(Received 7 January 2025, Accepted 24 February 2025, Scheduled 6 March 2025)

Speed Sensorless Control of Magnet Assisted Bearingless Synchronous Reluctance Motor Based on Improved BP Neural Network

Chao Chen and Huangqiu Zhu*

School of Electrical and Information Engineering, Jiangsu University, Zhenjiang 212013, China

ABSTRACT: In order to solve the problems of susceptibility to environmental disturbances, complex installation, and low reliability caused by the photoelectric code disks in the permanent magnet assisted bearingless synchronous reluctance motor (PMa-BSynRM) system, an improved BP neural network (BPNN) left-inverse system is proposed to establish the speed self-sensing system of the PMa-BSynRM. Firstly, the mathematical model of the PMa-BSynRM is established, and the left reversibility of the PMa-BSynRM speed subsystem is analyzed. Secondly, the traditional BP neural network is optimized from four aspects, including the number of neurons in the hidden layer, initial weight, connection weight, and learning rate. Then, the improved BPNN model is used as the inverse model of the left inverse system to fit the speed subsystem of the PMa-BSynRM. Finally, the simulation results show that accurate estimation of the speed and rotor position is achieved by the proposed speed self-detection system, and the accuracy and feasibility of the proposed speed self-detection system are validated by the experimental results.

1. INTRODUCTION

The bearingless synchronous reluctance motor (BSynRM) adds excitation windings to the stator slot of the traditional synchronous reluctance motor, thus generating levitation force and realizing contactless operation. This design is characterized by a simple structure, no friction, no lubrication, high precision, high critical speed, and long service life [1]. However, at high-speed operations, the rotor in the BSynRM lacks excitation, which results in lower torque density and power factor. In order to address the existing issues, a permanent magnet assisted bearingless synchronous reluctance motor (PMa-BSynRM) has been designed, which replaces the conventional salient pole rotor with a permanent magnet-assisted rotor. The numerous advantages of bearingless motors, permanent magnet motors, and magnetic bearings are integrated in the PMa-BSynRM, which holds significant potential for widespread application in ultra-high cleanliness environments across industries such as chemistry, pharmaceuticals, biomedical sciences, and semiconductor manufacturing [2–4].

The precise detection of the speed and rotor position is crucial for the stable operation of a PMa-BSynRM. At present, photoelectric code disks are generally used to detect the rotor speed and rotor position in the research applications available. However, due to low reliability, susceptibility to environmental disturbances influences on detection accuracy, and increased system complexity, mechanical sensors are not suitable for high-speed and high-precision working environments. Sensorless speed self-detection system has emerged.

The current speed self-sensing methods can be divided into the following categories. One is based on motor model estimation methods, which extract speed signals from physical quantities related to the rotational speed, including model reference adaptive system (MRAS) [5, 6], Kalman filter [7–9], and sliding mode observer (SMO) [10, 11]. The above methods are applicable to speed detection during high-speed operation of the motor, but those methods require an accurate mathematical model of the controlled object. One is the high-frequency signal injection method based on the convex pole effect of the motor [12, 13]. In [13], a pseudo-random high-frequency triangular-wave current signal injection based sensorless control scheme for synchronous reluctance motor drives at zero-low speed domain is utilized to suppress the high-frequency (HF) loss and HF audible noise. The high-frequency signal injection method does not rely on motor parameters and is applicable to zero/low-speed control, but it requires additional circuits to process high frequency signals. With the development of intelligent algorithms, many mature intelligent algorithms have been applied to speed estimation, and neural network algorithm has also been applied to the field of speed estimation because of its excellent approximation performance. In [14], a neural network observer based sensorless speed control strategy is proposed for the permanent magnet synchronous motor, and the back electromotive force is estimated by current and current error, thus realizing speed sensorless control. In [15], a neural network model reference adaptive system speed observer suited for linear induction motor drives is proposed to compute online, in recursive form, the machine linear speed. Although the neural network algorithm can obtain extremely high accuracy

* Corresponding author: Huangqiu Zhu (zhuhuangqiu@ujs.edu.cn).

in speed estimation, does not need an accurate mathematical model, and possesses excellent nonlinear fitting capabilities, it is susceptible to overfitting and easily gets trapped in local optima.

In this paper, an improved represent back propagation (BP) neural network (IBPNN) left-inverse system is adopted to estimate the speed of the PMa-BSynRM, addressing the issues inherent in neural networks mentioned earlier. Firstly, the normalized data is decomposed based on the singular value decomposition, and the number of hidden nodes is determined according to main eigenvalues, thus determining the network structure. Secondly, uniformly distributed random numbers are used to initialize the weights of BPNN and ensure that the sum of the input weights for each neuron is 1, thus avoiding excessive initial weights and preventing the activation function from entering the saturation zone. Finally, by introducing momentum back propagation and adaptive learning rate into BPNN, the possibility of the local optima can be reduced. The simulated and experimental results show that the method proposed in this paper has good tracking performance, control accuracy, and suspension force performance.

2. OPERATION PRINCIPLE AND MATHEMATICAL MODELING

2.1. Operation Principle of the PMa-BSynRM

The research object of this paper is PMa-BSynRM. The cross section of a 24-stator-slot, 4-rotor-pole PMa-BSynRM is shown in Fig. 1. Two distinct sets of concentric windings are embedded within the stator slots, and they are the outer torque winding with two pole pairs and the inner suspension force winding with one pole pair. This design, where the pole pair counts differ by 1 between the two concentric windings, is integral to ensuring the stable levitation of the bearingless motor. The radially magnetized permanent magnets are installed in the rotor with flux barriers.

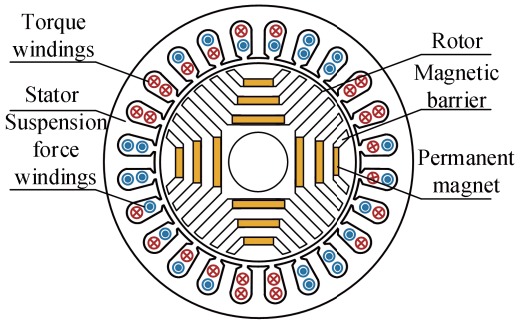

FIGURE 1. The cross section of the PMa-BSynRM.

The torque generated in the PMa-BSynRM is composed of reluctance torque and permanent magnet torque. The reluctance torque is produced in accordance with the principle that the magnetic field lines are always closed along the path of least reluctance. The air gap magnetic field generated by the auxiliary permanent magnet in PMa-BSynRM generates the tangential Lorentz force on the torque winding, thus generating the permanent magnetic torque driving the PMa-BSynRM rotation.

The principle of the suspension force generation is shown in Fig. 2. The PMa-BSynRM has 4-pole torque winding N_a and 2-pole radial suspension winding N_x. The permanent magnetic torque ψ_a is generated by the torque winding current and permanent magnets, and the reluctance torque ψ_x is generated by the suspension force winding current. Due to the interaction between ψ_x and ψ_a, the magnetic flux density at air gap 3 is increased, whereas the magnetic flux density at air gap 1 is decreased. The radial suspension force F_x is produced on the rotor in the direction opposite to the x axis due to the unbalanced magnetic flux density in the air gaps. Similarly, the radial suspension force along the y axis can be generated. By controlling the magnitude and direction of the current in the suspension force winding, a controllable unbalanced magnetic field is generated, and the stable suspension of the PMa-BSynRM is realized.

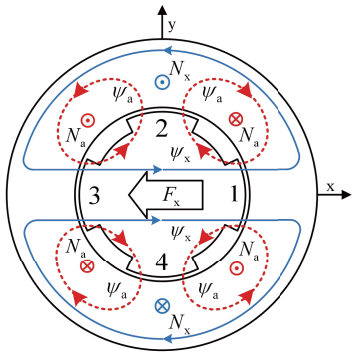

FIGURE 2. The principle of the suspension force generation.

