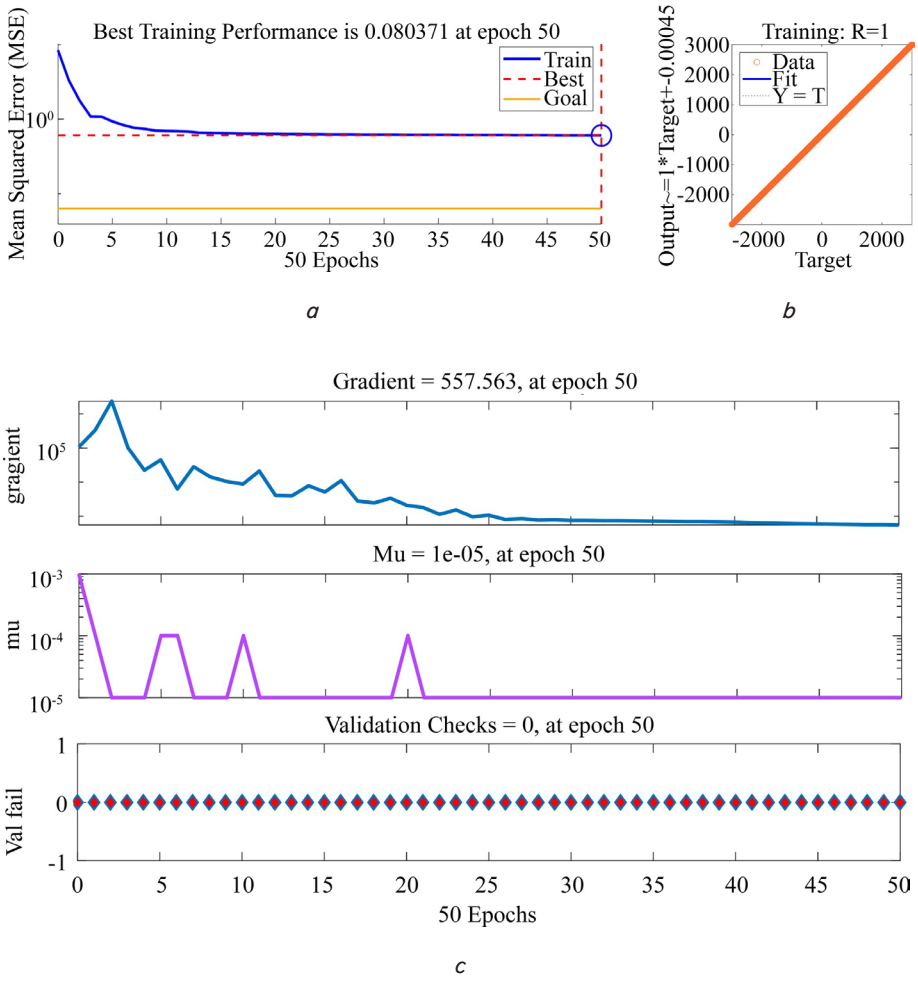


Fig. 5. Training result and performance evaluation of artificial neural network: *a* — convergence process; *b* — regression curve; *c* — training state

4. 3. Simulation setup and conditions

The PMSM drive model is based on three control methods, as shown in Fig. 1. The MATLAB/Simulink simulations of the drive system are constructed as shown in Fig. 6. For the tested PMSM model, parameters such as rated power, resistance, inductance, and torque were obtained from reliable sources in previous research and literature. These parameters are used to simulate the motor's behavior in a controlled environment, and the motor itself is not a physical, real-world engine. This approach allows to focus on evaluating the performance of the proposed control strategies without the complexities and variability of real hardware. The parameters of the motor model used for simulation are listed in Table 1.

The control methods were simulated in the same conditions to compare their performance. The controllers were evaluated based on their ability to handle dynamic load changes, speed reversals, and torque variations. The suitability of the control methods was verified through detailed simulations. Results such as overshoot, settling time, and THD were compared among the control methods to evaluate the performance of each method and demonstrate their superiority.

The data collected from the simulations included rotor speed, torque response, q-axis current, and THD in the stator current. The simulations were conducted under various scenarios, such as sudden load changes and speed reversals. The results were then analyzed to evaluate the control performance of each method in terms of efficiency, stability, and harmonic distortion.

Table 1

Parameters of the permanent magnet synchronous motor model used in the simulations

Parameter	Value	Parameter	Value
Rated Power P_r (kW)	5	d-axis inductance L_d (mH)	1.53
Rated Speed n_r (rpm)	3000	q-axis inductance L_q (mH)	1.53
Number of pole pairs	4	Rotor peak PM flux linkage φ_m (Wb)	0.175
Stator resistance R_s (Ω)	2.875	Viscous friction B_m (Ns/m)	10^{-6}
Total inertia J (kgm^2)	0.8×10^{-3}	Rated torque T_r (Nm)	25
DC bus Voltage (V)	500		

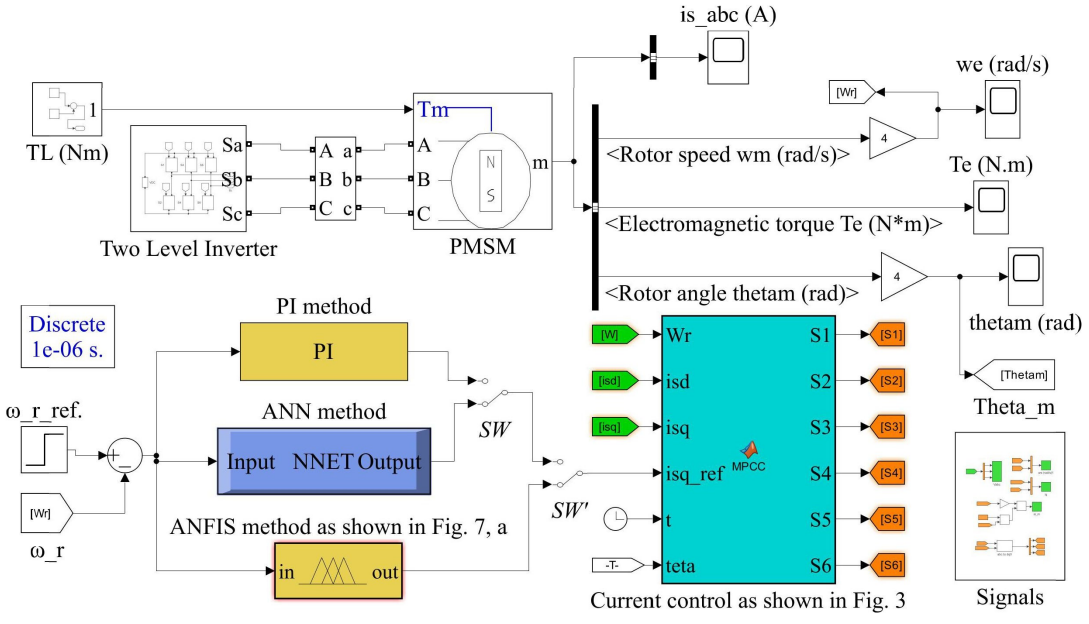


Fig. 6. The simulation model for permanent magnet synchronous motor drive system in the MATLAB/Simulink environment

5. Results of studying the methodology for optimizing the control method of the permanent magnet synchronous motor

5. 1. The proposed speed control using the adaptive neuro-fuzzy inference system based on model predictive current control

The proposed method is illustrated in Fig. 7 [15]. The architecture of the proposed ANFIS controller is shown in Fig. 7, *a*. It has two main inputs and one output, composed of five layers, each serving a specific function in the calculation and optimization process.

