

Article

Optimization of a Multi-Type PMSM Based on Pyramid Neural Network

Xiaoyu Liu *, Wenqian Peng, Liuyin Xie and Xiang Zhang

National Center for Applied Mathematics in Chongqing, Chongqing Normal University, Chongqing 401333, China; 2022110518019@stu.cqnu.edu.cn (W.P.); xly66666@126.com (L.X.); zx_2022211@163.com (X.Z.)

* Correspondence: xyliu@cqnu.edu.cn

Abstract: In this paper, a novel bat algorithm based on the quantum computing concept and pyramid neural network (PNN) is presented and applied to the electromagnetic motor optimization problem. Due to the problems of high loss, high temperature rise and threatening motor safety, it is necessary to optimize the design of high-speed permanent magnet synchronous motor (HPMSM) structure. In order to use less training data and avoid the problem of large computational costs due to repeated finite element simulation in the electromagnetic structure design, this paper adopted a performance-driven method to establish the PMSM model. This model could effectively reduce the dimensions of the parameter space and establish an effective high-quality model within a wide range of parameters. For the purpose of obtaining a reliable proxy model with less training data, this paper adopted a pyramid-shaped neural network, which could reduce the risk of overtraining and improve the utilization of specific problem knowledge embedded in the training data set. The quantum bat algorithm (QBA) was used to optimize the structure of the PMSM. Compared with the classical GA and PSO algorithms, the QBA has the characteristics of a rapid convergence speed, simple structure, strong searching ability and stronger local jumping mechanism. The correctness and effectiveness of the proposed PNN-based QBA method were verified using simulation analysis and a prototype test.

Keywords: high-speed permanent magnet synchronous motor; optimization algorithm; pyramid neural network; quantum bat algorithm



Citation: Liu, X.; Peng, W.; Xie, L.; Zhang, X. Optimization of a Multi-Type PMSM Based on Pyramid Neural Network. *Appl. Sci.* **2023**, *13*, 6810. <https://doi.org/10.3390/app13116810>

Academic Editor: Gang Lei

Received: 11 May 2023

Revised: 30 May 2023

Accepted: 31 May 2023

Published: 3 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The development of material technology has promoted the analysis and synthesis of modern structures of permanent magnet excited machines. Researchers have worked on improving machine parameters affecting magnetic, thermal and mechanical properties to allow for a greater density of energy. Compared with ordinary motors, high-speed permanent magnet synchronous motors (HPMSMs) [1] have obvious advantages such as a high speed, small geometric size, large power density and small moment of inertia [2]. HPMSMs can be directly connected with elevated-speed loads, which can eliminate traditional mechanical transmission devices, and they not only reduce mechanical noise but also improve the efficiency of the transmission system [3]. HPMSMs have broad application prospects in centrifugal compressors, flywheel energy storage, high speed grinders, hybrid electric vehicles, aviation, marine systems, distributed power generation systems and other fields [4]. However, under the collective influence of temperature, armature reaction, mechanical vibration and other factors, a serious temperature rise in the inner components of the motor will cause demagnetization of the permanent magnet [5]. This will lead to the degradation of the motor's performance and insufficient output. Heavy loss in an HPMSM will make it unable to drive the load and will threaten the safety of the system [6]. To obtain better performance from an HPMSM, developing its optimized construction has become a research hotspot [7].

Compared with the traditional algorithm, the concept of the meta-heuristic optimization algorithm is simple, easy to implement and can optimize the problem without requiring a lot of gradient information. Therefore, in recent years, meta-heuristic optimization algorithms have gradually replaced traditional algorithms and have been applied to practical problems. Currently, algorithms can be divided into five categories according to their source of inspiration: swarm intelligence algorithm (SI), evolutionary algorithm (EA), memetic algorithm (MA), human-based algorithm, physics-based algorithm (PhA) [8] and hybrid algorithm. The improved symbiotic organisms search (ISOS) [9] is an evolutionary algorithm with a good global search ability and convergence. The modified version of the salp swarm algorithm (mSSA) [10] is a swarm intelligence algorithm which has higher precision and a better convergence effect. The information-exchanged Gaussian arithmetic optimization algorithm with quasi-opposition learning (iegqos-AOA) [11] based on quasi-opposite learning is a kind of PhA. It has the characteristics of a simple algorithm structure and fast convergence. The GASA algorithm [12], based on the genetic algorithm and simulated annealing algorithm, is a hybrid algorithm, which has the characteristics of a simple structure and low calculation cost.

Quantum evolutionary computation [13] has become an important research hotspot in the field of intelligent computing in recent years. It makes full use of the theory of quantum bits and superposition states in quantum computation, uses qubit coding to represent individuals and employs quantum gate updating to perform evolutionary operations. At present, the combination of quantum theory and intelligent algorithms has become a popular research method [14]. The quantum bat algorithm (QBA) [15] incorporates the concept of quantum evolutionary computation into the bat algorithm, allowing it to fully utilize the advantages of the algorithm. Compared with classical algorithms, the QBA has the advantages of fast convergence, a simple structure, a strong search ability and a robust local jump mechanism. However, due to the large amount of finite element analysis required in the optimization algorithm, the cost of online calculation is very high. Therefore, it is necessary to introduce an effective model that can replace the finite element simulation in structural optimization design for permanent magnet synchronous motors.

Industry and researchers have been designing numerical solvers for decades to enable the rapid analysis of electromagnetic fields in motors. However, in the iterative process of optimization, the accurate finite element analysis of the electromagnetic field inside the motor usually requires a large amount of computational time. Therefore, numerous researchers began to try to use various agents, hoping to calculate the objective function in the motor optimization problem using the fast approximation method, so as to considerably reduce the computation time of finite element analysis [16]. For example, the response surface model (RSM) has been applied to the optimization of electric motor design [17]. These methods are usually proposed specifically to a problem. The description of the systems generally contains only a few parameters. Thus, these surrogate models lack the ability to be widely applicable to different types of motor structure design [18].

Recently, many meta-modeling methods based on big data [19] and machine learning have attracted wide attention. Neural networks (NNs) based on fuzzy logic have been used to model the influence of clearance on product quality and the blanking force in blanking process [20]. Artificial neural networks (ANNs) have been used to model the nonlinear relationship between the power harvested with a piezoelectric pendulum and its resistive load, magnetic excitation frequency, etc. [21]. A multilayer feed-forward artificial neural network algorithm was proposed and applied to estimate the output power and efficiency of an axial flux permanent magnet synchronous generator (AFPMSG) [22].

Among these meta-modeling methods, the deep learning (DL) algorithm is more related to the current problem of solving Poisson's equation [23]. It has the advantages of scalability of high-dimensional data, the ability to deal with massive data, opening up a new field for pattern recognition and curve fitting of complex problems such as dense regression, easy parallelization and automatic feature extraction. Therefore, the use of neural networks that are appropriate for electromagnetic field simulation problems may accelerate seeking

the solution of them. The DL model has particularly strong approximation ability, given a source of dielectric constant and the distribution information of correct predictions. Khan proposed to use the DL model to predict the solution of the Maxwell equation [24], the main electromagnetic calculation equation of electromagnetic equipment, and conducted a lot of research on this topic. The work took a coil, a transformer and a permanent magnet synchronous motor as the research object, and used the data obtained from the traditional finite element calculation. A deep convolutional neural network was used for training based on supervised learning to learn the law of magnetic field distribution [25]. The use of deep networks suitable for magnetic field analysis tasks may accelerate the solution of motor optimization problems.