2.2. Mathematical Model of the PMa-BSynRM

Due to the mutual inductance between two windings in PMa-BSynRM, there must be the coupling relationship between their control currents. Considering the influence of winding mutual inductance and rotor eccentric displacement on the internal flux linkage of the motor, the d- and q-axis components of the flux linkage for the two winding sets can be obtained:

$$\begin{bmatrix} \psi_{Md} \\ \psi_{Mq} \\ \psi_{Bd} \\ \psi_{Bq} \end{bmatrix} = \begin{bmatrix} L_{Md} & 0 & M_d x_d & -M_d y_q \\ 0 & L_{Mq} & M_q y_q & M_q x_d \\ M_d x_d & M_q y_q & L_B & 0 \\ -M_d y_q & M_q x_d & 0 & L_B \end{bmatrix} \cdot \begin{bmatrix} i_{Md} + i_0 \\ i_{Mq} \\ i_{Bd} \\ i_{Bq} \end{bmatrix} \tag{1}$$

where ψ_{Md} and ψ_{Mq} are the d- and q-axis equivalent air gap flux linkages of the torque windings, respectively; ψ_{Bd} and ψ_{Bq} are the d- and q-axis equivalent air gap flux linkages of the suspension force windings, respectively; L_{Md} and L_{Mq} are the d- and q-axis self-inductances of the torque windings, respectively; M_d and M_q are the suspension force constants; x_d and y_q are the d- and q-axis components of the rotor radial displacements, respectively; i_{Md} and i_{Mq} are the d- and q-axis equivalent control currents of the torque windings, respectively; i

is the equivalent excitation current of the permanent magnet which is defined in the positive direction of d-axis; L_B is the self-inductance of the suspension force windings; i_{Bd} and i_{Bq} are the d- and q-axis equivalent control currents of the suspension force windings, respectively.

The voltage equation of the PMa-BSynRM in the d- and q-axes is derived:

$$\begin{aligned}\begin{bmatrix} u_{Md} \\ u_{Mq} \\ u_{Bd} \\ u_{Bq} \end{bmatrix} &= \begin{bmatrix} R_M & 0 & 0 & 0 \\ 0 & R_M & 0 & 0 \\ 0 & 0 & R_B & 0 \\ 0 & 0 & 0 & R_B \end{bmatrix} \cdot \begin{bmatrix} i_{Md} \\ i_{Mq} \\ i_{Bd} \\ i_{Bq} \end{bmatrix} + \omega \begin{bmatrix} -\psi_{Mq} \\ \psi_{Md} \\ -\psi_{Bq} \\ \psi_{Bd} \end{bmatrix} \\ &+ \begin{bmatrix} L_{Md} & 0 & M_d x_d & -M_d y_q \\ 0 & L_{Mq} & M_q y_q & M_q x_d \\ M_d x_d & M_q y_q & L_B & 0 \\ -M_d y_q & M_q x_d & 0 & L_B \end{bmatrix} \cdot \frac{d}{dt} \begin{bmatrix} i_{Md} + i_0 \\ i_{Mq} \\ i_{Bd} \\ i_{Bq} \end{bmatrix} \end{aligned} \quad (2)$$

where u_{Md} and u_{Mq} are the d- and q-axis equivalent control voltages of the torque windings, respectively; u_{Bd} and u_{Bq} are the d- and q-axis equivalent control voltages of the suspension force windings, respectively; R_M is the equivalent torque winding resistance; R_B is the equivalent suspension winding resistance; ω is the electrical angular speed of the rotor.

The radial suspension force of the rotor is constituted by the d- and q-axis suspension force components obtained by the virtual displacement method and the unilateral magnetic attraction that is directly proportional to the eccentric displacement of the rotor, and the expression is as follows:

$$\begin{aligned}\begin{bmatrix} F_x \\ F_y \end{bmatrix} &= \begin{bmatrix} \partial W_m/\partial x_d \\ \partial W_m/\partial x_q \end{bmatrix} + \begin{bmatrix} F_{c\alpha} \\ F_{c\beta} \end{bmatrix} \\ &= \sqrt{M_d^2 (i_{Md} + i_0)^2 + M_q^2 i_{Mq}^2} \\ &\cdot \begin{bmatrix} -\cos(2\omega t + \lambda) & \sin(2\omega t + \lambda) \\ \sin(2\omega t + \lambda) & \cos(2\omega t + \lambda) \end{bmatrix} \cdot \begin{bmatrix} i_{B\alpha} \\ i_{B\beta} \end{bmatrix} \\ &+ [k_d (i_{Md} + i_0)^2 + k_q i_{Mq}^2] \cdot \begin{bmatrix} x_\alpha \\ y_\beta \end{bmatrix} \end{aligned} \quad (3)$$

where $W_m = 0.5[i_{Md} \; i_0 \; i_{Mq} \; i_{Bd} \; i_{Bq}] \cdot [\psi_{Md} \; \psi_{Mq} \; \psi_{Bd} \; \psi_{Bq}]^T$ is the magnetic co-energy; $F_{c\alpha}$ and $F_{c\beta}$ are the α- and β-axis unilateral magnetic pull forces, respectively; k_d and k_q are the eccentric displacement constants; x_α and y_β are the α- and β-axis components of the rotor radial displacements, respectively; $\lambda = \tan^{-1}[M_q(i_{Mq}i)/(M_d i_{Md})]$.

The electromagnetic torque equation of the PMa-BSynRM is further derived:

$$\begin{aligned} T_e &= (i_{Mq}\psi_{Md} - i_{Md}\psi_{Mq}) P_M \\ &= i_0 i_{Mq} L_{MMd} P_M + 0.5 (L_{Md} - L_{Mq}) \sin(2\gamma) i_M^2 P_M \\ &+ \begin{bmatrix} i_{Md} & -i_{Mq} \end{bmatrix} \cdot \begin{bmatrix} M_d x_d & -M_q y_q \\ M_d y_q & M_q x_d \end{bmatrix} \cdot \begin{bmatrix} i_{Bd} \\ i_{Bq} \end{bmatrix} \end{aligned} \quad (4)$$

where P_M is the pole number of the torque winding, γ phase angle of the torque winding current excitation, and L_{MMd} the component of the mutual inductance between the stator and rotor on the d axis.

According to the rotor dynamics theory, the motion equation of the rotor can be obtained:

$$\begin{cases} m\ddot{x} = F_x - f_x \\ m\ddot{y} = F_y - mg - f_y \\ \frac{d\omega_m}{dt} = \frac{1}{J}(T_e - T_L) \end{cases} \quad (5)$$

where m is the mass of the rotor; f_x and f_y are the interference forces applied along the x- and y-directions, respectively; g is the gravity constant; ω_m is the mechanical angular frequency of rotor; g is the inertia of the rotor motion; T_L is the load torque

3. ANALYSIS OF THE LEFT-INVERTIBILITY OF THE SPEED SUBSYSTEM

From the mathematical point of view, the definition of left inverse system is: Given a system Σ defined by the mapping $u \to y$, if there exists another system Σ_1 characterized by the mapping $v \to w$, such that under the condition that the initial value of Σ_1 satisfies Σ, whenever $v(t)$ equals $y(t)$, it also holds true that $w(t)$ equals $u(t)$. The system Σ_1 is referred to as the left inverse system of the system Σ, and the system Σ is left invertible.

General nonlinear systems can be expressed as:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad (6)$$

where $\mathbf{u} \in \mathbf{R}^p$ is the input variables of the system; $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_{um})^T \in \mathbf{R}^n$ is the state variable; $\mathbf{x}_m \in \mathbf{R}^s$ is the directly observable variable; $\mathbf{x}_{um} \in \mathbf{R}^{n-s}$ is the indirectly observable variables.