The first input is the error $\Delta\omega_r^e$ representing the difference between the sensed output and the reference speeds, while the second input is the rate of change of the speed error $\Delta\omega_r^{de}$. The system output corresponds to the reference current's peak value I^f.

The controller is normalized by applying scaling factors k_e, $k_{de,}$ and k_u to ensure optimal performance of the system

$$\begin{cases} I_{q\_ref}(t) = k_u \cdot I^f(t) + \int I_{q\_ref}(t)\,\mathrm{d}t; \\ \Delta\omega_r^e(t) = k_e \cdot \Delta\omega_r(t); \\ \Delta\omega_r^{de}(t) = k_{de} \cdot \dfrac{d}{dt}\Delta\omega_r(t), \end{cases} \tag{17}$$

where $\Delta\omega_r$ – the error between the reference and actual rotor speeds; $\Delta\omega_r^e$ is obtained from $\Delta\omega_r$ through the scaling factor k_e and $\Delta\omega_r^{de}$ – the derivative of $\Delta\omega_r$ through the scaling factor k_{de}. These two signals are the input to the ANFIS; I^f – the control signal, after processing through the fuzzy layers in the ANFIS system; I_{q_ref} – the final control signal calculated by the ANFIS system, and through a scaling factor k_u, which is the reference q-axis current signal.

As illustrated in Fig. 7, *b*, applying Sugeno fuzzy system rules, the if-then rules can be given as

$$\begin{cases} \text{Rule 1: If } \Delta\omega_r^e \text{ is } A_1 \text{ and } \Delta\omega_r^{de} \text{ is } B_1, \\ \text{then } I^f = p_1\Delta\omega_r^e + q_1\Delta\omega_r^{de} + b_1; \\ \text{Rule 2: If } \Delta\omega_r^e \text{ is } A_2 \text{ and } \Delta\omega_r^{de} \text{ is } B_2, \\ \text{then } I^f = p_2\Delta\omega_r^e + q_2\Delta\omega_r^{de} + b_2, \end{cases} \tag{18}$$

where $\Delta\omega_r^e$ and $\Delta\omega_r^{de}$ – the inputs to ANFIS; A_1 and A_2 – the fuzzy sets for error, described as low and high with values of $\Delta\omega_r^e$, respectively; B_1 and B_2 – the fuzzy sets for change in error, described as low and high with values of $\Delta\omega_r^{de}$, respectively; p_1, p_2 – the coefficients for the error in the two rules; q_1, q_2 – the coefficients for the change in error in the two rules; b_1, b_2 – the bias in the two rules; I_f – the output.

The ANFIS system encompasses fuzzification, rule evaluation, defuzzification, and output, representing five layers, each serving a distinct function as described below.

Each MF has a unique weight of unity. The node functions at layer 1 are calculated as follows

$$O_i^{L-1} = \mu_{A_i}\left(\Delta\omega_r^e\right), \; i = 1,2. \tag{19}$$

In layer 2, the fixed nodes calculate the firing strength using fuzzy AND logic. This logic computes the firing strength as follows

$$O_i^{L-2} = w_i = \mu_{Ai}\left(\Delta\omega_r^e\right) \cdot \mu_{B_i}\left(\Delta\omega_r^{de}\right), i = 1,2. \tag{20}$$

The outputs of layer 3 provide the normalized firing strength, which is computed by the following equation

$$O_i^{L-3} = \bar{w}_i = \frac{w_i}{\sum_i w_i}, \; i = 1,2. \tag{21}$$

Each node functions adaptively in layer 4, with modified parameters π_i, w_i, I_{f_i}. The node functions at layer 4 are calculated as

$$O_i^{L\_4} = \frac{\sum_i \overline{w}_i \cdot I_{f_i}}{\sum_i \overline{w}_i}, \; i = 1,2. \tag{22}$$

The output layer is the final stage, where the overall result is determined. It calculates the outcome by summing all incoming signals, each of which is linearly associated with the consequent parameters, as shown below

$$O_i^{L\_5} = I_f = \sum_i \overline{w}_i \cdot I_{f_i}, \; i = 1,2. \tag{23}$$

The ANFIS controller is characterized by the input variables $\Delta\omega_r^e$ and $\Delta\omega_r^{de}$, and its output, which represents the peak value of the reference current I_{q_ref}. The training of ANFIS utilizes the back-propagation algorithm. This training process consists of four main steps:

1) loading the training data from the model to be learned;

2) transform the data into fuzzy sets by defining appropriate membership functions (MFs) and establishing the "if-then" rules;

3) update the parameters using the training algorithm;

4) convert fuzzy values into numerical values and save the model.

This paper improves speed control by applying the ANFIS system. Fuzzy logic parameters are used to train ANFIS parameters. Membership function (MF) parameters are fine-tuned through a learning process using a mixture algorithm commonly used in neural network training. The architecture of ANFIS consists of multiple layers in the ANN, as illustrated in Fig. 8, *a*. The summary parameters of the ANFIS model are shown in Table 1. Fig. 8, *b* shows the rule viewer according to the input data. This figure displays 30 of the fuzzy rules used in the ANFIS controller. In this study, a total of 100 fuzzy rules were generated based on the input-output relationships in the ANFIS model. These rules are developed based on the mathematical expression in equation (18), which defines the relationship between the inputs and outputs of the control system. They are designed to capture the nonlinear characteristics of the system and enable the ANFIS controller to handle complex control scenarios. Fig. 8, *c* represents the fuzzy rule space in the ANFIS, displayed as a 3D surface plot.

Fig. 8 illustrates the operation of the ANFIS model, demonstrating how it processes input data through multiple layers, applies fuzzy rules, and generates output. The rule viewer and fuzzy rule Space offer insights into the influence of input variables on the output, aiding in a better understanding of the system's behavior and facilitating its performance optimization.

The computational process is performed in four steps. First, the training data is loaded, and then this data is converted into a fuzzy set. IF-THEN rules are generated to correspond to the weighted fuzzy input. Then, an error-oriented training method is used to update the system parameters, fine-tuning the controller's performance. Finally, the defuzzification process converts the fuzzy data into numerical values, generating the final model, which is saved for future use and subsequent applications.