In a deep neural network (DNN) [26], the information extracted from different layers is insufficient and the feature map of different layers is used for prediction. The performance of the network may be affected if the features of some layers in the basic network are not sufficiently obtained. The multi-branch feature pyramid network refers to the feature map of different layers obtained through the bottom-up basic network and top-down transverse connection, and the feature map of different layers can then be predicted. It has been used in various fields such as target detection, image segmentation, saliency target detection and image detection. A multi-branch network [27] has one branch referring to a basic network structure, and other branches with other structures. The basic network obtains rough features, and, after adding other branches, obtains richer details or spatial information. The purpose of adding branches is to obtain richer and more detailed features on the basis of rough features.

The multilevel feature and context pyramid network (MLFCPNet) [28] uses structural information and channel relationships to better represent features. The spatial pyramid attention net (SPANet) [29] enhances the underlying network by horizontally adding spatial pyramid attention (SPA) blocks. It uses both structural regularization and structural information to achieve a better learning ability. The recognition accuracy is significantly improved without significantly increasing the computation required. It is simple in concept, but powerful in practice.

The accuracy of a deep learning model largely depends on the training data. The existing works of deep learning-based motor electromagnetic field analysis can only be used for the simulation analysis of a specific type of motor, because the training data source is relatively individual.

In this paper, a regression modeling method based on a pyramid neural network (PNN) [30] combined with a DNN with spatial pyramid attention (SPA) blocks was employed. Based on the QBA with the PNN model, the structural parameters of the HPMSM were optimized to obtain the optimal combination to minimize the losses from the motor. The network structure of the PNN multi-hidden layer calculation model of the HPMSM provided high-precision and high-efficiency model support for the QBA global optimization. At the same time, in order to make the proposed algorithm universal, the general model was used to generate a training data set, considering the embedded type including radial, V type, U type and surface-mounted type [31] magnets as alternative HPMSM structures in this paper. The performance of the optimized HPMSM was verified using finite element simulation analysis and prototype experiments.

2. Dataset Preparation

In this study, we first established the electromagnetic field distribution database of the permanent magnet synchronous motor (PMSM) as training data. Since the complexity and quality of training samples will directly affect the accuracy of the prediction model, we adopted the general pattern of the PMSM in [32]. Then, the electromagnetic field inside the motor was simulated using finite element analysis to generate the electromagnetic field distribution database. A total of 600 groups of data were established for each type of PMSM. Moreover, 2400 groups of distributed data meeting the training accuracy requirements of

neural network were used as training data to predict the electromagnetic field distribution of PMSM with different structures.

2.1. Alternative Arrangement of PMs

In order to make the training database established in this paper have wide applicability, the rotor structure mainly used by the PMSM is shown in Figure 1. The arrangement of magnets shown in Figure 1 (i.e., U-shaped, V-shaped, radial, and surface mounted) could be obtained by adjusting the geometric parameters of the general pattern of the PM arrangement.

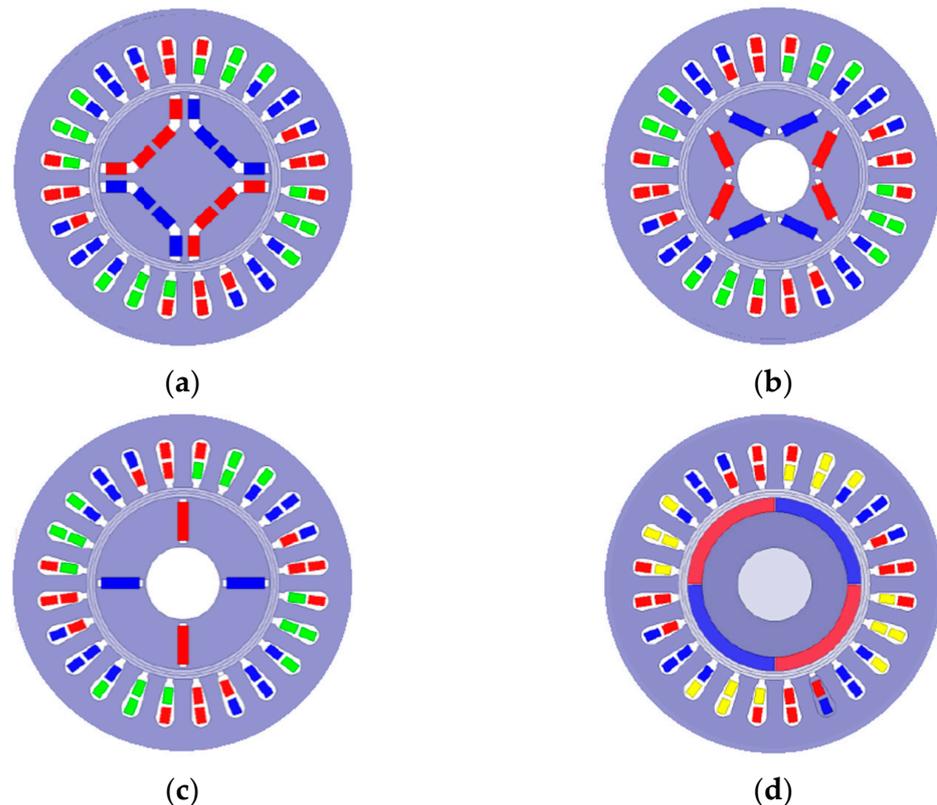


Figure 1. Rotor types of the general pattern. (a) U type; (b) V type; (c) radial type; (d) surface-mounted type.

2.2. Basic Data of the PMSM

PMSM has the characteristics of high power density, high stator loss density and small heat dissipation area. The high-speed rotating rotor is subjected to centrifugal force and, due to air friction and eddy current loss, it will also produce high temperature rise. To solve these problems, the structure of PMSM must be optimized.

In order to adapt to the design requirements of HPMSM, the following materials were used for the motor. A 0.1 mm ultrathin silicon steel sheet 10JNEX900 was applied in the design of the motor. Since the N38UH permanent magnet can withstand a maximum temperature of 180 °C and the coercivity of N38UH is 885 kA/m, this paper used N38UH to make rotor permanent magnet.

Figure 2a shows the basic design of the stator slot of the HPMSM, and the parameters of the slot were designed as shown in Table 1. Figure 2b shows the side view of the PMSM. The main design parameters of PMSM are shown in Table 1. In order to calculate the size of the ampere-conductor in the total electrical load, the limitation of the conductor current density in the stator slot should be determined, and the magnetic circuit calculation method should be adopted.

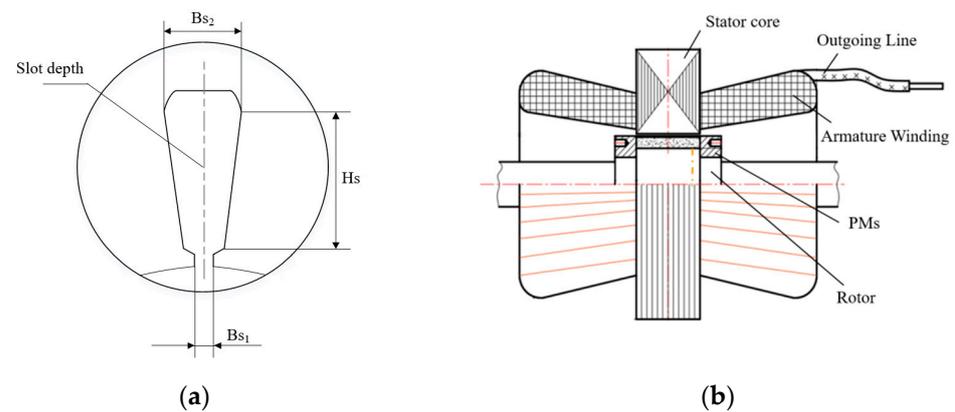


Figure 2. Configuration and mechanical structure of the HPMSM. (a) Stator slot. (b) Side view of the mechanical structure.