If the left-inverse system of the inner sensor system Z exists, the input variable of Z is $\mathbf{x}_{um}$, and the output variable of Z is $\mathbf{x}_m$. The output variable $\mathbf{x}_m$ can be expressed as:

$$\mathbf{x}_m = f_z(\mathbf{x}_{um}, \dot{\mathbf{x}}_m, \ddot{\mathbf{x}}_m, \ldots, \mathbf{u}) \quad (7)$$

If the left inverse system of Z exists, the left inverse system can be expressed as:

$$\mathbf{x}_{um} = \bar{f}(\mathbf{x}_m, \dot{\mathbf{x}}_m, \ddot{\mathbf{x}}_m, \ldots, \mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}}, \ldots) \quad (8)$$

Assuming that the indirectly observable variables in the inner sensor system Z is $\mathbf{x}_{um} = \omega$, the input variable is $\mathbf{u} = [u_{Md}, \; u_{Mq}]^T$, and the directly observable variable is $\mathbf{x}_m = [i_{Md}, \; i_{Mq}]^T$, combining the voltage equation of the PMa-BSynRM, the current differential equation of the torque winding can be obtained:

$$\begin{cases} \dot{i}_{Md} = -\frac{R_M}{L_{Md}} i_{Md} + \frac{u_{Md}}{L_{Md}} + \frac{L_{Mq}}{L_{Md}} \omega i_{Mq} \\ \dot{i}_{Mq} = -\frac{R_M}{L_{Mq}} i_{Mq} + \frac{u_{Mq}}{L_{Mq}} - \frac{L_{Md}}{L_{Mq}} \omega (i_{Md} + i_0) \end{cases} \quad (9)$$

where $\dot{i}_{Md}$ and $\dot{i}_{Mq}$ are the first derivative of i_{Md} and i_{Mq}, respectively.

The corresponding Jacobi matrix is:

$$\mathbf{A}=\left[\frac{\partial\left(\mathbf{x}_m,\mathbf{x}_m\right)}{\partial\mathbf{x}_{um}}\right]=\begin{bmatrix}\frac{\partial i_{Md}}{\partial\omega}\\ \frac{\partial i_{Mq}}{\partial\omega}\\ \frac{\partial\dot{i}_{Md}}{\partial\omega}\\ \frac{\partial\dot{i}_{Mq}}{\partial\omega}\end{bmatrix}=\begin{bmatrix}0\\ 0\\ \frac{L_{Mq}}{L_{Md}}i_{Mq}\\ -\frac{L_{Md}}{L_{Mq}}\left(i_{Md}+i_0\right)\end{bmatrix} \tag{10}$$

Higher-order derivative of i_{Mq} is selected to compute the partial derivatives with respect to $\mathbf{x}_{um}$:

$$\begin{cases}\text{rank}\left(\frac{\partial\dot{i}_{Mq}}{\partial x_{um}}\right)=rank\left(-\frac{L_{Md}}{L_{Mq}}\left(i_{Md}+i_0\right)\right)=1\\ rank\left(\frac{\partial\left(\dot{i}_{Mq},\ddot{i}_{Mq}\right)}{\partial x_{um}}\right)=rank\left(\frac{\partial\left(\dot{i}_{Mq},\ddot{i}_{Mq},\dddot{i}_{Mq}\right)}{\partial x_{um}}\right)=\ldots=1\end{cases} \tag{11}$$

According to Equation (11), it can be determined that the higher derivative of i_{Mq} does not contain useful information for establishing the PMa-BSynRM speed subsystem. Moreover, the rank of the derivative of $\mathbf{x}_{um}$ is equal to the number of $\mathbf{x}_{um}$, which shows that the PMa-BSynRM speed subsystem can be established on the basis of $\dot{i}_{Mq}$, and the other derivative of i_{Md} is no longer needed.

The determinant of the Jacobi matrix is:

$$\det\left(\frac{\partial\dot{i}_{Mq}}{\partial x_{um}}\right)=-\frac{L_{Md}}{L_{Mq}}\left(i_{Md}+i_0\right)\neq 0 \tag{12}$$

Therefore, it can be judged that there is a left inverse system in the speed subsystem of the PMa-BSynRM, and the expression of the left inverse system is:

$$\mathbf{x}_{um}=\bar{f}(\mathbf{x}_m,\dot{i}_{Mq},\mathbf{u})=\bar{f}\left(i_{Md},i_{Mq},\dot{i}_{Mq},u_{Md},u_{Mq}\right) \tag{13}$$

4. THE IMPROVED BP NEURAL NETWORK SPEED LEFT INVERSE SYSTEM MODELING

4.1. Establishment of BPNN

The BPNN is a multi-layer feedforward neural network based on error back propagation, which can process information by adjusting the relationship between internal nodes. Using the advantage of BPNN that it can accurately approximate nonlinear functions, the left-invertibility model of speed subsystem can be fitted. In this paper, a three-layer BPNN including input layer, hidden layer, and output layer is selected, and all the layers are connected by weights, as shown in Fig. 3.

According to Equation (13), the number of input layer neurons is 5, and the number of output layer neurons is 1. The number of hidden layer neurons is obtained by empirical formula, which is:

$$k=\sqrt{r+v}+c \tag{14}$$

where k is the number of hidden layer neurons; r and v are the numbers of input layer neurons and output layer neurons, respectively; c is a random integer from 1 to 10.

According to Equation (14), it can be determined that the number of hidden layer neurons in BPNN model ranges from 4 to 13. The optimal number of the neurons in the hidden layer is found through experimental comparison, and the structure of neural network is determined.

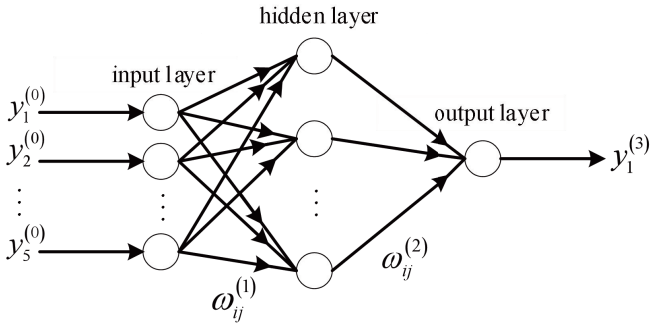

FIGURE 3. Structure of the BPNN.

In the three-layer feedforward network shown in Fig. 3, the first layer is the input layer; the second layer is the hidden layer; and the third layer is the output layer. The Sigmoid function is used as the activation function of neurons. When the number of neurons in the qth layer (q = 1, 2, 3) is N_q, the input and output of the ith neuron in the qth layer is:

$$y_i^{(q)}=f\left(s_i^{(q)}\right)=\frac{1}{1+e^{-s_i^{(q)}}} \tag{15}$$

$$s_i^{(q)}=\sum_{j=1}^{N_{q-1}}\omega_{ij}^{(q)}y_j^{(q-1)} \tag{16}$$

where $y_i^{(q)}$ and $s_i^{(q)}$ are the input and output of the ith neuron in the qth layer, respectively, and ω_{ij} is the weight of the ith neuron input from the $(q-1)$th layer to the qth layer.

In order to calculate each fitness value and observe the fitting accuracy, the following equation is given as fitness function:

$$E=\frac{1}{2}\sum_{i=1}^{N_3}(y_i-y_i^{(3)})^2 \tag{17}$$

where N is the number of training samples, y_i the expected output, and $y_i^{(3)}$ the actual output.