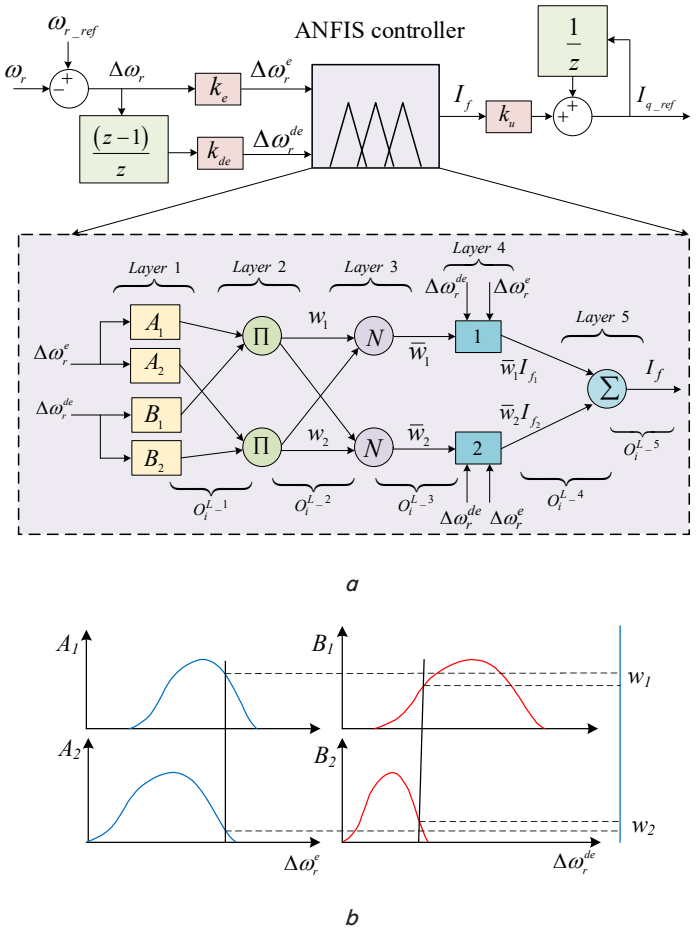
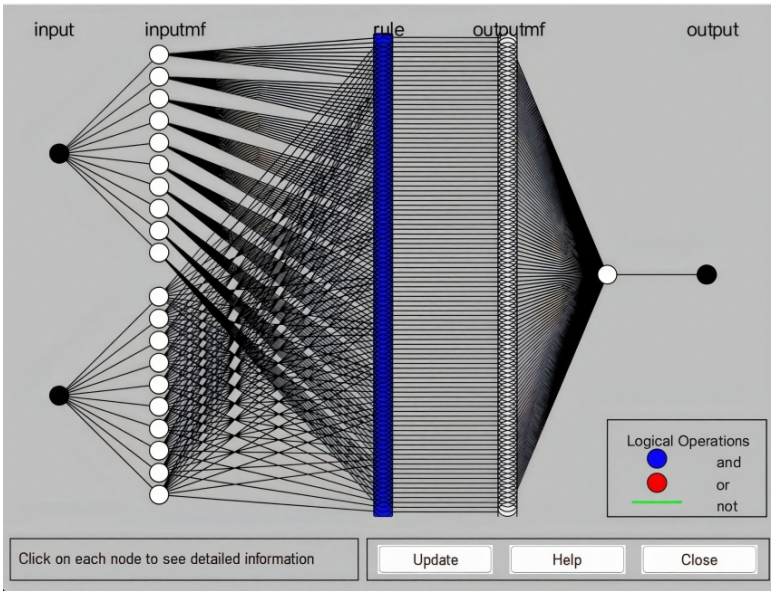
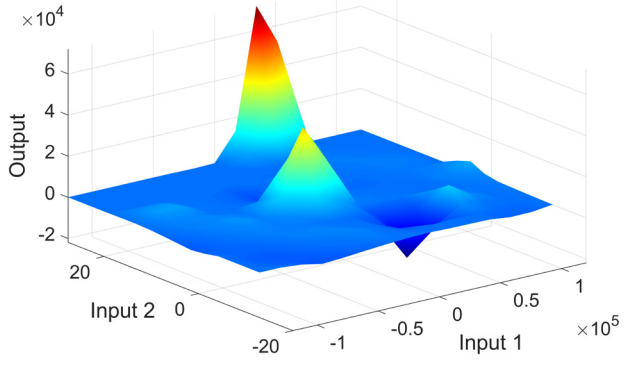


Fig. 7. The architecture of the proposed adaptive neuro-fuzzy inference system: *a* — adaptive network; *b* — fuzzy inference system



a

b



c

Fig. 8. The surface view of adaptive neuro-fuzzy inference system:
a — layer interconnections; *b* — ruler viewer; *c* — fuzzy logic rule surface

Table 2

Parameters of the adaptive neuro-fuzzy inference system controller for the power factor correction system

Parameters	Type/values
Fuzzy structure	Takagi-Sugeno
Input layer	2
Output layer	1
Input MF number	[10 10]
Rules	100
Optimization method	Hybrid algorithm
Training epochs	1000

5. 2. Results of the dynamic response under various operating conditions

5. 2. 1. No-load starting

The aim of this scenario is to evaluate the motor's performance using three different control methods during no-load startup conditions and when reaching the rated speed of 314 rad/s (3000 rpm). The starting characteristics of the PMSM drive system under no-load conditions and its performance when reaching the rated speed are depicted in Fig. 9.

In Fig. 9, *a*, the speed variation during no-load startup is shown. When using the PI and ANN controllers, over-regulation and oscillations in speed are observed during the starting process. However, the ANFIS controller regulates the speed smoothly without over-regulation, resulting in a quicker and more stable startup.

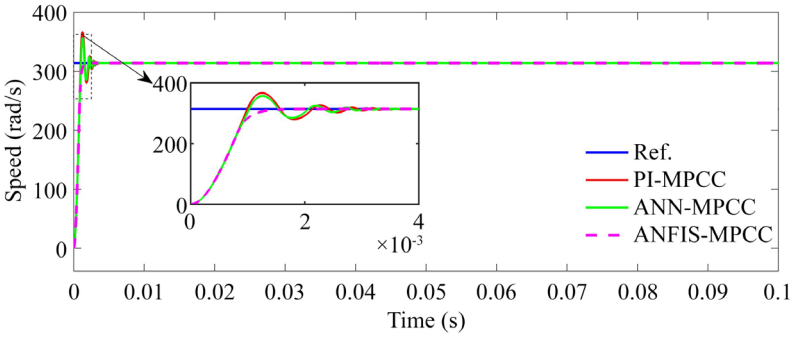
Fig. 9, *b* illustrates the torque variation during startup. The torque stabilizes once the motor reaches its rated speed. With the ANFIS controller, the torque remains stable without significant fluctuations, while both the PI and ANN controllers experience notable fluctuations at the beginning and end of the startup process.

5. 2. 2. Speed reversal

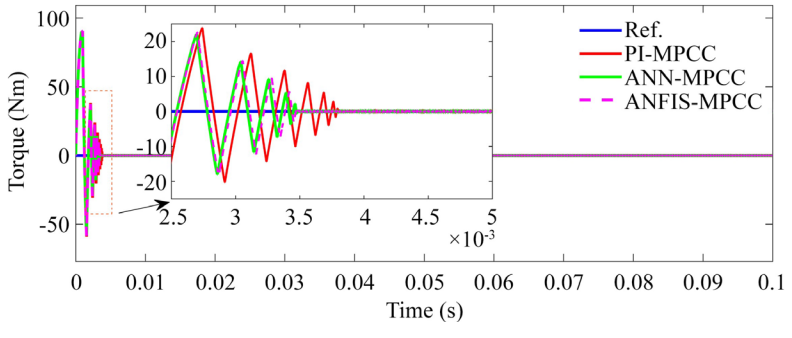
The objective of this scenario is to evaluate the performance of the three control methods under dynamic conditions. The test is performed when the motor runs in the forward direction with a steady speed of 314 rad/s, and at 0.5 seconds, the motor undergoes a speed reversal to operate at the same speed in the opposite direction. The obtained results are summarized in Fig. 10.

Fig. 10, *a* shows the change in rotor speed when the speed reversal occurs at 0.5 seconds. The ANFIS-MPCC controller demonstrates the fastest and most accurate speed stabilization compared to the other methods. The PI-MPCC and ANN-MPCC controllers show higher oscillations during the speed reversal.