Table 1. Main design parameters of the PMSM.

Parameters	Values	Parameters	Values
Outer stator diameter	120 mm	Rated speed	100 krpm
Inner stator diameter	60 mm	Rated power	10 kW
Air gap length	2 mm	Rated voltage	AC 380 V
Axial length	70 mm	Rated current	20 A
PM thickness	4 mm	PM type	N38UH

The main parameters of stator core are slot opening width, slot width, tooth tip height and slot depth, and the main parameters of rotor are permanent magnet (PM) thickness, PM length (except for surface mounted type), magnetic gap length(except for surface-mounted type), angle of magnetic pole (only for surface-mounted type) and pole pair number. These were set as optimization variables and their initial value ranges are shown in Table 2. Based on the general rotor model and the stator model, the structure of permanent magnet synchronous motor can be designed by determining the parameters.

Table 2. Initial design parameters of the HPMSM.

Type	Design Parameters	Value Ranges
Stator	Slot opening width	0.5–2.0 mm
	Slot width	1–2 mm
	Tooth tip height	1–2 mm
	Slot depth	10–20 mm
Rotor	PM width	3–6 mm
	PM length (except for surface-mounted type)	3–10 mm
	Magnetic gap length (except for surface-mounted type)	0.5–2 mm
	Angle of magnetic pole (only for surface-mounted type)	0–3 deg
	Pole pair number	1 or 2

2.3. Database Generation

In this paper, by adjusting the structural parameters of four different types of motors, 500 groups of electromagnetic field distribution data of each type of motors were obtained using FEM solver. Although the calculation time of electromagnetic field distribution takes about 1 min in a single case, it will still take a huge amount of time to establish the whole training database if the parameter is adjusted manually. Therefore, Python scripts were created to automate FEM simulations of different types of HPMSM with different structural

parameters, and batch commands were used to realize parallel calculation process. The calculations covered in this article were all performed on a High Performance Computing (HPC) supercomputer platform.

Figure 3 shows the flow chart of HPMSM parameter database generation. First, the structural parameters were randomly combined using Python script, corresponding material properties and boundaries were added through input and excitation was assigned, and then FEM simulation was carried out. The calculated electromagnetic field distribution was an ultra-long sequence containing magnetic potential data of all nodes, which, together with the (x, y) coordinates of all nodes, constituted a sample. However, with the change of motor structure, the number of nodes changed, leading to different sequence lengths of different samples. If it was directly trained in the deep learning model, the program could not be read. Secondly, the electromagnetic field data generated using the finite element calculation is often multi-digit decimal, and the derived data may contain non-numeric lines, which makes it difficult to identify the program. Therefore, it was essential to adopt the corresponding method to preprocess the data. The preprocessing steps used in this paper included data cleaning and data transformation.

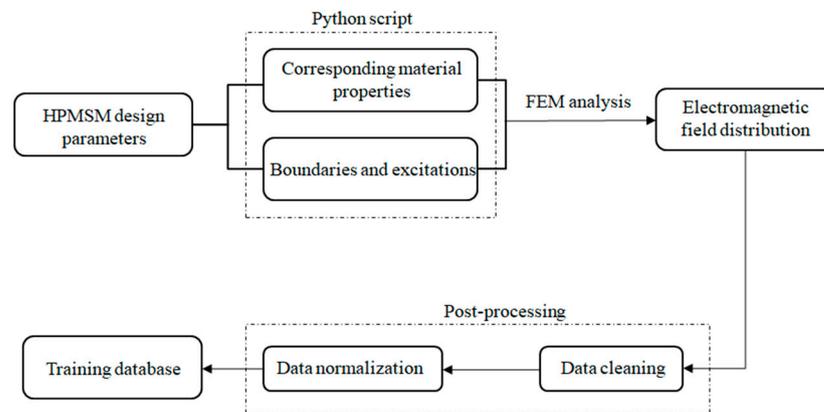


Figure 3. Flow chart of the database generation.

According to the range and equal step size of the above design variables, the four design variables were combined with each other in accordance with the principle of controlling the shift of a single variable to produce 600 groups of structural coordination, that is, 600 different motor structure schemes. Examples of the motor schemes are shown in Tables 3 and 4.

Table 3. Motor schemes of the U-type, V-type and Radial type.

Motor Scheme	Slot Opening Width	Slot Width	Tooth Tip Height	Slot Depth	PM Width	PM Length	Magnetic Gap Length	Pole Pair Number
Case1	0.5	1.0	1.0	10.0	3.0	3.0	0.5	1
Case2	0.5	1.0	1.0	10.0	3.0	3.0	0.7	1
Case3	0.5	1.0	1.0	10.0	3.0	4.0	0.5	1
Case4	0.5	1.0	1.0	10.0	4.0	3.0	0.5	1
Case5	0.5	1.0	1.0	12.0	3.0	3.0	0.5	1
...
Case597	1.9	1.8	2.0	18.0	5.0	9.0	1.9	2
Case598	1.9	2.0	1.8	18.0	5.0	9.0	1.9	2
Case599	2.0	1.8	1.8	18.0	5.0	9.0	1.9	2
Case600	2.0	2.0	2.0	20.0	6.0	10.0	2.0	2

Table 4. Motor schemes of the Surface mounted type.

Motor Scheme	Slot Opening Width	Slot Width	Tooth Tip Height	Slot Depth	PM Width	Angle of Magnetic Pole	Pole Pair Number
Case1	0.5	1.0	1.0	10.0	3.0	0	1
Case2	0.5	1.0	1.0	10.0	3.0	0	1
Case3	0.5	1.0	1.0	10.0	3.0	0	1
Case4	0.5	1.0	1.0	10.0	4.0	0	1
Case5	0.5	1.0	1.0	12.0	3.0	0	1
...
Case597	1.9	1.8	2.0	18.0	5.0	3	2
Case598	1.9	2.0	1.8	18.0	5.0	3	2
Case599	2.0	1.8	1.8	18.0	5.0	3	2
Case600	2.0	2.0	2.0	20.0	6.0	3	2

The values of variables were changed and finite element calculation was carried out. The magnetic potential values and node coordinates obtained from each calculation were derived as a sample. By repeating the above operations, a total of 600×4 samples was obtained to facilitate the training of the deep learning model.

3. Prediction Model and HPMSM Optimization

3.1. PNN Prediction Model

The PNN was composed of multilayer adaptive nonlinear elements, which continuously reduced the model prediction error through layer-to-layer learning. Compared with a traditional network, the PNN has a strong ability to process nonlinear data. By comparing the nonlinear expression of a large number of nonlinear functions with the real value, the model repeatedly updates the weight and offset values to explore the hidden internal features of the data that are difficult to find using manual calculation. Due to the introduction of nonlinear functions, the deep learning model has a very high degree of fitting for nonlinear problems. Therefore, it is widely used in nonlinear engineering problems. PNN has proved to be effective in regression modeling such as speech acoustics, bearing fault diagnosis and ship detection. In this paper, PNN was introduced to establish a fast calculation model for HPMSMs.

The preprocessed data set above was divided into a training set and a test set according to the ratio of 2:1, and the PNN model was established. The divided 400×4 groups of training set samples were then given to the PNN model for learning and training according to the prescribed format, and the 200×4 groups of test data set samples were input into the trained PNN network prediction model according to the prescribed format, and the prediction results were compared and analyzed.

3.1.1. PNN Structure

The network structure of the PNN algorithm includes a visible layer and a hidden layer. The visible layer includes the input layer and the output layer. The basic structure of the PNN is shown in Figure 4.