The use of the first-order gradient descent method is employed to adjust the weights, with the objective of minimizing the fitness function E. According to the first-order method, the BP learning algorithm can be obtained:

$$\omega_{ij}^{(q)}(l)=\omega_{ij}^{(q)}(l-1)+\eta D_{ij}^{(q)}(l) \tag{18}$$

$$\begin{cases}D_{ij}^{(q)}(l)=-\frac{\partial E}{\partial\omega_{ij}^{q}}(l)=\sum_{p=1}^{P}\delta_{pi}^{(q)}(l)y_{pj}^{(q)}(l)\\ y_{pj}^{(q)}(l)=f\left(s_{pi}^{(q)}\right)=\frac{1}{1+e^{-s_{pi}^{(q)}}}\\ \delta_{pi}^{(q)}(l)=-\frac{\partial E}{\partial s_{pi}^{(q)}}(l)=\left(\sum_{k=1}^{N_{q+1}}\delta_{pk}^{(q+1)}(l)\omega_{ki}^{(q+1)}\right)f'\left(s_{pi}^{(q)}\right)\end{cases} \tag{19}$$

where $\omega_{ij}^{(q)}(l)$ is the weight after the lth ($l = 0, 1, \ldots, L$) learning iteration; η is the learning rate; $D_{ij}^{(q)}(l)$ is the negative gradient at time l; $y_{pj}^{(q)}(l)$ is the output at the lth learning iteration; $\delta_{pi}^{(q)}(l)$ is the learning error at the lth learning iteration.

4.2. The Improvement of BPNN

The improvement of BPNN is carried out from four aspects: the determination of the number of hidden layer neurons, the optimization of the initial weight, the optimization of the weight, and the optimization of the network learning rate.

1) The Determination of the Number of Hidden Layer Neurons: In practice, different attempts are needed to determine the number of hidden layer neurons, which is not conducive to the selection of network parameters and training networks. Therefore, the normalized data is decomposed according to singular value decomposition, and the number of neurons in the hidden layer is determined according to the main eigenvalues.

Normalizing the collected sample X_{ij}:

$$X_{ij}^{(1)} = \frac{X_{ij} - m_j}{s_j} \tag{20}$$

where $X_{ij}^{(1)}$ is the normalized collected samples; m_j and s_j are mean and standard deviation of $X_{.j}$, respectively.

In order to further eliminate the difference between the training data caused by the missing values and different sensors, unit vectorization of the sample is needed:

$$X_{ij}^{(2)} = \frac{X_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{n}\left(X_{ij}^{(1)}\right)^2}} \tag{21}$$

where $X_{ij}^{(2)}$ is collected samples after unit vectorization.

The processed training data is arranged into the matrix X of m rows and n columns, and guarantees $m > n$:

$$X = \begin{bmatrix} X_{11}^{(2)} & X_{12}^{(2)} & \ldots & X_{1n}^{(2)} \\ X_{21}^{(2)} & X_{22}^{(2)} & \ldots & X_{2n}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ X_{m1}^{(2)} & X_{m2}^{(2)} & \ldots & X_{mn}^{(2)} \end{bmatrix} \tag{22}$$

Singular value of matrix x is decomposed into:

$$X = U\Sigma V^T \tag{23}$$

where U is the $m \times m$ unitary matrix, Σ the positive semi-definite $m \times n$ diagonal matrix, and V^T the $n \times n$ unitary matrix.

The elements Σ_i on the diagonal of the matrix Σ are arranged from large to small, namely $\Sigma_1, \Sigma_2, \ldots, \Sigma_n$:

$$\sum_i = \sqrt{\lambda_i} \quad (i = 1, 2, \ldots, n) \tag{24}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of $X^T X$.

Since m is greater than n, the eigenvalue of $X^T X$ is easy to get, so the singular value of matrix X can be calculated by the eigenvalue of $X^T X$. Finally, find an integer k that satisfies:

$$k = \min_k \left\{ \left(\sum_{j=1}^{k} \sum_{j}^{2} / \sum_{i=1}^{n} \sum_{i}^{2} \right) \geq 0.99 \right\} \tag{25}$$

The integer k can be used as the best estimate of the number of BPNN hidden layer neurons, and the structure of BPNN can be determined directly without repeated experiments [16].

2) The Optimization of the Initial Weight: In neural network learning, the final outcome and the convergence rate of network learning are significantly influenced by the selection of initial network weights. The rapid convergence of the network model to the optimal solution can be facilitated by ideal initial values. In BPNN, BP learning algorithm usually adopts gradient descent method, so it is more sensitive to the selection of initial weights. In this paper, uniformly distributed random numbers are used to initialize the weights of BPNN and ensure that the sum of the input weights for each neuron is 1. This approach is beneficial for avoiding excessive initial weights and prevents the Sigmoid function from entering the saturation zone, which makes the gradient in the back propagation become very small, and then the network parameters are difficult to be effectively adjusted.

3) The Optimization of the Weight: In essence, the BP learning algorithm mentioned in Equation (18) is a simple static optimization algorithm with rapid descent. When correcting the weight, it only corrects according to the negative gradient at time l, without considering the gradient direction at previous moment, thus causing the learning process to oscillate. In order to solve the above problems, the momentum term is added on the original basis:

$$\Delta\omega_{ij}^{(q)}(l) = \alpha\Delta\omega_{ij}^{(q)}(l-1) + \eta D_{ij}^{(q)}(l)\, y_i^{(q)} \tag{26}$$

where $\Delta\omega_{ij}^{(q)}(l)$ is the incremental change of weight from the ith layer to the jth layer, α the forgetting factor, and $\alpha\Delta\omega_{ij}^{(q)}(l-1)$ the momentum term.

4) The Optimization of the Network Learning Rate: Considering that the learning rate of the original BP algorithm is constant, in order to prevent the learning process from oscillating and diverging due to the excessive decline of errors, the learning rate of BPNN is usually small enough. However, if the learning rate is too small, it will inevitably reduce the convergence speed of BPNN learning process. Therefore, the adaptive law of learning rate is put forward to automatically adjust the learning rate:

$$\Delta\eta(l) = \begin{cases} a, & \delta(l) < 0 \\ -b\eta(l-1) & \delta(l) > 0 \\ 0, & otherwise \end{cases} \tag{27}$$

where $\Delta\eta(l)$ is the compensation value of the lth learning rate of BP algorithm; a and b are the positive constants; $\delta(l) =$

TABLE 1. The comparison of the two networks.

Network	Number of hidden layer neurons	Mean number of iterations	Fitness error/10^{-3}
BPNN	11	179	2.8
IBPNN	10	62	0.38

$\frac{\partial E}{\partial \omega_{ij}^{q}}(l)$ is the gradient value of square difference function in the lth iteration.

The iteration speed and accuracy of BPNN and IBPNN are shown in Table 1.

As shown in Table 1, the IBPNN has been greatly improved compared with the traditional BPNN in terms of average iteration times and fitting error, which shows that the IBPNN has a good training effect. In addition, there is little difference in the optimal number of the hidden layer neurons, which indicates that the IBPNN can avoid repeated experiments required by the BPNN.