In Fig. 10, *b*, the torque response to the control methods is also observed. All three methods generate some oscillations in torque at the time of speed reversal, but ANFIS-MPCC demonstrates more stability with fewer oscillations and stable torque after the reversal.

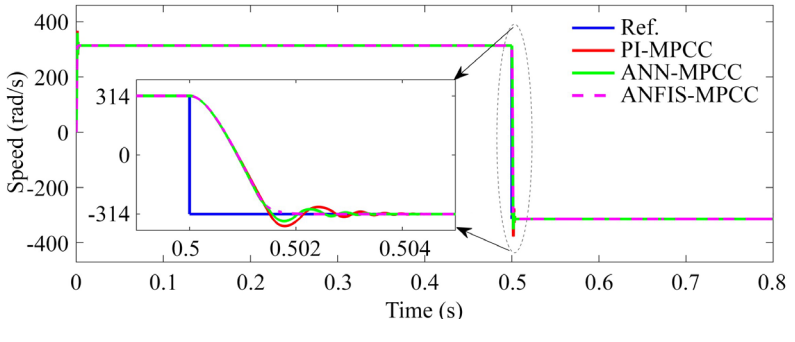


a

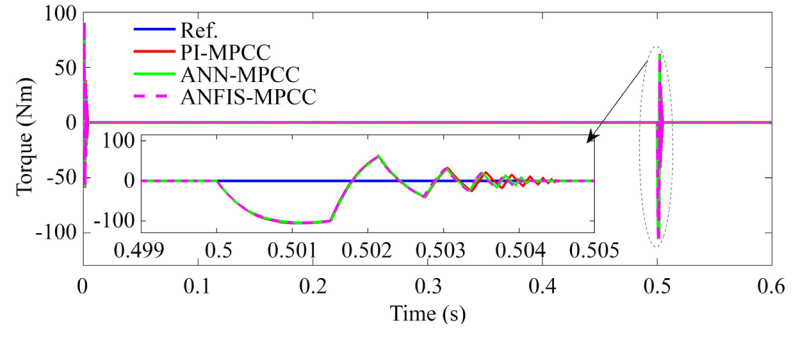


b

Fig. 9. The starting results conditions no-load at rated speed: *a* — Rotor speed response; *b* — Torque response



a



b

Fig. 10. The speed reversal results in no-load: *a* — rotor speed response; *b* — torque response

5. 2. 3. Sudden load changes at rated speed

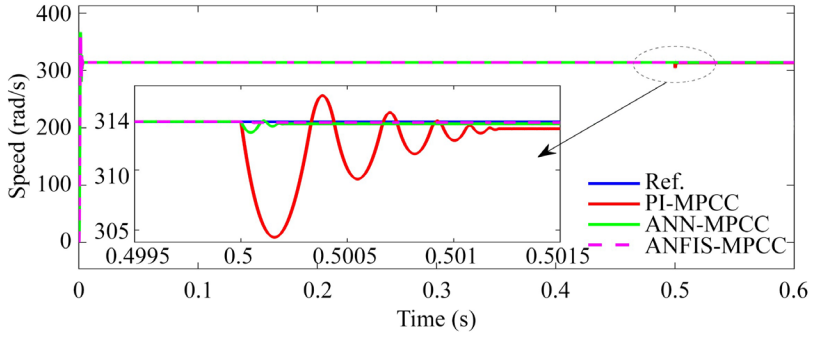
The scenario examines the dynamic response of a PMSM when subjected to a sudden increase in load torque. The sudden load change rapidly affects the motor's torque and speed, potentially leading to deceleration if the required torque exceeds the motor's controllability. The controller must respond promptly to adjust the torque and speed, ensuring motor stability.

In this study, the motor is assumed to operate in no-load mode at a rated speed. At 0.5 seconds, a rated load of 25 Nm is suddenly applied. The dynamic response results of the PMSM are presented in Fig. 11.

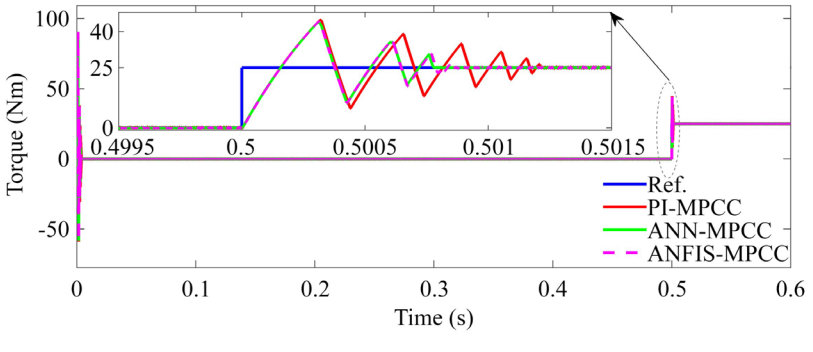
The rotor speed response during the sudden load increase is depicted in Fig. 11, *a*, which shows that the speed exhibits oscillations before stabilizing. Notably, the PI controller demonstrates significant overshoot and slower stabilization compared to other methods. In contrast, the ANN and ANFIS controllers provide smoother speed regulation with less oscillation.

Furthermore, the torque response under the same conditions is illustrated in Fig. 11, *b*, where similar oscillations are observed when the load is applied. Specifically, the PI controller shows pronounced torque oscillations, while the ANN and ANFIS controllers yield more stable torque responses with fewer fluctuations.

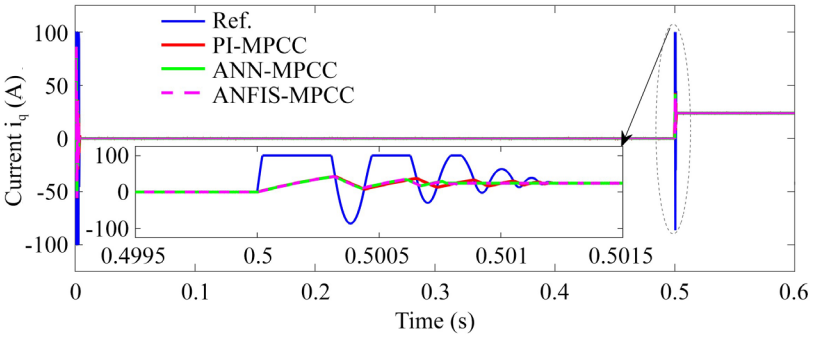
In addition to the speed and torque responses, the current response of the motor is also affected by the sudden load change, as shown in Fig. 11, *c*. A sharp increase in *q*-axis current is observed when the load is suddenly applied, particularly with the PI controller, indicating excessive current demand. In contrast, the current response is smoother with the ANN and ANFIS controllers, demonstrating their superior ability to handle load changes.



a



b



c

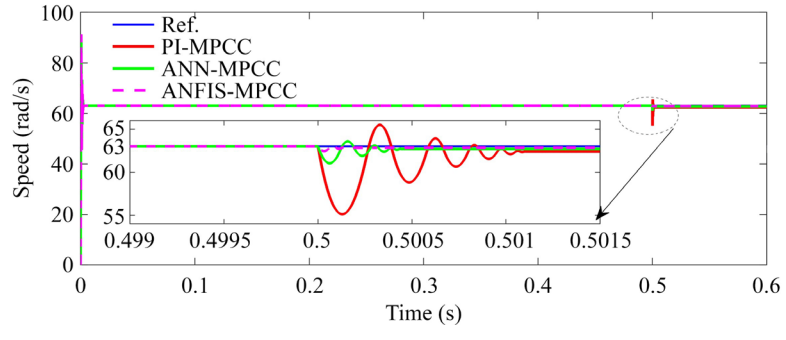
Fig. 11. The sudden change results of load at rated speed: *a* — rotor speed response; *b* — torque response; *c* — *q*-axis current response