The idea behind this encoder–decoder structure was to convert low-dimensional input parameters into very high-dimensional spaces through the encoder to increase the number of degrees of freedom and to facilitate the handling of nonlinear input–output relationships associated with PMSM structures. The information was then submitted to a lower dimensional space using the decoder. This process can actually be thought of as a dimensional evolution between input and output parameters.

The PNN adopted in this paper had a multilevel feature and context pyramid network, in which each layer could be represented as the weight W multiplied by the vector x composed of neurons, plus the deviation b . We considered a regular grid of N elements, of which the value of parameter i said for $y_i \in [0, 1]$, and our goal was to optimize the parameter value $y_i \in [0, 1]$ in order to obtain an optimal structure.

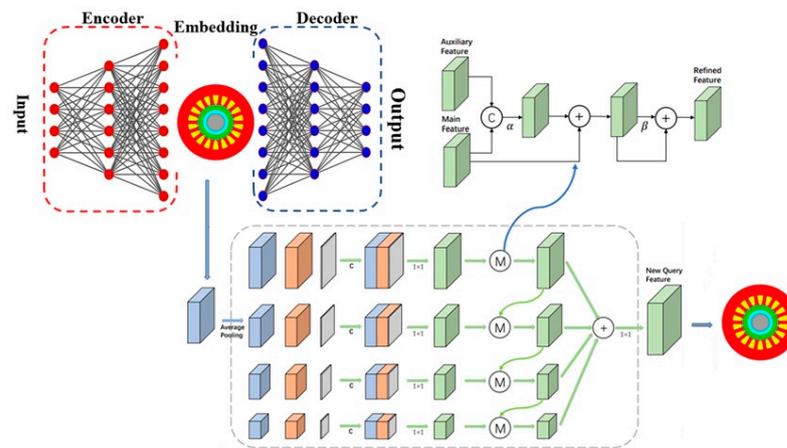


Figure 4. Framework diagram of the optimization method based on PNN.

In order to realize the mapping of the PNN model from the parameter space to an objective function, and to effectively model both simple linear mapping and highly nonlinear problems, activation functions were added to each layer of the PNN in this paper:

$$y_i = \sigma(x_i + \bar{b}(X)), i \in \{1, 2, \dots, N\} \tag{1}$$

$$\sum_{i=1}^N y_i = V_0 \tag{2}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

where $X = \{x_1, x_2, \dots, x_N\} \in R^N$, the functions σ makes sure the magnetic potential in $[0, 1]$, and the function $b(x)$ is set up to ensure that the PNN meets the volume constraint of the optimization problem.

This project intended to adopt a deep neural network with layer $L + 1$, and the number of neurons in each layer (n_0, n_1, \dots, n_L) and activation function $\mu:R \rightarrow R$. The activation function $a^l \in R^{n_l}$ and the pre-activation function $\tilde{a}^l \in R^{n_l}$ were defined recursively on all l layers, as shown in Equation (4).

$$\begin{aligned} a^0(x) &= x \\ \tilde{a}^{l+1}(x) &= \frac{\alpha}{\sqrt{n_l}} W^l(x) a^l(x) + \beta b^l \\ a^{l+1}(x) &= \mu(\tilde{a}^{l+1}(x)) \end{aligned} \tag{4}$$

In Equation (4), super parameters $\alpha, \beta \in [0, 1]$ indicate the contribution of the weight item and bias item, respectively. Parameter $\theta = (\theta_p)_p$ of the weight matrix W^l and the deviation vector b^l is an independent and equally distributed standard normal random variable $N(0, 1)$. The output of the network was $f_\theta(x) = \tilde{a}^l(x)$. In order to ensure that the variance of neurons at each layer was equal to 1 when initializing, we set α and β to satisfy $\alpha^2 + \beta^2 = 1$ and $E_{X \sim N(0,1)}[\mu(X)^2] = 1$.

In this paper, the loss function was defined as

$$\theta \mapsto C(Y^{NN}(\theta)) = C(\sum(X(\theta))) \tag{5}$$

Since the initial value in the optimization method was usually determined, but the neural network was initialized randomly, the initial density field in the deep learning-based optimization method was random and non-constant. To avoid this problem, instead of using the initial density field, we chose to add a constant, such as Equation (6), to calculate $X(\theta)$:

$$\begin{aligned}
 &\forall i \in \{1, \dots, N\} \forall i \in \{1, \dots, N\}, \\
 X_i(\theta) = \bar{f}_{\theta(t)}(z_i) &= f_{\theta(t)}(z_i) - f_{\theta(t=0)}(z_i) + \log\left(\frac{V_0}{N-V_0}\right)
 \end{aligned}
 \tag{6}$$

3.1.2. PNN Model Verification

In order to test the accuracy of the activation function, the determination coefficient R^2 , which is defined in Equation (7), was introduced. When the value of R^2 was between $[0, 1]$, the activation function was feasible, and the closer the value was to 1, the higher the prediction accuracy was. The value of R^2 in this paper was 0.993. This shows that the activation function in this paper had a high prediction accuracy and strong generalization ability, which provided a good basis for the following modeling and optimization.

$$\begin{cases} R^2 = 1 - \frac{E_{sse}}{E_{sst}} \\ E_{sse} = \sum_{i=1}^n (\hat{y}_i - y_i)^2 \\ E_{sst} = \sum_{i=1}^n (y_i - \bar{y})^2 \end{cases}
 \tag{7}$$

$$MSE = \frac{1}{N} \sum_{t=1}^N (f_t - y_t)^2
 \tag{8}$$

$$MAE = \frac{1}{N} \sum_{t=1}^N |f_t - y_t|
 \tag{9}$$

To detect the accuracy of the PNN network, the mean squared error and mean absolute error, which are defined in Equations (8) and (9), were introduced. The closer the values were to 0, the higher the accuracy was.

PNN was configured with 16 time block features and 8 space block features. The Adam optimizer [30] was used for model training, with a learning rate of 0.0001 and a batch size of 400. Each method was run five times and the average result was reported.

In Figure 5, the broken blue line represents the error curve of the total loss, and the broken red line represents the error curve of the output torque. With the increase of the number of iterations, the MAE values of both models processing the total loss and the output torque showed downward trends, and the change trend was basically the same. After the number of training runs reached 400, the MAE values tended to become stable and converge to a smaller value. Thus, in this paper the number of training data runs considered appropriate was 400.

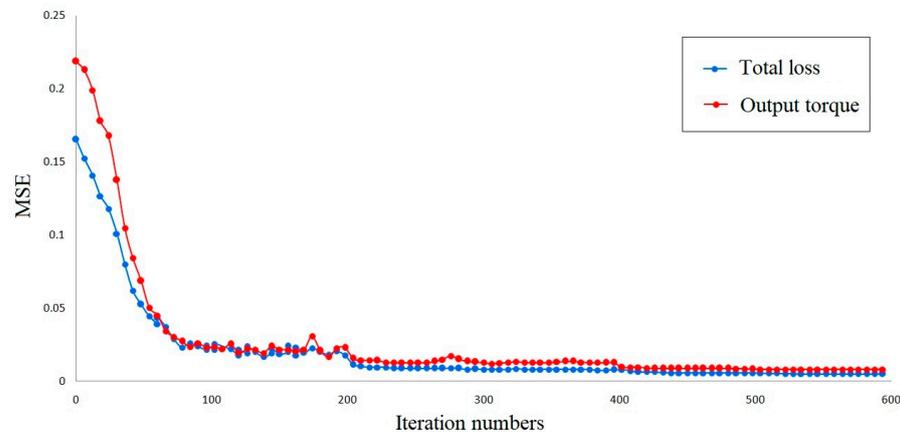


Figure 5. The diagram of the relationship between iteration numbers and MAE.