4.3. Speed Self-Sensing System Modeling

In this paper, IBPNN is used to establish the PMa-BSynRM speed self-sensing system model. The flowchart of IBPNN is shown in Fig. 4, and the specific steps are as follows:

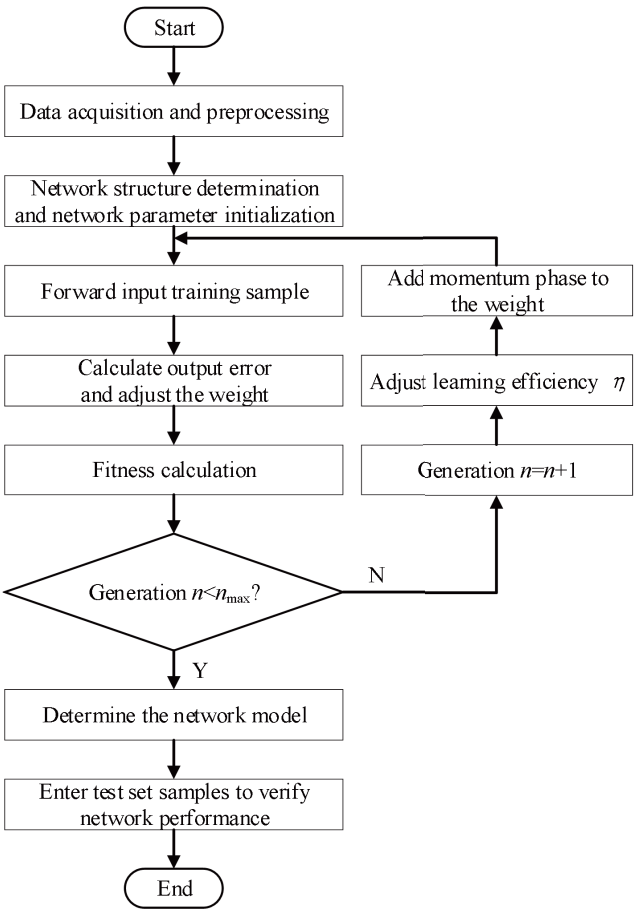

FIGURE 4. The flow chart of IBPNN.

1) According to Equation (13), it is confirmed that the inputs and outputs of IBPNN are $\mathbf{x} = (i_{Md}, i_{Mq}, \dot{i}_{Mq}, u_{Md}, u_{Mq})^T$ and $\mathbf{y} = \omega$. Sample the data with a sampling interval of 1 second and a precision of 1 millisecond, resulting in 1000 sets of sampled data. Divide the data into the training set comprising 80% of the data and the testing set consisting of the remaining 20%.

2) The structural parameters of the IBPNN are determined before the commencement of training. The sampling data is normalized, and singular value decomposition is performed on the normalized data. The number of neurons in the hidden layer is determined based on the principal eigenvalues.

3) The number of connection weights is confirmed by the given neural network structure, and the network connection weights are initialized by uniformly distributed random numbers.

4) The training set samples are taken for forward propagation, and the input and output of neurons in each layer are obtained. The output error is calculated and propagated back, and the connection weight is updated by the improved BP algorithm.

5) When the error between the actual speed and predicted speed is less than the minimum error requirement or reaches the maximum number of iterations, the network model is determined, and the prediction results are output; otherwise, return to step 4.

6) The test set samples are input into the determined network model to verify the network performance and analyze the simulation results.

Mean absolute error (MAE), mean square error (MSE), and root mean square error (RMSE) are all indicators for evaluating the performance of the network model. The three error expressions are as follows:

$$\begin{cases} \delta_{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - y_i^{(3)}\right| \\ \delta_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^{(3)}\right)^2 \\ \delta_{RMSE} = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\left(y_i - y_i^{(3)}\right)^2} \end{cases} \tag{28}$$

where n is the total number of samples in the test set.

The various errors of BPNN, particle swarm optimization (PSO)-BPNN [17], and IBPNN are shown in Table 2. As can be seen from the table, the errors of IBPNN are smaller than those of BPNN and PSO-BPNN, which shows that the prediction accuracy and training effect of IBPNN are better.

Combined with the IBPNN model, the speed sensorless control system of the PMa-BSynRM is established, as shown in Fig. 5.

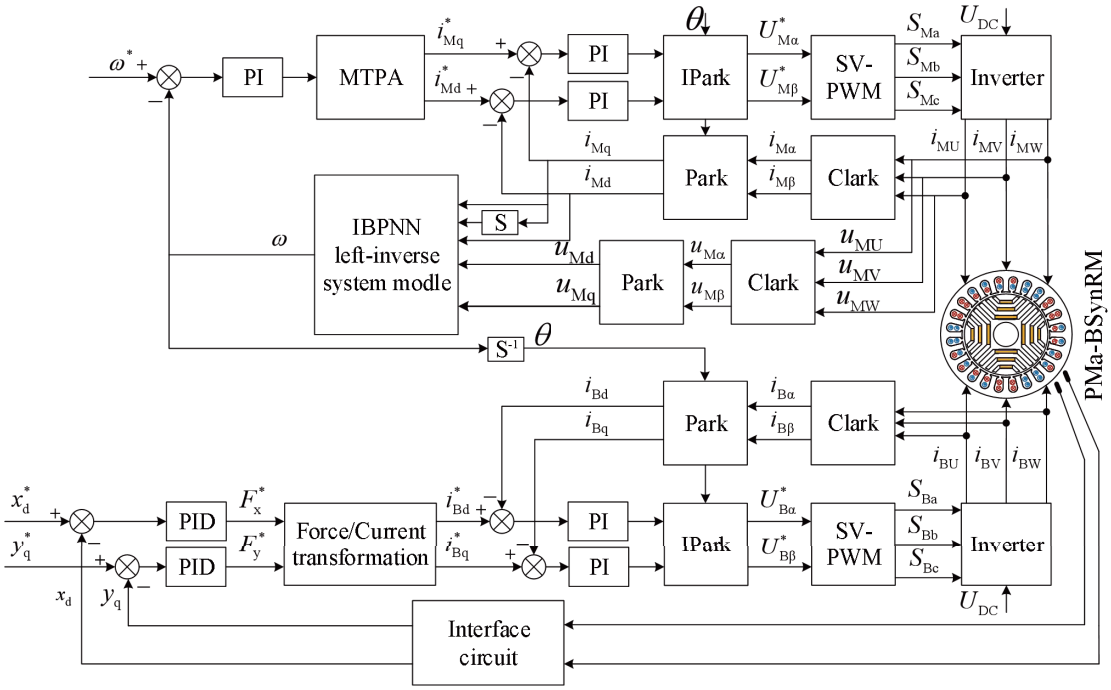

FIGURE 5. Speed self-sensing control block diagram of the PMa-BSynRM.

TABLE 2. Various errors of BPNN, PSO-BPNN and IBPNN.

Network	MAE/10^{-3}	MSE/10^{-5}	RMSE/10^{-3}	training accuracy/%
BPNN	1.62	2.87	5.36	91.2
PSO-BPNN	0.75	0.56	2.37	93.6
IBPNN	0.46	0.42	2.05	95.7

5. SIMULATION TEST

5.1. Network Optimization

BPNN and IBPNN are compared in MATLAB. The initial learning rate η is set to 0.01. The minimum error requirement is set to 4×10^{-4}, and the maximum number of iterations is set to 200. The range of forgetting factor α in the improved BP algorithm is set to $(0, 1)$. In the adaptive learning rate algorithm, the constant a is set to 0.01 and the constant b set to 0.1. The fitness curves of the IBPNN and standard BPNN are shown in Fig. 6. It can be seen that the standard BPNN starts to converge in the 179th generation with a fitness value of 2.845×10^{-3} and does not meet the error requirements within 200 generations, whereas the IBPNN starts to converge in the 57th generation with a fitness value of 3.751×10^{-4} and meets the error requirements. Overall, the IBPNN has higher fitting accuracy and faster convergence speed than the standard BPNN.

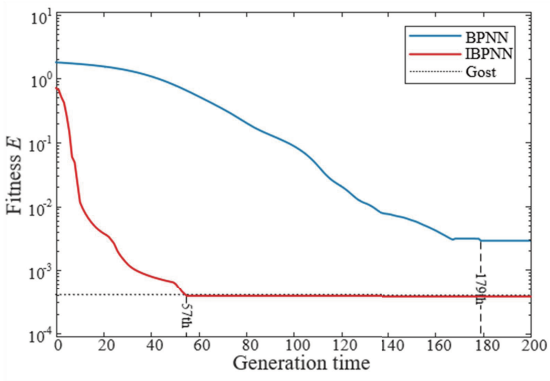

FIGURE 6. The fitness curves of the IBPNN and BPNN.