5. 2. 4. Sudden load changes at slow speed

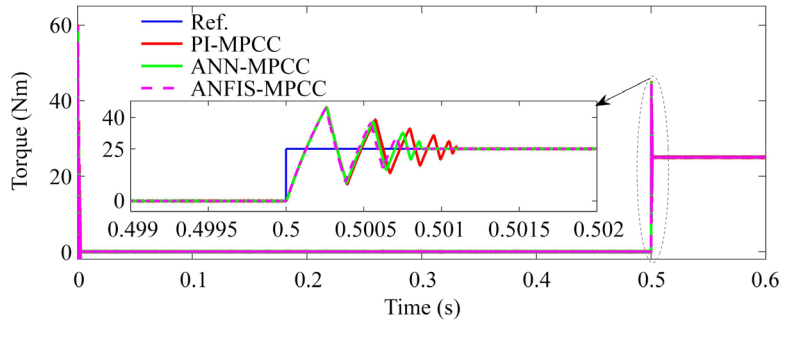
In this section, the practical scenario of the PMSM drive system operating at low speed with frequent reversals of rated load torque is examined. In order to evaluate the proposed method in this study, it is assumed that the PMSM operates at 20% of the rated speed, and at 0.5 s, a rated load is suddenly applied. The dynamic response results of the motor under these conditions are shown in Fig. 12.

As seen in Fig. 12, *a*, the rotor speed exhibits oscillations immediately after the sudden load application, initially decreasing before stabilizing gradually. The PI controller shows stronger oscillations and a longer settling time, while the ANN and ANFIS controllers provide smoother speed regulation with fewer oscillations.

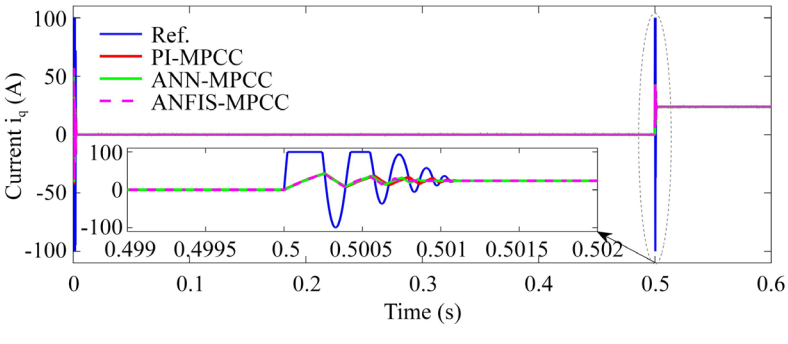
Similarly, Fig. 12, *b* illustrates the torque response, which also shows oscillations after the sudden load is applied. The PI controller presents more pronounced torque fluctuations, while both ANN and ANFIS controllers maintain more stable torque responses with smaller deviations.



a



b



c

Fig. 12. The sudden change results at the slow speed: *a* — rotor speed response; *b* — torque response; *c* — *q*-axis current response

Fig. 12, *c* shows the *q*-axis current response, which sharply increases when the load is applied, particularly with the PI controller, indicating a high demand for current. In contrast, the current response with the ANN and ANFIS controllers is smoother, demonstrating superior control capability during load changes and ensuring more effective control.

5. 2. 5. Load reversal from rated load to zero

This section simulates the effect of load reversal. The speed is held at 314 rad/s, and the load torque is 25 Nm. At 0.5 s, the load is reversed to zero. The simulation results of the motor under this condition are shown in Fig. 13.

When the load is reversed abruptly, the rotor speed exhibits oscillations, initially decreasing and then gradually stabilizing. This is illustrated in Fig. 13, *a*, where the PI controller demonstrates stronger oscillations and longer settling times compared to the ANN and ANFIS controllers, which provide smoother regulation with fewer oscillations. Therefore, based on the results obtained from the simulation in Fig. 9–13, *a*, the speed response of the three control methods is summarized in Table 3. Observing this table, the high transient performance indicators provided by the ANFIS-MPCC strategy, such as low overshoot and fast settling time, are particularly useful for control objects where high dynamic performance is critical. Such applications include industrial motor control systems, electric vehicles, robotics, and other systems that require precise control during rapid load and speed variations. These systems often need quick stabilization and the ability to minimize overshoot, ensuring smooth operation and optimal performance in dynamic conditions. For example, in electric vehicle motor control, the ANFIS-MPCC strategy's ability to quickly stabilize the motor speed and minimize overshoot is crucial for both energy efficiency and safety. Therefore, this strategy is especially beneficial in scenarios with high demands for stability and precision under rapidly changing conditions.

Table 3

The rotor speed performance comparison of control methods is based on the overshoot and settling time, considering the other scenarios

Study cases (Subsections)	Control methods		
	PI-MPCC	ANN-MPCC	ANFIS-MPCC
Overshoot (%)			
Fig. 9, *a*	16.819	13.396	0.015
Fig. 10, *a*	12.484	3.026	0.861
Fig. 11, *a*	3.043	0.276	0.067
Fig. 12, *a*	12.484	3.026	0.861
Fig. 13, *a*	1.620	0.128	0.014
Settling time (s)			
Fig. 9, *a*	0.00336	0.00304	0.00147
Fig. 10, *a*	0.504007	0.50358	0.00147
Fig. 11, *a*	0.067	0.50021	0.50009
Fig. 12, *a*	0.50106	0.50039	0.50008
Fig. 13, *a*	0.50102	0.50020	0.50001

