New Optimization Design Method for a Double Secondary Linear Motor Based on R-DNN Modeling Method and MCS Optimization Algorithm

Weitao Wang1, Jiwen Zhao2*, Yang Zhou1 and Fei Dong1
(1. School of Electrical Engineering and Automation, Anhui University, Hefei 230601, China; 2. School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009, China)

Abstract: Traditional linear motor optimization methods typically use analytical models combined with intelligent optimization algorithms. However, this approach has disadvantages, e.g., the analytical model might not be accurate enough, and the intelligent optimization algorithm can easily fall into local optimization. A new linear motor optimization strategy combining an R-deep neural network (R-DNN) and modified cuckoo search (MCS) is proposed; additionally, the thrust lifting and thrust fluctuation reductions are regarded as optimization objectives. The R-DNN is a deep neural network modeling method using the rectified linear unit (RELU) activation function, and the MCS provides a faster convergence speed and stronger data search capability as compared with genetic algorithms, particle swarm optimization, and standard CS algorithms. Finally, the validity and accuracy of this work are proven based on prototype experiments.

Keywords: Double secondary linear motor (DSLM), machine learning modeling, R-deep neural network (R-DNN) algorithm, intelligent optimization algorithm, modified cuckoo search (MCS) algorithm

1 Introduction

Linear motors can convert electrical energy into mechanical energy for linear motion. They have advantages in regards to their small size, rapid dynamic response, and high positioning accuracy. Therefore, they are widely used in industrial automation [1-5].

Linear motor parameter optimization is important in the research of linear motors. The traditional parameter optimization method for a linear motor is usually an analytical model combined with an intelligent optimization algorithm. In Ref. [6], an analytical model is established based on the equivalent circuit method, and then a genetic algorithm (GA) is used to search for the optimal design parameters of the motor. The experimental results show that the optimized parameters correspond to an increase of thrust and a decrease of thrust fluctuation. In Ref. [7], the thrust and efficiency are specified as objective functions, and then a pareto-based multi-objective differential evolutionary (DE) algorithm is adopted to search for the optimal size parameters of the motor, ultimately obtaining satisfactory thrust performance improvement results. In Ref. [8], a performance analysis model for a linear motor is established based on an equivalent magnetization method. Then, a particle swarm optimization (PSO) algorithm is used for an iterative calculation of the objective functions, to determine the optimal combination of structural parameters under multiple working conditions.

These parameter optimization methods (based on analytical models and optimization algorithms) have important reference value for the study of linear motor. Analytical model can help researchers understand the mutual restriction and connection between the internal parameters of motor, and has important reference value for the qualitative analysis of motor, intelligent optimization algorithm can accurately locate the optimal structural parameters of the motor [9]. However, there are many idealized processes in the analytical
model, resulting in a slightly lower analytical accuracy, and the general optimization algorithms have difficulty escaping from local optimal values [10].

In this study, a deep neural network modelling method using a rectified linear unit (RELU) activation function is investigated, to build a more accurate agent model for a double secondary linear motor (DSLM). To determine the optimal structure parameters of the linear motor, we introduce a novel and efficient global cuckoo search (CS) optimization algorithm, and improve it. Compared with the standard CS algorithm and two other common optimization algorithms (PSO and GA), the modified CS (MCS) algorithm searches for the global optimal value more quickly and accurately. Finally, the effectiveness of the proposed optimization design method is verified, based on experiments.

2 Structure of DSLM

The topological structure of an air-core DSLM as applied to the servo system of a microsecond laser engraving machine is shown in Fig. 1. It comprises two secondary back-irons, six rectangular coils, and periodic permanent magnets. The slot-pole combination of the DSLM is 7P/12S. The DSLM has no cogging effect or core loss, owing to its coreless and symmetrical structure.

![image](image-url)

Fig. 1 Structure of the DSLM

The optimization design problem for the motor should include the optimization goal, i.e., the variable to be optimized, and the boundary of the variable [11]. A high output thrust and low thrust fluctuations are important factors affecting the application of a DSLM in microsecond laser engraving machines. Therefore, this study selects thrust and thrust fluctuations as optimization goals. Considering the cost constraints, processing technology, previous experience, and dimensional constraints, the permanent magnet width, permanent magnet thickness, polar distance, air gap, and coil width are selected as the key parameters to be optimized, and their boundaries are shown in Tab. 1. The layout of the key parameters in the DSLM topology is shown in Fig. 2.

| Variables                | Boundary |
|-------------------------|----------|
| Polar distance $r/m$    | 18-21    |
| Magnet thickness $h/m$  | 3-6      |
| Magnet width $r/m$      | 13-16    |
| Air gap height $d/m$    | 9-12     |
| Coil width $d/m$        | 5-8      |

![image](image-url)

Fig. 2 Layout of key parameters in DSLM

When analyzing the magnetic field of the DSLM, the motor’s magnetic field is often simplified as a two-dimensional magnetic field, lateral end effects are often not considered, the demagnetization curve of the magnet is assumed to be a straight line, and the permeability of the secondary core is assumed to be infinite [12-13]. Based on these assumptions and idealizations, the magnetic field of the motor can be calculated using Eq. (1).

\[
B = \sum_{n=1}^{\infty} \frac{(-1)^{n+1} 4B_{r} \sinh(m_{n}h) \times \sin(m_{n}x) \cosh(m_{n}y) \sin\left(\frac{m_{n}r}{2}\right)}{\tau m_{n} \sinh\left[m_{n}\left(\frac{\delta}{2} + h\right)\right]} \tag{1}
\]

where $B_{r}$ is the residual magnetization, and $m_{n}$ is calculated using Eq. (2), as follows

\[
m_{n} = \frac{(2n - 1)\pi}{\tau} \tag{2}
\]

where $n$ is a positive integer.