5.2. Speed Tracking

Considering the gravity of the rotor, the initial displacements of the rotor are set to $x = 0$ mm and $y = -0.24$ mm, respectively. The rated speed is 5000 r/min, and the control period of the system is 0.1 ms. The simulated waveform of rotor speed is shown in Fig. 7. At 0 s, the speed starts to rise from 0 to 3000 r/min. The speed of BPNN reaches 3000 r/min at 0.046 s, and the adjustment time is 16 ms. The speed of IBPNN reaches 3000 r/min at 0.045 s, and the adjustment time is 14 ms. At 0.1 s, the speed continues to rise from 3000 to 5000 r/min. The speed of BPNN reaches 5000 r/min at 0.147 s, and the adjustment time is 20 ms. The speed of IBPNN reaches 5000 r/min at 0.14 s, and the adjustment time is 18 ms.

From the above data analysis, it is illustrated that IBPNN can track the actual speed well within the allowable speed error

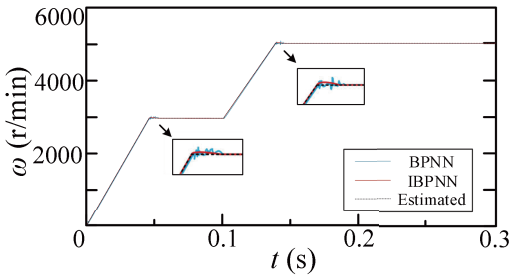

FIGURE 7. The simulated waveform of the speed when the speed accelerates from 0 r/min to 5000 r/min.

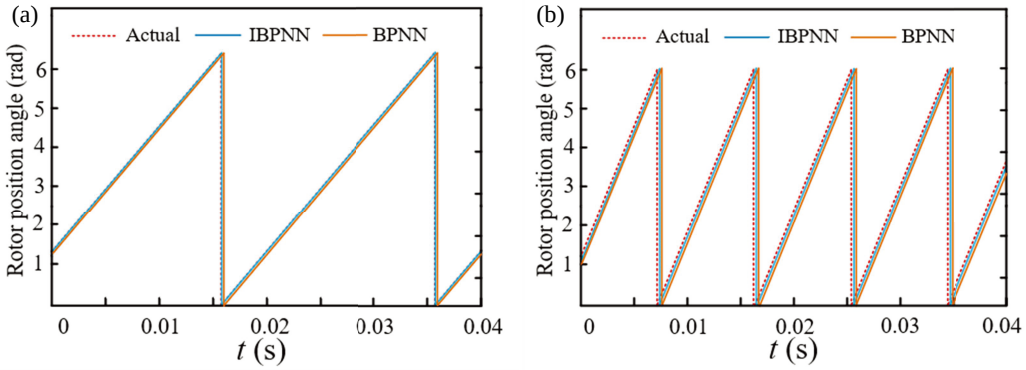

FIGURE 8. The simulated waveforms of rotor position. (a) The speed is 3000 r/min. (b) The speed is 5000 r/min.

range, but the BPNN method needs longer adjustment, and the speed error in the adjustment process is large. Compared with BPNN, the dynamic fitting result of IBPNN is more satisfactory.

5.3. Rotor Position Tracking

The simulated waveforms of rotor position angle, when the speed changes, are shown in Fig. 8. As shown in Fig. 8, the rotor position estimation error of the BPNN method increases with the increase of rotational speed, and the average position estimation error at 3000 r/min and 5000 r/min is 0.104 rad and 0.359 rad, respectively. The rotor position estimation error of the IBPNN method increases with the increase of rotational speed, and the average position estimation error at 3000 r/min and 5000 r/min is 0.036 rad and 0.203 rad, respectively.

As can be seen from the above, the estimated rotor position, as determined by the IBPNN and BPNN controller, can be dynamically tracked to the actual rotor position. Compared with BPNN method, the rotor position error of the IBPNN method is smaller, and the angle error only increases slightly with the increase of speed, which shows that the IBPNN method can effectively improve the accuracy of rotor position estimation.

5.4. Rotor Displacement

The simulated waveforms of radial displacement in the x- and y-directions are shown in Fig. 9. From Fig. 9, it can be seen that when the speed changes obviously, the maximum magnitudes of BPNN in the x- and y-directions are 7 μm and 19 μm, respectively, and the maximum magnitudes of IBPNN in the x- and y-directions are 6 μm and 17 μm, respectively. When the speed is stable, the motor can be stably suspended on the axis.

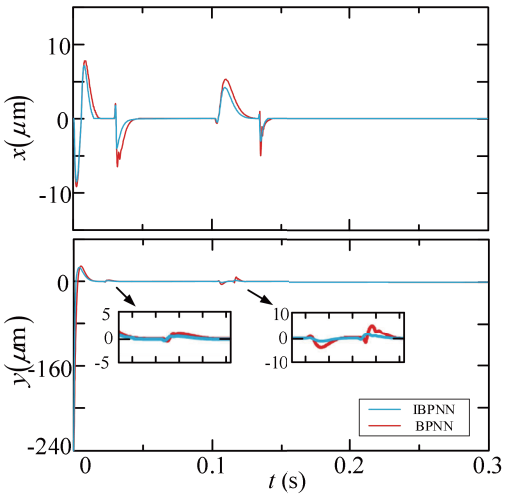

FIGURE 9. The simulated waveform of radial displacement in the x- and y-direction when the speed accelerates from 0 r/min to 5000 r/min.

As can be seen from the above, under the two control methods, the radial displacement of the rotor is much less than the 0.25 mm clearance between the rotor and the auxiliary bearing, which realizes the stable suspension operation of the rotor. The suspension force performance of the IBPNN control method is superior to the traditional BPNN control method, mainly because the rotor angle needs to be estimated in the control of the

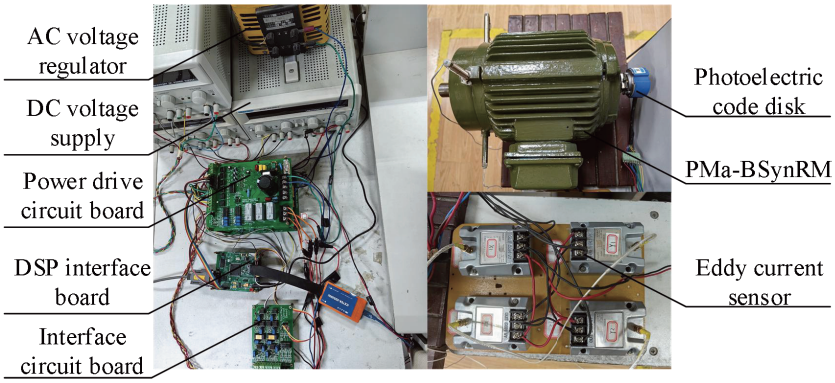

FIGURE 10. The experimental platform of PMa-BSynRM.