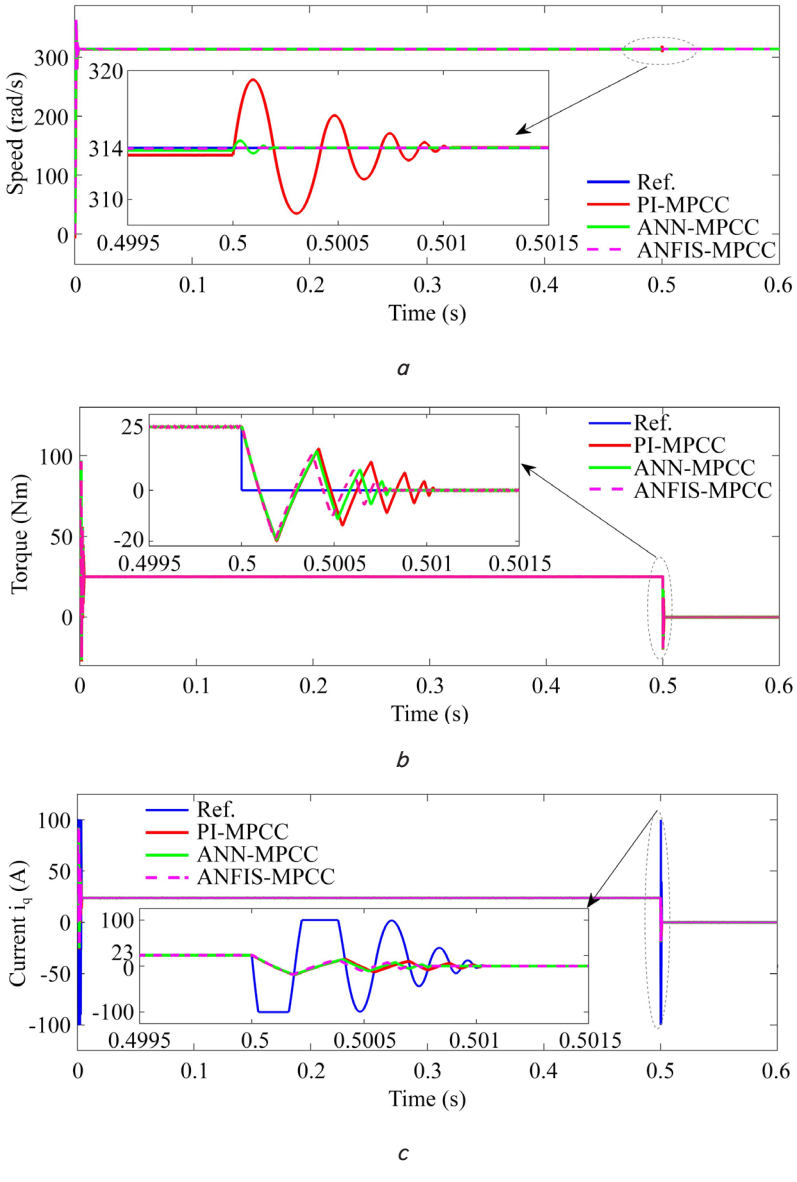


Fig. 13. The sudden load reversal results from the rated load to zero at the slow speed: *a* — rotor speed response; *b* — torque response; *c* — *q*-axis current response

Similarly, the torque response exhibits oscillations after the load reversal, as depicted in Fig. 13, *b*. The PI controller shows more pronounced torque fluctuations, whereas the ANN and ANFIS controllers ensure a more stable torque output with smaller deviations.

Furthermore, the *q*-axis current response, presented in Fig. 13, *c*, reveals a sharp increase in current when the load is reversed, particularly with the PI controller. In contrast, the ANN and ANFIS controllers exhibit a smoother current response, indicating better controllability and control performance during load changes.

5. 3. THD results of the stator current

5. 3. 1. When operating with the rated load at both low speed

This study evaluates the performance of three control methods by analyzing the THD in the stator current under different operating conditions. The THD results during rated load conditions at both rated speed (314 rad/s) and low speed (20% of rated speed) are presented in Fig. 14, 15, respectively.

At rated speed, as observed in Fig. 14, *b*, the PI controller exhibits a THD of 2.57%, with noticeable oscillations in the stator current waveform, indicating a relatively high level of harmonic distortion. In contrast, the ANN controller achieves a lower THD of 2.11%, as shown in Fig. 14, *d*, with smoother transitions in the waveform, reflecting better control of the current. The THD reduces to 2.0% when applying the ANFIS controller, as shown in Fig. 14, *f*. It can be seen that the smoothest waveform and the lowest harmonic distortion are achieved, which confirms its superior efficiency in minimizing harmonics compared to the other two methods.

When operating at low speed, as illustrated in Fig. 15, the THD results follow a similar pattern. The PI controller's THD remains high at 2.49%, with more pronounced oscillations, as shown in Fig. 15, *c*, while using the ANN controller, THD reduces to 2.1%, which is illustrated in Fig, *e*. The ANFIS controller continues to outperform the others, with the cleanest current waveform, demonstrating its superior capability in handling dynamic conditions. From Fig. 15, *f*, the THD achieves 2.02%.

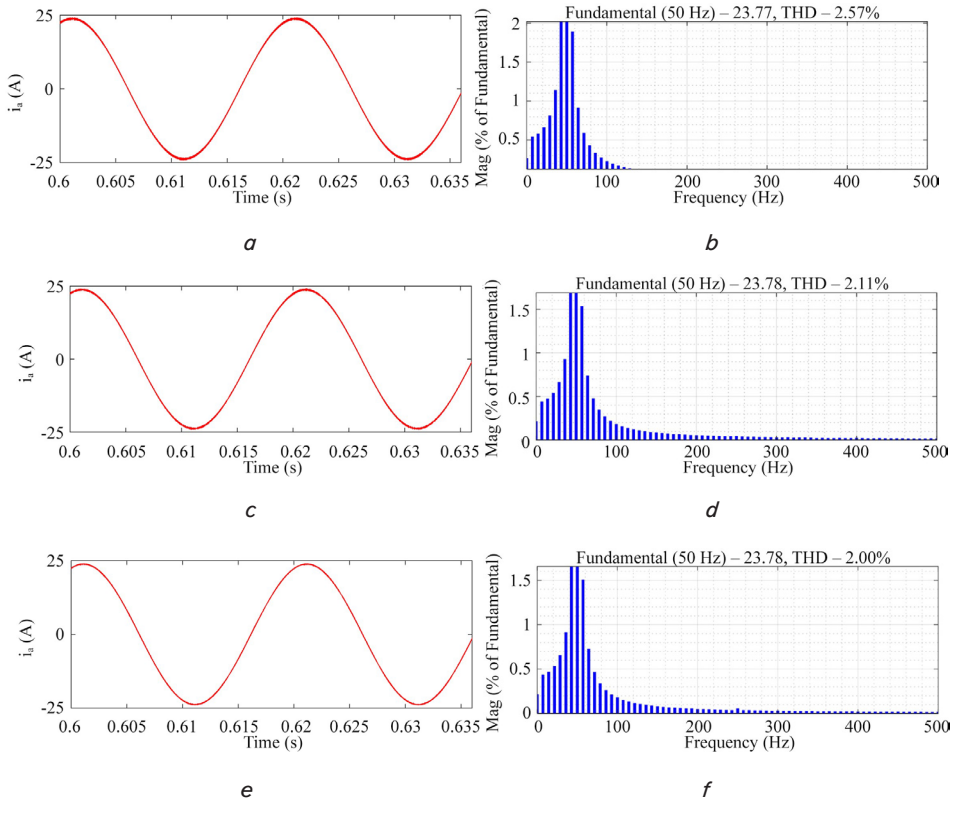


Fig. 14. The total harmonic distortion results in the phase A stator current during the rated load operation at rated speed: *a, b* — using proportional-integral controller; *c, d* — using artificial neural network controller; *e, f* — using adaptive neuro-fuzzy inference system controller