In this paper, 200 test samples were used to predict the PNN fast calculation model, and the accuracy of the model was verified. The accuracy verification of the PNN model is shown in Figure 6. Figure 6a,b represent MAE error graphs of test samples of PNN calculation models established in this paper for two objective functions.

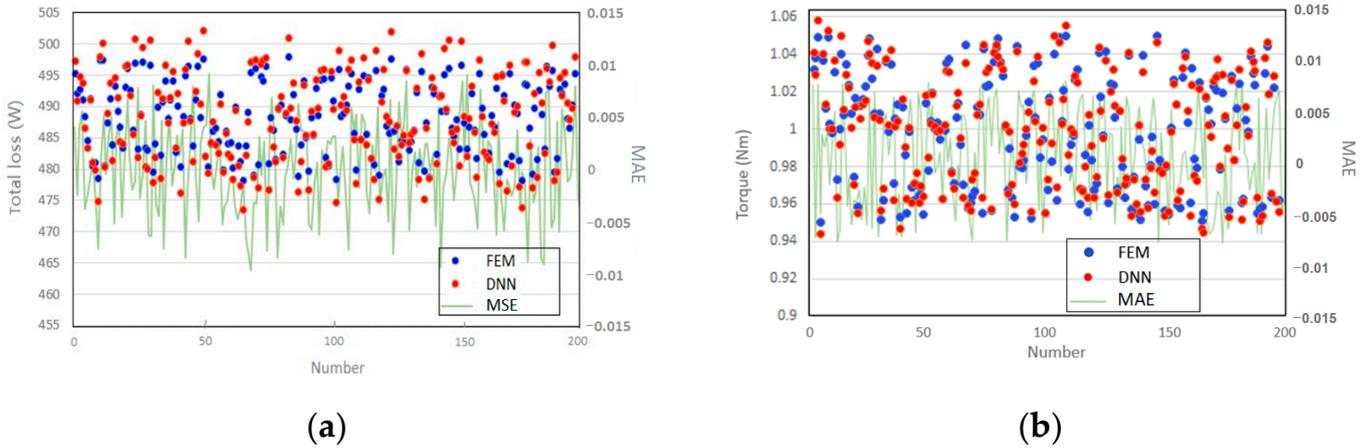


Figure 6. Accuracy verification of the PNN model. (a) Total loss. (b) Output torque.

From Figure 6a,b it could be seen that the PNN model had a high fitting accuracy for loss and torque of the HPMSM. According to the above analysis results, PNN had a strong adaptive ability and generalization ability. As an accurate and fast calculation model, the PNN model could provide support for subsequent optimization and effectively improved the efficiency of optimization.

3.2. Quantum Bat Algorithm

Iterating to find the optimal solution in the QBA can be likened to bats searching for prey and automatically changing the loudness and firing rate of their pulses according to the prey information. They make these changes temporarily at first, since in the search stage of exploring for prey a bat will emit loud ultrasonic pulses with a low pulse transmitting frequency. Once the target prey is locked, as the distance between the bat and the prey slowly decreases, the loudness of the emitted pulses will fade, and the emission rate of the pulses will increase.

In the QBA, the probability amplitude of acquit is directly used as the code of the current position of the bat. Considering the randomness of coding during population initialization, the coding scheme in (10) is adopted:

$$P_i = \begin{bmatrix} \cos \theta_{i1} & \cos \theta_{i2} & \dots & \cos \theta_{id} \\ \sin \theta_{i1} & \sin \theta_{i2} & \dots & \sin \theta_{id} \end{bmatrix} \quad (10)$$

where $\theta_{ij} = 2\pi \cdot rand$ and $rand$ are random numbers in $(0, 1)$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, d$; n is the population size; and d is the spatial dimension. Thus, each bat in the population occupies two positions in the traversal space, and their probability amplitudes corresponding to the quantum state $|0\rangle$ and $|1\rangle$ are

$$P_{ic} = (\cos \theta_{i1} \quad \cos \theta_{i2} \quad \dots \quad \cos \theta_{id}) \quad (11)$$

$$P_{is} = (\sin \theta_{i1} \quad \sin \theta_{i2} \quad \dots \quad \sin \theta_{id}) \quad (12)$$

where P_{ic} and P_{is} are the cosine and sine positions, respectively. Each probability amplitude of a qubit corresponds to an optimization variable in the solution space. Each bat corresponds to two locations in the optimization problem: the probability amplitude β_j^i of quantum states $|0\rangle$ responds to X_{jc}^i , while the probability amplitude η_j^i of quantum states $|1\rangle$ responds to X_{js}^i .

$$X_{jc}^i = (b_j(1 + \beta_j^i) + a_j(1 - \beta_j^i))/2 \tag{13}$$

$$X_{js}^i = (b_j(1 + \eta_j^i) + a_j(1 - \eta_j^i))/2 \tag{14}$$

where the value interval of the variable parameters of the problem is $[a_j, b_j]$, $i = 1, 2, \dots, n$, and $j = 1, 2, \dots, d$.

The position of the bat varies according to the quantum revolving gate. Therefore, the update of the quantum revolving gate angle is used to replace the update of the bat's motion speed in the ordinary bat algorithm (BA), and the update of the quantum bit probability amplitude is used to replace the update of the bat's position.

Suppose that the optimal position searched by the whole population is

$$P_g = (\cos \theta_{g1}, \cos \theta_{g2}, \dots, \cos \theta_{gd}) \tag{15}$$

Then, the bat status update rules are as follows:

① In the global search of bat P_i , the update rule of the qubit amplitude angle increment is

$$\Delta\theta_{ij}(t + 1) = \Delta\theta_{ij}(t) + F(i)(\Delta\theta_{ij}) \tag{16}$$

$$\Delta\theta_g = \begin{cases} 2\pi + \theta_{gj} - \theta_{ij}, & \theta_{gj} - \theta_{ij} < -\pi \\ \theta_{gj} - \theta_{ij}, & -\pi \leq \theta_{gj} - \theta_{ij} \leq \pi \\ \theta_{gj} - \theta_{ij} - 2\pi, & \theta_{gj} - \theta_{ij} > \pi \end{cases} \tag{17}$$

② In the local search of bat P_i , the update rule of the qubit amplitude angle increment is

$$\Delta\theta_{ij}(t + 1) = e^{-(\tau \cdot t / \max_{gen}) \cdot \text{mean}(A) \delta} \tag{18}$$

where τ is a constant, t refers to the current iteration number, \max_{gen} is the maximum iteration number, $\text{mean}(A)$ is the average loudness of each bat and δ is the expected criterion.

③ The updated position of the bat is

$$\tilde{P}_{ic} = (\cos(\theta_{i1}(t) + \Delta \cos \theta_{i1}(t + 1)), \dots, \cos(\theta_{id}(t) + \Delta \theta_{id}(t + 1))) \tag{19}$$

$$\tilde{P}_{is} = (\sin(\theta_{i1}(t) + \Delta \cos \theta_{i1}(t + 1)), \dots, \sin(\theta_{id}(t) + \Delta \theta_{id}(t + 1))) \tag{20}$$

In order to avoid the algorithm falling into local optimization, and to improve the diversity of the population, a quantum non-revolving gate was introduced to perform the mutation operation. Let the mutation probability be p_m , and let each bat set a random number $rand$ in the range(0,1). If $rand < p_m$, the quantum non-revolving gate would be used to exchange the two probability amplitudes of the bat, while its rotation vector would remain unchanged. The flow chart of the QBA is shown in Figure 7.