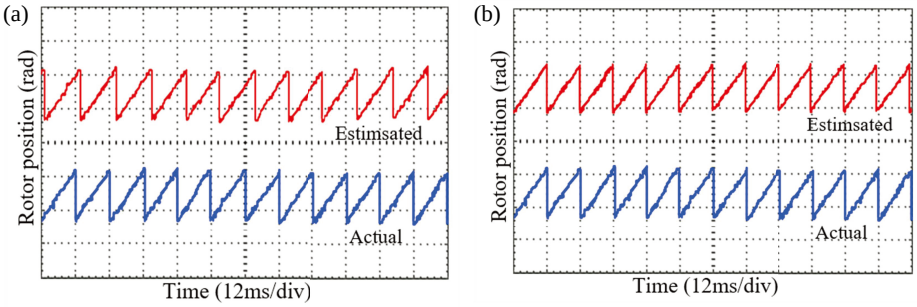

FIGURE 11. The experimental waveforms of rotor position when the speed is 5000 r/min. (a) BPNN. (b) IBPNN.

suspension force system, as shown in Fig. 5, and the rotor angle fitted by IBPNN method has higher accuracy. Therefore, the accuracy of rotor angle estimation will also affect the suspension performance of the PMa-BSynRM.

6. THE PMA-BSYNRM EXPERIMENTAL VALIDATION

In order to further validate the effectiveness of the proposed program, a digital control system is designed for experimental verification based on a PMa-BSynRM, using TMS320F28335 as the core processor, and the experimental platform is shown in Fig. 10. It mainly includes a PMa-BSynRM, digital signal processor (DSP), power board, interface circuit board, PC host, photoelectric encoder, etc. One end of the PMa-BSynRM is supported by a three-degree-of-freedom aligning ball bearing, and the other end is equipped with an auxiliary bearing, which can realize two-degree-of-freedom suspension. The PMa-BSynRM is driven by an inverter, and the power supply module uses an insulated gate bipolar transistor (IGBT) three-phase full-bridge circuit. The digital signal processor (DSP) is used to monitor and process the experimental data. The displacement signal and speed signal of the PMa-BSynRM are obtained by eddy current sensors and photoelectric code disks, respectively.

The main specifications and parameters of the platform are shown in Table 3.

The training samples of the IBPNN in sensorless experiments are provided by open-loop experiments, and the expression of

TABLE 3. Parameters and description of the experimental platform.

Symbol (unit)	Values	Symbol (unit)	values
U_n (V)	220	δ_0 (mm)	0.25
P_n (kW)	1.1	Ψ_0 (Wb)	0.0359
n_n (r/min)	5000	N_M/N_B	72/24
P_M/P_B	2/1	K_m	0.00059
m (kg)	1.6	PM material	NdFeB
L_{SI}/L_{SO} (mm)	80/130	L_M (mm)	1.65

open-loop velocity is:

$$\begin{cases} \omega = \frac{L_{Md}\dot{i}_{Md}+R_M i_{Md}-u_{Md}}{L_{Mq} i_{Mq}} \\ \omega = \frac{L_{Mq}\dot{i}_{Mq}+R_M i_{Mq}-u_{Mq}}{-L_{Md}(i_{Md}+i_0)} \end{cases} \tag{29}$$

Because the open-loop experiment has poor stability and large error in actual operation, it is necessary to estimate the speed of the PMa-BSynRM system with trained neural network. Therefore, the experiment is based on the offline BPNN.

The static suspension experiment and speed variation experiment are all based on the no-load experiment.

6.1. Steady-State Experiment

In this section, the steady-state performances of the traditional BPNN and proposed IBPNN are compared.

The experimental waveforms of rotor position when the rotor is running at 5000 r/min and stably suspended are shown in Fig. 11. The average position estimation error of the

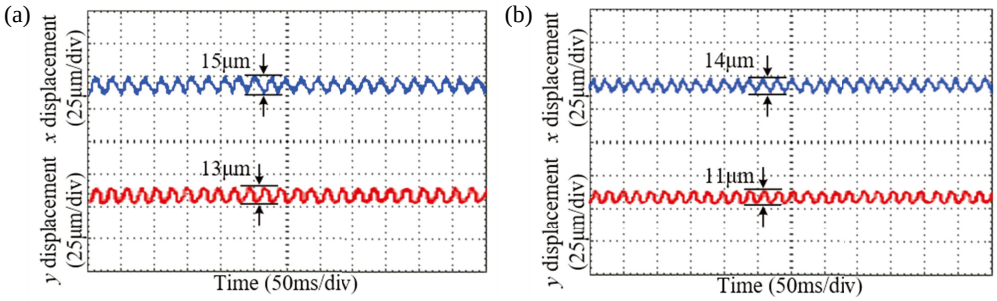

FIGURE 12. The experimental waveforms of radial displacement in the x- and y-directions when the speed is 5000 r/min. (a) BPNN. (b) IBPNN.

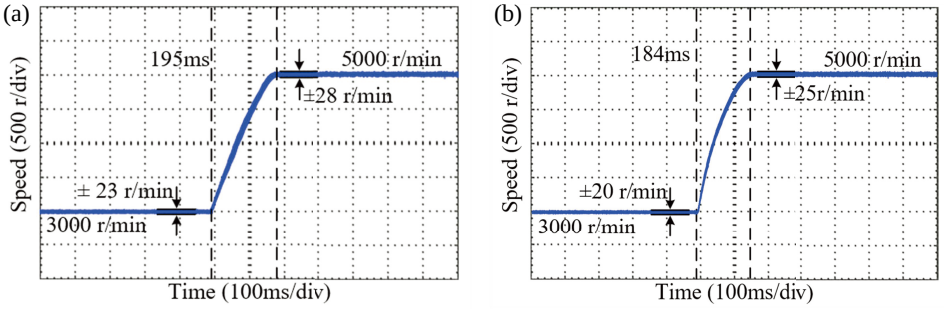

FIGURE 13. The experimental waveforms of motor speed when the speed accelerates from 3000 r/min to 5000 r/min. (a) BPNN. (b) IBPNN.

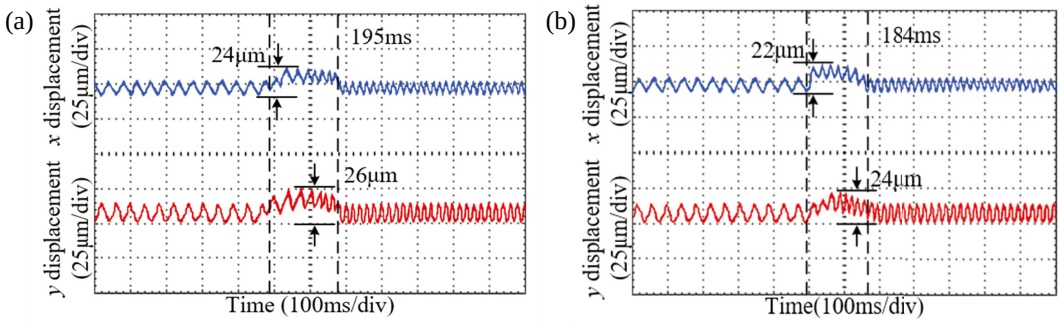

FIGURE 14. The experimental waveforms of radial displacement in the x- and y-directions when the speed accelerates from 3000 r/min to 5000 r/min. (a) BPNN. (b) IBPNN.

BPNN method at 5000 r/min is 0.478 rad, and the average position estimation error of the IBPNN method at 5000 r/min is 0.247 rad. Compared with BPNN method, the rotor position estimation error based on IBPNN method decreases by 48.32% at 5000 r/min.

The experimental waveforms of radial displacement in the x- and y-directions when the speed is 5000 r/min are shown in Fig. 12. The blue and red lines in Fig. 12(a) represent the BPNN experimental waveforms in the x- and y-directions when the speed is 5000 r/min, respectively, and the blue and red lines in Fig. 12(b) represent the IBPNN experimental waveforms in the x- and y-directions when the speed is 5000 r/min, respectively. The displacement amplitudes of BPNN in the x- and y-directions are 15 μm and 13 μm, respectively, and the displacement amplitudes of IBPNN in the x- and y-directions are 14 μm and 11 μm, respectively. It can be seen that compared with BPNN method, at steady state, the rotor position fitted by IBPNN method is more accurate, so its suspension force performance is better, which is consistent with the simulation results.