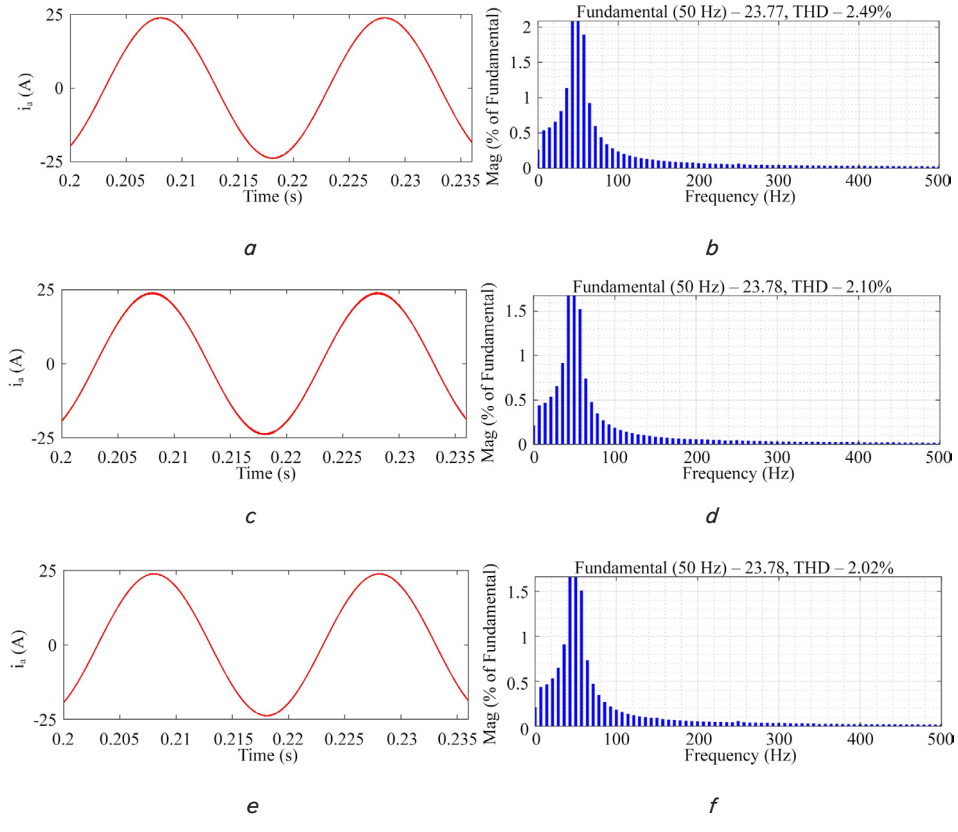


Fig. 15. The total harmonic distortion in the phase A stator current during the rated-load operation at slow speed (20% rated speed): *a, b* — using proportional-integral controller; *c, d* — using artificial neural network controller; *e, f* — using adaptive neuro-fuzzy inference system controller

These results indicate that while the PI controller is effective for speed regulation, it struggles to minimize harmonic distortion, particularly under low-speed conditions. The ANN controller offers some improvement, but the ANFIS controller provides the best performance in terms of both speed regulation and harmonic distortion reduction, making it the optimal choice for high-efficiency PMSM drive systems.

6. Discussion of the study results on the dynamic response and control performance of a permanent magnet synchronous motor using three control methods

This study discusses the performance and significance of the simulated MPCC technique for the inner loop controller, which incorporates PI, ANN, and ANFIS for the outer loop controllers. The effectiveness of these control approaches is validated through MATLAB-Simulink simulations for a PMSM model, examining key responses under both variable-speed operation with fixed load torque and sudden full-load torque conditions.

The results obtained by applying the ANFIS-MPCC from scenarios are compared with those of the PI-MPCC and ANN-MPCC. This comparative study focuses on key performance metrics such as dynamic response and control performance, including overshoot, settling time, and THD.

The overshoot of the rotor speed response of ANFIS-MPCC is 0.015%, which is significantly lower than the PI-MPCC (16.819%) and ANN-MPCC (13.396%) methods, as illustrated in Fig. 9, *a*. Moreover, the settling time of the rotor speed response of ANFIS-MPCC is also the fastest at 0.00147 seconds, outperforming PI-MPCC (0.00336 seconds) and ANN-MPCC (0.00304 seconds). Notably, while avoiding overshoot is crucial in optimizing transient processes, a small amount of overshoot can be acceptable in many practical situations, particularly in systems that require fast response and flexible control. In such cases, a small overshoot can be tolerated as long as other critical factors, such as settling time and THD, are improved and maintained within acceptable limits. A summary of the results from other cases is presented in Table 3, which further demonstrates that ANFIS-MPCC exhibits a superior ability to quickly stabilize the motor's speed, minimize oscillations, and enhance the system's dynamic response.

ANFIS-MPCC maintains torque consistency during load changes and speed reversals, which can be observed in Fig. 9–13, *b* with less fluctuation compared to PI-MPCC and ANN-MPCC. These results highlight ANFIS-MPCC's superior ability to handle rapid load changes and speed reversals, ensuring smooth motor operation

In simulations involving sudden load changes, ANFIS-MPCC adjusts the q-axis current smoothly, avoiding sharp increases. This helps minimize energy loss and ensures stable motor performance. In contrast, PI-MPCC shows a sharp increase in *q*-axis current during load changes, leading to unnecessary energy consumption. This can be demonstrated from the results in Fig. 10–13, *b*.

ANFIS-MPCC shows a significant reduction in THD, with values of 2.0% at rated speed and 2.02% at low speed, compared to PI-MPCC (2.57% and 2.49 %) and ANN-MPCC (2.11%). This demonstrates ANFIS-MPCC's effectiveness in reducing harmonic distortion and improving energy efficiency, which is important for motor longevity and performance.

One of the key factors contributing to ANFIS-MPCC's superior performance is its ability to handle nonlinear calculations through fuzzy logic. The fuzzification formula in ANFIS calculates the degree of error and change in error, allowing the control system to respond more flexibly to rapid load changes. This is achieved through (19), which calculates the degree of errors. As a result, ANFIS-MPCC can make more accurate and smoother control decisions, thereby minimizing overshoot and enabling the system to stabilize faster.

Furthermore, ANFIS-MPCC uses fuzzy rules to determine the degree of influence of input factors, as shown in (20). This equation represents the activation level of each fuzzy rule, allowing the system to adjust control parameters optimally. This is particularly important when the load changes suddenly, helping to reduce overshoot and quickly return the motor to a stable state. (21), (22) also play a significant role in standardizing the activation levels of the rules, helping the control system maintain stability and respond quickly throughout the load change process.

Notably, ANFIS-MPCC's use of (22) enables the control system to make more accurate decisions about the control signal, contributing to reduced overshoot and shorter settling times. Overall, the results demonstrate the effectiveness of ANFIS-MPCC in controlling PMSM motors under dynamic conditions.

When comparing ANFIS-MPCC with the MV-FCS-MPC method from [9], it is clear that both methods incorporate fuzzy logic to improve dynamic performance. However, the MV-FCS-MPC method offers a more direct approach in handling computational efficiency and steady-state performance. In contrast, ANFIS-MPCC excels in managing nonlinearities and adapting to rapid load changes, making it a better fit for applications that prioritize dynamic adaptability.

In comparison with the FCS-MPCC method from [5], both ANFIS-MPCC and FCS-MPCC aim to optimize the performance of PMSM control. However, each method has its own strengths. ANFIS-MPCC excels in handling nonlinearities and adapting quickly to load changes, thanks to the combination of fuzzy logic and neural networks, which helps reduce overshoot and achieve fast settling times. Nevertheless, this method requires more complex computations.

Compared to the TV-MFPC in [4], the ANFIS-MPCC method has significant advantages in terms of fast response and flexibility in adapting to sudden load changes and speed reversals. ANFIS-MPCC achieves very low overshoot (0.015%) and extremely fast settling time (0.00147 s), allowing the system to operate stably and efficiently in applications that require strong dynamic response, such as electric vehicles. In contrast, TV-MFPC in [4], while more stable and reducing current ripple, is less quick and adaptable when handling rapid load changes compared to ANFIS-MPCC. Although ANFIS-MPCC demonstrates the ability to quickly stabilize rotor speed and achieve an extremely fast settling time (0.00147 s), the practical implementation of this strategy may face challenges. Achieving such a fast settling time requires high hardware demands, including fast processing capabilities and real-time response, especially since the ANFIS model combined with MPCC requires significant computational resources. Implementing this in a real-world system would require powerful processors and real-time control systems, which could introduce challenges regarding cost and processing capability in industrial applications.