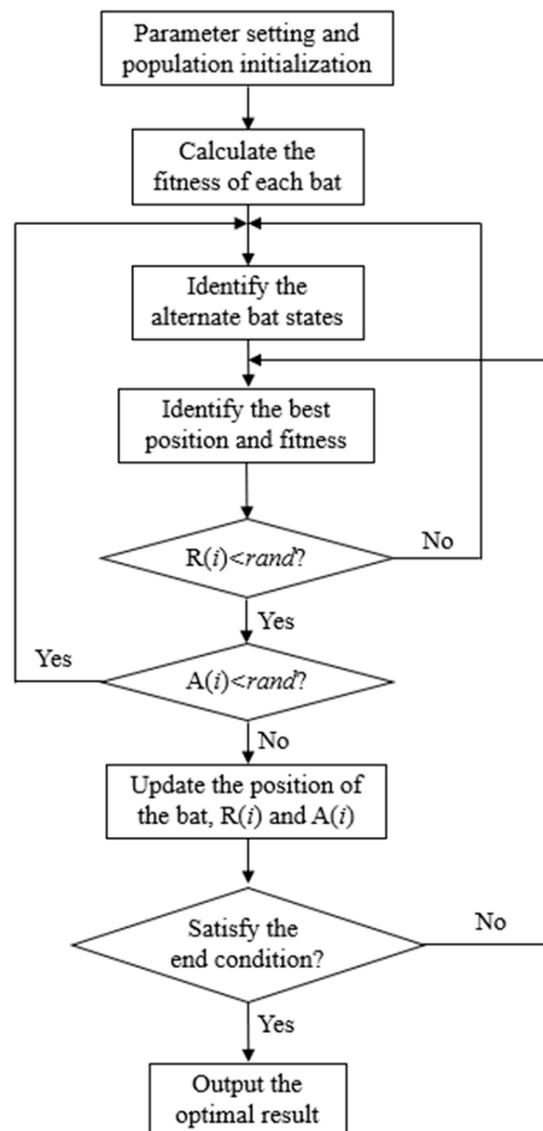


Figure 7. Flow chart of the QBA.

3.3. Optimization Results and Analysis

The setting of model parameters has a great influence on prediction accuracy. The parameters of the QBA in this paper were set as follows: the initial population was set to 200; the maximum number of iterations \max_{gen} was set to 1000; the maximum and minimum values of the emission frequency were set to 1 and -1 , respectively; and the expected criterion δ was set to 0.01.

QBA method based on PNN model was used to optimize PMSM. In four type of PMSM, the surface mounted type has the best performance after optimization. Its evolution is shown in Figure 8. In Figure 8a, the total loss decreases with the increase of the number of iterations and becomes stable after 200 iterations. In Figure 8b, the output torque increases with the increase of the number of iterations and becomes stable after 100 iterations. It shows that the convergence of the model is good and the optimal solution of the algorithm has been reached at the maximum number of iterations. The optimal solution obtained after multiple iteration optimization is shown in Table 5.

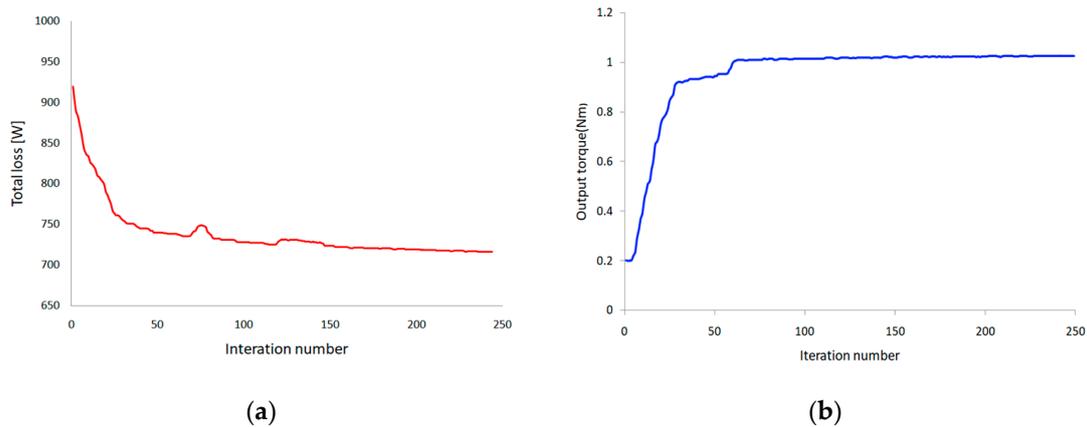


Figure 8. Evolution algebra of PNN-based QBA (a) Total loss. (b) Output torque.

Table 5. Optimization results.

Parameters	Values
Pole pair numbers	1
Number of slots	24
Overall diameter (mm)	200
Outer diameter of stator(mm)	130
Inner diameter of stator (mm)	56
Outer diameter of rotor (mm)	53
Tooth width(mm)	3.6
Slot depth (mm)	18
Width of slot opening (mm)	1.8
Length of stator core (mm)	40
Thickness of PMs (mm)	7
Angle of magnetic pole (rad)	0

When the PMSM was working at the rated working point, in which the rated speed was 100 krpm, as shown in Figure 9, it could be seen from the magnetic density cloud map that the magnetic density amplitude of the optimized structure was reduced to 1.2948 T compared with 1.6412 T before optimization. As shown in Figure 10, the average output torque was 0.986 N·M before optimization and increased to 1.026 N·M after optimization. According to Table 6, after optimization, various loss values decreased significantly compared with those before optimization, with the total loss reduced from 938.1 W before optimization to 747.8 W after optimization. The waveform of the back electromotive force (EMF) of the optimized motor is shown in Figure 11. The effective value of the back EMF was 201.9 V, and the waveform had a good sinusoidal property.

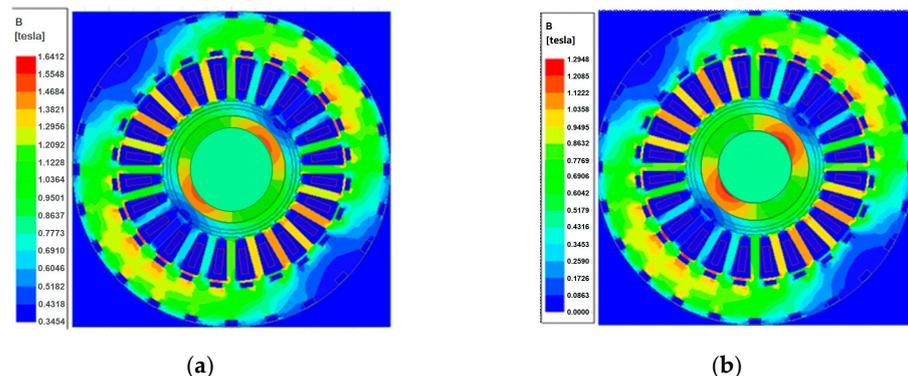


Figure 9. Magnetic dense cloud map of the HPMSM. (a) Before optimization (b) After optimization.