6.2. Speed Variation Experiment

In this section, the dynamic performances of the traditional BPNN and proposed IBPNN are compared in acceleration condition.

The experimental waveforms of motor speed when the speed accelerates from 3000 r/min to 5000 r/min are shown in Fig. 13. The speed fluctuation of BPNN method is ±23 r/min at 3000 r/min, and the speed fluctuation is ±28 r/min at 5000 r/min. The adjustment time during acceleration is 195 ms. The speed fluctuation of IBPNN method is ±20 r/min at 3000 r/min, and the speed fluctuation is ±25 r/min at 5000 r/min. The adjustment time during acceleration is 184 ms. Compared with BPNN method, the motor speed error based on IBPNN method decreases by 13.04% at

3000 r/min and by 10.71% at 5000 r/min, and the adjustment time decreases by 5.64%. It can be seen that compared with BPNN method, IBPNN method has better fitting accuracy and dynamic tracking performance.

The experimental waveforms of radial displacement in the x- and y-directions when the speed accelerates from 3000 r/min to 5000 r/min are shown in Fig. 14. The blue and red lines in Fig. 14(a) represent the BPNN experimental waveforms with x- and y-directions when the speed accelerates from 3000 r/min to 5000 r/min, respectively, and the blue and red lines in Fig. 14(b) represent the IBPNN experiments with x- and y-directions when the speed accelerates from 3000 r/min to 5000 r/min, respectively. The displacement amplitude of BPNN method is about 24 µm and 26 µm in the x- and y-directions, respectively, and the displacement amplitude of IBPNN method is about 22 µm and 24 µm in the x- and y-directions, respectively. As can be seen from the above, compared with BPNN method, when the motor changes speed, suspension force performance of IBPNN method is better.

7. CONCLUSION

In this paper, a speed sensorless control method of the PMa-BSynRM is proposed. The rotor speed is fitted by establishing the IBPNN model among the voltage and current of the torque windings and the rotor speed. Through theoretical analysis, simulation, and experimental study, the following conclusions are drawn:

1. The proposed IBPNN improves the situation that the BP algorithm is easy to fall into the local optimum, reduces the oscillation trend during the learning process, and improves the optimization accuracy of the algorithm while speeding up the convergence speed.

2. The strategy uses singular value decomposition method to estimate the number of hidden layer neurons, which solves the problem that different attempts are needed to determine the number of hidden layer neurons in practice.

3. The experimental results show that compared with BPNN method, the speed sensorless control of the PMa-BSynRM based on IBPNN method has better tracking performance, control accuracy, and suspension force performance in the steady-state and speed variation.

4. The future work will focus on improving the tracking performance of the speed in the start-up phase of the PMa-BSynRM.

ACKNOWLEDGEMENT

This project was sponsored in part by National Natural Science Foundation of China (62273168).

REFERENCES

[1] Zhu, H. and Y. Shi, "Displacement self-sensing control of permanent magnet assisted bearingless synchronous reluctance motor based on least square support vector machine optimized by improved NSGA-II," *IEEE Transactions on Industrial Electronics*, Vol. 71, No. 2, 1201–1211, Feb. 2024.

[2] Pei, T., D. Li, J. Liu, and W. Kong, "A novel flux-reversal bearingless slice motor controlled by rotor angle independent suspension current," *IEEE Transactions on Industrial Electronics*, Vol. 70, No. 6, 5615–5625, Aug. 2022.

[3] Islam, M. Z., A. Arafat, and S. Choi, "Design of five-phase bearingless permanent magnet assisted synchronous reluctance motor for high speed applications," in *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, 4419–4424, Portland, OR, USA, Sep. 2018.

[4] Ding, H., H. Zhu, and Y. Hua, "Optimization design of bearingless synchronous reluctance motor," *IEEE Transactions on Applied Superconductivity*, Vol. 28, No. 3, 1–5, Apr. 2018.

[5] Prabhakaran, K. K. and A. Karthikeyan, "Electromagnetic torque-based model reference adaptive system speed estimator for sensorless surface mount permanent magnet synchronous motor drive," *IEEE Transactions on Industrial Electronics*, Vol. 67, No. 7, 5936–5947, Jul. 2020.

[6] Sun, C., Z. Yang, X. Sun, and Q. Ding, "An improved mathematical model for speed sensorless control of fixed pole bearingless induction motor," *IEEE Transactions on Industrial Electronics*, Vol. 71, No. 2, 1286–1295, Feb. 2024.

[7] Gao, F., Z. Yin, C. Bai, D. Yuan, and J. Liu, "Speed sensorless control method of synchronous reluctance motor based on resonant Kalman filter," *IEEE Transactions on Industrial Electronics*, Vol. 70, No. 8, 7627–7641, Aug. 2023.

[8] Pasqualotto, D., S. Rigon, and M. Zigliotto, "Sensorless speed control of synchronous reluctance motor drives based on extended Kalman filter and neural magnetic model," *IEEE Transactions on Industrial Electronics*, Vol. 70, No. 2, 1321–1330, Feb. 2023.

[9] Mynar, Z., P. Vaclavek, and P. Blaha, "Synchronous reluctance motor parameter and state estimation using extended Kalman filter and current derivative measurement," *IEEE Transactions on Industrial Electronics*, Vol. 68, No. 3, 1972–1981, Mar. 2021.

[10] Qiao, Z., T. Shi, Y. Wang, Y. Yan, C. Xia, and X. He, "New sliding-mode observer for position sensorless control of permanent-magnet synchronous motor," *IEEE Transactions on Industrial Electronics*, Vol. 60, No. 2, 710–719, Feb. 2013.

[11] Repecho, V., J. B. Waqar, D. Biel, and A. Dòria-Cerezo, "Zero speed sensorless scheme for permanent magnet synchronous machine under decoupled sliding-mode control," *IEEE Transactions on Industrial Electronics*, Vol. 69, No. 2, 1288–1297, Feb. 2022.

[12] Chen, S., W. Ding, X. Wu, L. Huo, R. Hu, and S. Shi, "Sensorless control of IPMSM drives using high-frequency pulse voltage injection with random pulse sequence for audible noise reduction," *IEEE Transactions on Power Electronics*, Vol. 38, No. 8, 9395–9408, Aug. 2023.

[13] Chen, J., Y. Fan, W. Wang, C. H. T. Lee, and Y. Wang, "Sensorless control for SynRM drives using a pseudo-random high-frequency triangular-wave current signal injection scheme," *IEEE Transactions on Power Electronics*, Vol. 37, No. 6, 7122–7131, Jun. 2022.

[14] Han, F., Z. Wang, Y. Li, and T. Han, "Sensorless speed control of permanent magnet synchronous motor based on RBF neural network," in *2015 34th Chinese Control Conference (CCC)*, 4325–4330, Hangzhou, China, Sep. 2015.

[15] Cirrincione, M., A. Accetta, M. Pucci, and G. Vitale, "MRAS speed observer for high-performance linear induction motor drives based on linear neural networks," *IEEE Transactions on*

Power Electronics, Vol. 28, No. 1, 123–134, Jan. 2013.

[16] Cai, G.-W., Z. Fang, and Y.-F. Chen, "Estimating the number of hidden nodes of the single-hidden-layer feedforward neural networks," in *2019 15th International Conference on Computational Intelligence and Security (CIS)*, 172–176, Macao, 2019.

[17] Wang, J., G. Liu, and H. Zhu, "Displacement self-sensing control of permanent magnet assisted bearingless synchronous reluctance motor based on BP neural network optimized by improved PSO," *Progress In Electromagnetics Research C*, Vol. 143, 189–198, 2024.