Table 4 presents a detailed comparison of the THD values for three control methods under varying conditions of

load and speed reversal. The results clearly demonstrate that ANFIS-MPCC offers the best performance in terms of reducing THD, with values of 0.88% at rated speed and 0.90% at low speed, outperforming both PI-MPCC and ANN-MPCC. This is a critical performance metric as it indicates ANFIS-MPCC's ability to maintain smooth operation and lower harmonic distortion, which is essential for improving system stability and efficiency in dynamic environments.

Compared to the FCS-MPCC method in [5], which also reduces THD, ANFIS-MPCC shows better performance in dynamic environments. While FCS-MPCC excels in computational efficiency and steady-state control, it struggles with rapid load changes and nonlinear behavior, which ANFIS-MPCC handles more effectively.

One of the main challenges is the computational complexity of the ANFIS-MPCC method, which needs to be optimized for real-time applications. Its current complexity limits its use in industrial environments with constrained computational resources. Future research should focus on improving the computational efficiency of the ANFIS-MPCC method to make it more suitable for practical implementation, particularly in applications that require rapid response and stability, such as electric vehicles and industrial control systems.

Although the ANFIS-MPCC method provides high performance in reducing overshoot and settling time, one of the key challenges is the computational complexity arising from the combination of fuzzy logic and neural networks. This computational process may face difficulties when implemented in real-time applications, especially those requiring fast and accurate responses. Therefore, to enhance the performance and applicability of ANFIS-MPCC, research and optimization of computational techniques are necessary to reduce computational complexity without compromising control performance. This improvement will enable the method to better meet the real-time demands of modern motor control systems.

7. Conclusions

1. The ANFIS-MPCC control strategy has been successfully developed by integrating MPCC with ANFIS. This integration has significantly improved the dynamic performance of PMSM, as evidenced by the reduction in key indicators such as overshoot, settling time, and THD. The ANFIS-MPCC method has demonstrated superior performance, reducing overshoot and settling time, outperforming traditional methods like PI-MPCC and ANN-MPCC. Specifically, the overshoot is reduced to 0.015%, settling time to 0.00147 seconds, and THD to 2.0% at rated speed and 2.02% at low speed.

2. The dynamic response of PMSM has been thoroughly evaluated under various operating conditions, including no-load starting, speed reversal, sudden load changes, and load reversal from rated load to zero. The results confirm that ANFIS-MPCC provides the fastest and most stable response, with overshoot reduced to 0.015%, settling time to 0.00147 seconds, and speed stabilization achieved in a very short time. The method demonstrated its ability to quickly minimize overshoot and stabilize speed and torque, outperforming PI-MPCC and ANN-MPCC under all test conditions.

3. The ANFIS-MPCC method has successfully reduced THD in the stator current at both rated and low speeds. At rated speed, THD has been reduced to 2.0%, and at low speed, it is reduced to 2.02%. This significant reduction in THD improves the energy efficiency of PMSM, reducing losses and enhancing motor performance under various operating conditions, which is a notable improvement over PI-MPCC and ANN-MPCC.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this study.

Financing

The study was performed without financial support.

Data availability

The manuscript has no associated data.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Fu, R., Cao, Y. (2021). Hybrid flux predictor-based predictive flux control of permanent magnet synchronous motor drives. IET Electric Power Applications, 16 (4), 472–482. https://doi.org/10.1049/elp2.12168
2. Hu, J., Fu, Z., Xu, R., Jin, T., Feng, Z., Wang, S. (2024). Low-Complexity Model Predictive Control for Series-Winding PMSM with Extended Voltage Vectors. Electronics, 14 (1), 127. https://doi.org/10.3390/electronics14010127
3. Pang, S., Zhang, Y., Huangfu, Y., Li, X., Tan, B., Li, P. et al. (2024). A Virtual MPC-Based Artificial Neural Network Controller for PMSM Drives in Aircraft Electric Propulsion System. IEEE Transactions on Industry Applications, 60 (2), 3603–3612. https://doi.org/10.1109/tia.2023.3338605
4. Wen, D., Zhang, Y., Zhang, Y. (2023). Three-vector model-free predictive control for permanent magnet synchronous motor. IET Power Electronics, 16 (16), 2754–2768. https://doi.org/10.1049/pel2.12599
5. Wang, H., Wu, X., Zheng, X., Yuan, X. (2023). Model Predictive Current Control of Nine-Phase Open-End Winding PMSMs With an Online Virtual Vector Synthesis Strategy. IEEE Transactions on Industrial Electronics, 70 (3), 2199–2208. https://doi.org/10.1109/tie.2022.3174241

6. Sangar, B., Singh, M., Sreejeth, M. (2024). An improved ANFIS model predictive current control approach for minimizing torque and current ripples in PMSM-driven electric vehicle. Electrical Engineering, 106 (5), 5897–5907. https://doi.org/10.1007/s00202-024-02346-3
7. Alitasb, G. K. (2024). Integer PI, fractional PI and fractional PI data trained ANFIS speed controllers for indirect field oriented control of induction motor. Heliyon, 10 (18), e37822. https://doi.org/10.1016/j.heliyon.2024.e37822
8. Rodriguez, C. A., Ponce, P., Molina, A. (2016). ANFIS and MPC controllers for a reconfigurable lower limb exoskeleton. Soft Computing, 21 (3), 571–584. https://doi.org/10.1007/s00500-016-2321-9
9. Bouguenna, I. F., Tahour, A., Kennel, R., Abdelrahem, M. (2021). Multiple-Vector Model Predictive Control with Fuzzy Logic for PMSM Electric Drive Systems. Energies, 14 (6), 1727. https://doi.org/10.3390/en14061727
10. Dianov, A., Tinazzi, F., Calligaro, S., Bolognani, S. (2022). Review and Classification of MTPA Control Algorithms for Synchronous Motors. IEEE Transactions on Power Electronics, 37 (4), 3990–4007. https://doi.org/10.1109/tpel.2021.3123062
11. Gade, C. R., W, R. S. (2022). Control of Permanent Magnet Synchronous Motor Using MPC–MTPA Control for Deployment in Electric Tractor. Sustainability, 14 (19), 12428. https://doi.org/10.3390/su141912428
12. Yu, H., Wang, J., Xin, Z. (2022). Model Predictive Control for PMSM Based on Discrete Space Vector Modulation with RLS Parameter Identification. Energies, 15 (11), 4041. https://doi.org/10.3390/en15114041
13. Dai, L., Tung, D. (2017). Modeling for Development of Simulation Tool: A Case Study of Grid Connected Doubly Fed Induction Generator Based on Wind Energy Conversion System. International Journal of Applied Engineering Research, 12 (11), 2981–2996.
14. Utomo, W. M., Muhammad Zin, N., Haron, Z. A., Sim, S. Y., Bohari, A. A. et al. (2014). Speed Tracking of Field Oriented Control Permanent Magnet Synchronous Motor Using Neural Network. International Journal of Power Electronics and Drive Systems (IJPEDS), 4 (3). https://doi.org/10.11591/ijpeds.v4i3.5941
15. Jang, J.-S. R. (1993). ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics, 23 (3), 665–685. https://doi.org/10.1109/21.256541