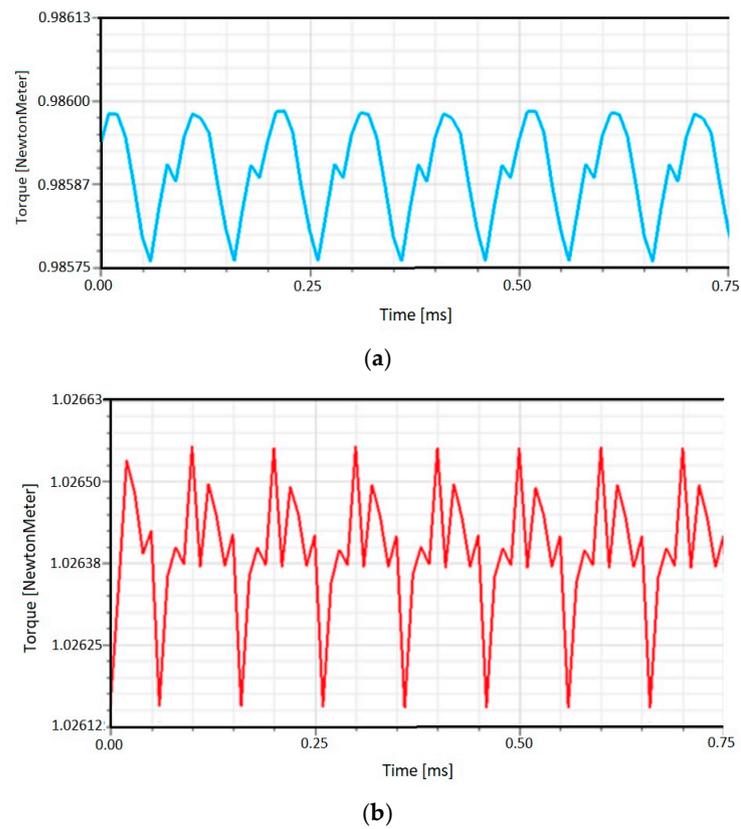


Figure 10. Output torque of the HPMSM. (a) Before optimization (b) After optimization.

Table 6. Comparison of loss value of HPMSM.

Parameters	Stator Core Loss	Copper Loss	Eddy Current Loss	Wind Friction Loss	Stray Loss
Before optimization	483	53.1	44	311	47
After optimization	345	49.8	31	300	22

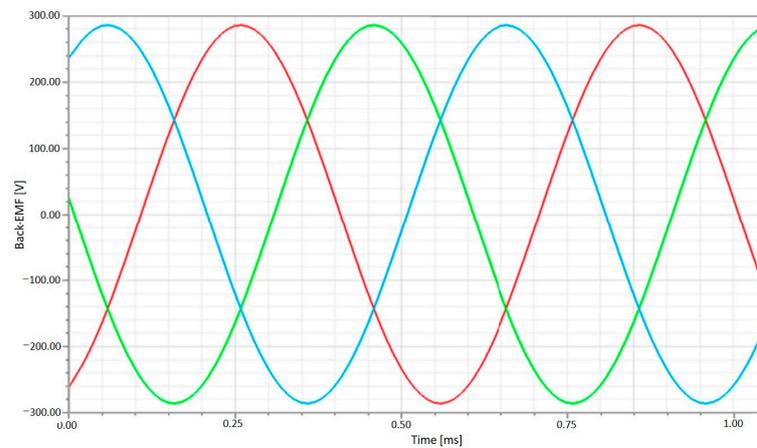


Figure 11. No load back electromotive force.

According to the comparison of the magnetic dense cloud map, output torque, loss data and the waveform of the back EME, it could be seen that the QBA optimization method adopted in this paper based on the PNN model could significantly improve the performance of HPMSMs and achieved the purpose of the optimization design.

4. Experimental Verification

According to the motor's structural parameters optimized using the PNN model, the HPMSM prototype was made and the test platform was built, as shown in Figure 12. The assembled stator and rotor are shown in Figure 12a,b. A servomotor was used to provide the load torque, and the tested waveforms are displayed on an oscilloscope.

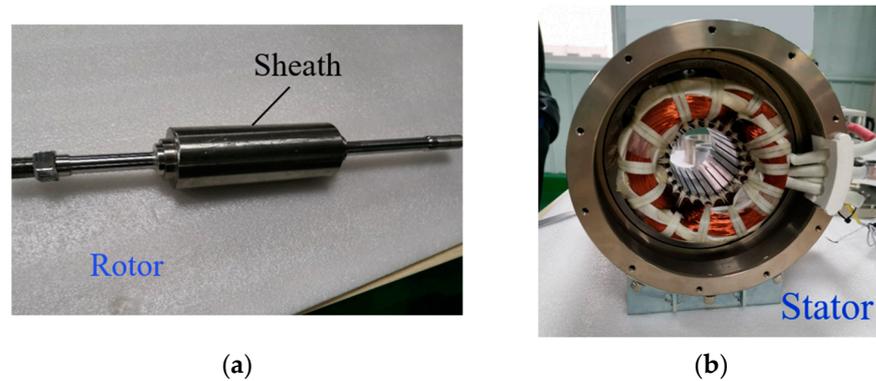


Figure 12. Structure of the prototype. (a) Rotor. (b) Stator.

Electromagnetic torque is an important index of motor performance. In order to compare the simulation results with the experimental results, the torque under a certain speed and load current was tested, and the results are shown in Table 7. By comparing the calculated torque with the measured torque, it could be seen that the error of the simulation result was less than 5.5%, which was acceptable.

Table 7. Comparison of electromagnetic torque under different phase current.

Speed (krpm)	Current (A)	Simulation Results of Torque (N·M)	Experiment Results of Torque (N·M)
20	20.4	1.055	1.1
30	20	1.026	1.0
40	20	1.026	1.0
50	20.5	1.055	1.1
60	20.5	1.055	1.1

5. Conclusions

This paper systematically studied the QBA optimization method combined with a PNN model for the structure optimization problem of HPMSMs. The PNN models for the objective functions of the loss and torque were proven to be sufficiently accurate to replace the FEM simulation. Instead of calculating an FEM simulation for each particle, applying PNN models could shorten the working time of the optimization process.

In this paper, a fast prediction model of electromagnetic field distribution based on PNN was studied. Based on the rapid predictive neural network model, the structure of permanent magnet synchronous motor was optimized globally, and the optimal motor structure parameters were obtained, so that the designed permanent magnet synchronous motor could achieve a good performance. The correctness of the model was verified using a finite element simulation and prototype test.

The research content and conclusion of this paper are as follows:

- (1) An efficient modeling method based on PNN deep learning was studied. This method constructed the mapping relationship between input structure parameters and the performance of the PMSM. It was able to solve the problem of rapid electromagnetic field prediction in the optimal design of PMSM.

- (2) The QBA optimization algorithm was used to determine the optimal structural parameter combination of permanent magnet synchronous motors and obtain the best performance.
- (3) The experimental results of the prototype showed that this method is effective and advanced for the structural optimization design of HPMSM.

Author Contributions: Conceptualization, X.L.; methodology, X.L.; software, W.P.; validation, L.X., W.P. and X.Z.; formal analysis, X.L.; investigation, X.Z.; resources, W.P.; data curation, L.X.; writing—original draft preparation, X.L.; writing—review and editing, X.L.; visualization, W.P. and X.L.; project administration, X.L.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Research Program of Chongqing Municipal Education Commission under grant number KJQN202200543.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are not publicly available due to [3rd Party Data Restrictions apply to the availability of these data].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ahn, J.-H.; Han, C.; Kim, C.-W.; Choi, J.-Y. Rotor Design of High-Speed Permanent Magnet Synchronous Motors Considering Rotor Magnet and Sleeve Materials. *IEEE Trans. Appl. Supercond.* **2017**, *28*, 5201504. [[CrossRef](#)]
2. Fang, S.; Liu, H.; Wang, H.; Yang, H.; Lin, H. High Power Density PMSM With Lightweight Structure and High-Performance Soft Magnetic Alloy Core. *IEEE Trans. Appl. Supercond.* **2019**, *29*, 0602805. [[CrossRef](#)]
3. Shin, K.H.; Bang, T.K.; Cho, H.W.; Choi, J.Y. Design and Analysis of High-Speed Permanent Magnet Synchronous Generator With Rotor Structure Considering Electromechanical Characteristics. *IEEE Trans. Appl. Supercond.* **2020**, *30*, 5204305. [[CrossRef](#)]
4. Bailey, C.; Saban, D.M.; Guedes-Pinto, P. Design of High-Speed Direct-Connected Permanent-Magnet Motors and Generators for the Petrochemical Industry. *IEEE Trans. Ind. Appl.* **2009**, *45*, 1159–1165. [[CrossRef](#)]
5. Chen, Y.; Zhou, J.; Fang, Y.; Gao, Y.; Xia, Y. Multi-field coupling finite-element analysis of the temperature rise in permanent magnet synchronous motor applied for high speed train. In Proceedings of the 19th International Conference on Electrical Machines and Systems (ICEMS), Chiba, Japan, 13–16 November 2016.
6. Cao, H.; Kou, B.; Zhang, D.; Li, W.; Zhang, X. Research on loss of high speed permanent magnet synchronous motor for flywheel energy storage. In Proceedings of the 16th International Symposium on Electromagnetic Launch Technology, Beijing, China, 15–19 May 2012; pp. 1–6. [[CrossRef](#)]
7. Barcaro, M.; Bianchi, N.; Magnussen, F. Permanent-Magnet Optimization in Permanent-Magnet-Assisted Synchronous Reluctance Motor for a Wide Constant-Power Speed Range. *IEEE Trans. Ind. Electron.* **2011**, *59*, 2495–2502. [[CrossRef](#)]
8. Jia, Y.-W.; Chen, X.-T.; Yao, C.-B.; Li, X. CEO election optimization algorithm and its application in constrained optimization problem. *Soft Comput.* **2023**, *27*, 7363–7400. [[CrossRef](#)]
9. Çelik, E. A powerful variant of symbiotic organisms search algorithm for global optimization. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103294. [[CrossRef](#)]
10. Çelik, E.; Öztürk, N.; Arya, Y. Advancement of the search process of salp swarm algorithm for global optimization problems. *Expert Syst. Appl.* **2021**, *182*, 115292. [[CrossRef](#)]
11. Çelik, E. IEGQO-AOA: Information-Exchanged Gaussian Arithmetic Optimization Algorithm with Quasi-opposition learning. *Knowl.-Based Syst.* **2023**, *60*, 110169. [[CrossRef](#)]
12. Kaplan, O.; Elik, E. Simplified model and genetic algorithm based simulated annealing approach for excitation current estimation of synchronous motor. *Adv. Electr. Comput. Eng.* **2018**, *18*, 75–84. [[CrossRef](#)]
13. Ho, S.-L.; Yang, S.; Ni, P.; Huang, J. A Quantum-inspired evolutionary algorithm for multi-objective design. *IEEE Trans. Magn.* **2013**, *49*, 1609–1612. [[CrossRef](#)]
14. Li, Y.; Yang, S.; Ren, Z. A methodology based on quantum evolutionary algorithm for topology optimization of electromagnetic devices. *IEEE Trans. Magn.* **2019**, *55*, 7204004. [[CrossRef](#)]
15. ul Hassan, H.T.; Asghar, M.U.; Zamir, M.Z.; Faiz, H.M.A. Economic load dispatch using novel bat algorithm with quantum and mechanical behavior. In Proceedings of the 2017 International Symposium on Wireless Systems and Networks (ISWSN), Lahore, Pakistan, 19–22 November 2017.
16. Nguyen, V.V.; Hartmann, D.; König, M. A distributed agent-based approach for simulation-based optimization. *Adv. Eng. Inform.* **2012**, *26*, 814–832. [[CrossRef](#)]

17. Chelladurai, S.J.S.; Murugan, K.; Ray, A.P.; Upadhyaya, M.; Narasimharaj, V.; Gnanasekaran, S. Optimization of process parameters using response surface methodology: A review. *Mater. Today Proc.* **2021**, *37*, 1301–1304. [[CrossRef](#)]
18. Zhao, W.; Ma, A.; Ji, J.; Chen, X.; Yao, T. Multi objective optimization of a double-side linear Vernier PM motor using response surface method and differential evolution. *IEEE Trans. Ind. Electron.* **2019**, *67*, 80–90. [[CrossRef](#)]
19. Çelik, E.; Çavuşoğlu, O.; Gürün, H.; Öztürk, N. Estimation of the Clearance Effect in the Blanking Process of CuZn30 Sheet Metal Using Neural Network—A Comparative Study. *Bilişim Teknol. Derg.* **2018**, *11*, 187–193. [[CrossRef](#)]
20. Çelik, E.; Uzun, Y.; Kurt, E.; Öztürk, N.; Topaloğlu, N. A neural network design for the estimation of nonlinear behavior of a magnetically-excited piezoelectric harvester. *J. Electron. Mater.* **2018**, *47*, 4412–4420. [[CrossRef](#)]
21. Çelik, E.; Gör, H.; Öztürk, N.; Kurt, E. Application of artificial neural network to estimate power generation and efficiency of a new axial flux permanent magnet synchronous generator. *Int. J. Hydrog. Energy* **2017**, *42*, 17692–17699. [[CrossRef](#)]
22. Erraissi, A.; Belangour, A. Data sources and ingestion big data layers: Meta-modeling of key concepts and features. *Int. J. Eng. Technol.* **2018**, *7*, 3607–3612.
23. Khan, A.; Ghorbanian, V.; Lowther, D. Deep learning for magnetic field estimation. *IEEE Trans. Magn.* **2019**, *55*, 7202304. [[CrossRef](#)]
24. Khan, A.; Mohammadi, M.H.; Ghorbanian, V.; Lowther, D. Efficiency Map Prediction of Motor Drives Using Deep Learning. *IEEE Trans. Magn.* **2020**, *56*, 7511504. [[CrossRef](#)]
25. Puzirev, V. Deep learning electromagnetic inversion with convolutional neural networks. *Geophys. J. Int.* **2019**, *218*, 817–832. [[CrossRef](#)]
26. Canziani, A.; Paszke, A.; Culurciello, E. An analysis of deep neural network models for practical applications. *arXiv* **2016**, arXiv:1605.07678.
27. Zhang, H.; Shao, J.; Salakhutdinov, R. Deep neural networks with multi-branch architectures are intrinsically less non-convex. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, Naha, Okinawa, Japan, 16–18 April 2019; pp. 1099–1109.
28. Ou, H.; Qing, C.; Xu, X.; Jin, J. Multi-level context pyramid network for visual sentiment analysis. *Sensors* **2021**, *21*, 2136. [[CrossRef](#)] [[PubMed](#)]
29. Ma, J.; Dai, Y.; Tan, Y.-P. Atrous convolutions spatial pyramid network for crowd counting and density estimation. *Neurocomputing* **2019**, *350*, 91–101. [[CrossRef](#)]
30. He, Q.; Dong, Z.; Chen, F.; Deng, S.; Liang, W.; Yang, Y. Pyramid: Enabling Hierarchical Neural Networks with Edge Computing. In Proceedings of the ACM Web Conference 2022, Virtual, 25–29 April 2022. [[CrossRef](#)]
31. Liu, X.; Lin, Q.; Fu, W. Optimal design of permanent magnet arrangement in synchronous motors. *Energies* **2017**, *10*, 1700. [[CrossRef](#)]
32. Liu, X.; Fu, W.N.; Niu, S. Optimal structure design of permanent magnet motors based on a general pattern of rotor topologies. *IEEE Trans. Magn.* **2017**, *53*, 1–4. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.