The back electromotive force (EMF) of phase $a$ can be calculated using Eq. (3).

\[
E_{a} = E_{d} - E_{c} = 4Nlf \tau \left( \int_{r}^{\pi+d} Bdx \right) - \int_{r}^{\pi+d} Bdx = \frac{64}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^{n} Nlf \tau B_{r} \sinh(hm_{n}) \times \sin(m_{n}x) \sin\left(\frac{m_{n}r}{2}\right)}{d \tau m_{n}^{2} \sinh\left[m_{n}\left(\frac{\delta}{2} + h\right)\right]} \times \cos\left[\frac{2m_{n}x + m_{n}d + m_{n}w}{2}\right] \tag{3}
\]
where $E_A$ is the EMF of coil A; $E_X$ is the EMF of coil X; $L$ is the effective length of the conductor; $N$ is the number of conductors contained in each coil; $f$ is the frequency of the current; and $w$ is the distance between two adjacent coils.

Fig. 3 shows the finite element analysis (FEA) model of the DSLM, as established by Ansoft Maxwell. The desktop computer configuration parameters used in the simulation are as follows: CPU, Intel(R) Xeon(R) E5-2680; computer main frequency, 3.20 GHz; RAM, 64 GB.

Fig. 3  FEA model of the DSLM

The back-EMF values as calculated by the FEA and analytical models are compared in Fig. 4.

Fig. 4  Waveform of the back-EMF

The comparison results in Fig. 4 show that some assumptions and idealizations reduce the prediction accuracy of the analytical models. As a result, the analytical models are more suitable for qualitative analysis than for precise quantitative analysis [14]. Thus, it is crucial to obtain a more accurate motor model. In this study, a modeling algorithm called R-deep neural network (DNN) is used to replace the analytical model.

3  R-DNN modeling

3.1 Create sample library

The selected design variables can form $4^5 = 1024$ combinations, according to an orthogonal data combination method. The FEA method is used to calculate the thrust performance corresponding to each parameter combination. Then, a sample library can be obtained, as shown in Tab. 2.

| NO. | $\tau$ | $h$ | $\tau_c$ | $\delta$ | $d$ | $F/N$ | $F_{\text{ripple}}$% |
|-----|--------|------|---------|---------|-----|------|-----------------|
| 1   | 18     | 3    | 13      | 9       | 5   | 57.347 | 4.527           |
| 2   | 18     | 3    | 13      | 9       | 6   | 59.081 | 3.028           |
| 3   | 18     | 3    | 13      | 9       | 7   | 62.438 | 2.724           |
| 4   | 18     | 3    | 13      | 9       | 8   | 60.485 | 4.318           |
|   | | | | | | | |
| 1021| 21     | 6    | 16      | 12      | 5   | 52.797 | 5.768           |
| 1022| 21     | 6    | 16      | 12      | 6   | 54.014 | 4.373           |
| 1023| 21     | 6    | 16      | 12      | 7   | 56.308 | 3.574           |
| 1024| 21     | 6    | 16      | 12      | 8   | 53.215 | 7.606           |

3.2 R-DNN regression modeling

A DNN is a powerful neural network model [15]. It is composed of multi-layer adaptive nonlinear elements, and has been effectively used in wind speed prediction [16], fault diagnosis [17] and image processing [18].

In this study, an enhanced DNN algorithm, the R-DNN is introduced to train a DSLM model with satisfactory accuracy. The main steps for establishing the DSLM model with the R-DNN can be summarized as follows.

1. Two sets of data are divided from the sample...
library. One is used to train the R-DNN model, the other is used to test its accuracy.

(2) The number of hidden layers and number of neurons in each layer are set. RELU is selected as the activation function, and the output of i-th layer can be expressed as follows

\[ y_i = \sigma(x_i) = \sigma(W_i y_{i-1} + b_i) \] (4)

where \( W_i \) and \( b_i \) are the weight and bias between the \( i \) and \( i-1 \) layers, respectively, and \( \sigma(x) \) is the activation function, which is defined as follows

\[ \sigma(x) = \max\{0, x\} \] (5)

(3) The R-DNN model is trained by an error back propagation algorithm (BP algorithm), \( W_i \) and \( b_i \) are updated by the Stochastic gradient descent method.

\[ (W_i, b_i) - \varepsilon \frac{\partial D}{\partial (W_i, b_i)} \rightarrow (W_{i+1}, b_{i+1}) \] (6)

where \( \varepsilon \) is the learning rate, and \( D \) is the cost function, usually the mean squared error (MSE).

\[ MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \] (7)

where \( \hat{y}_i \) is the predicted value, and \( y_i \) is the real value.

(4) The model is regularized by dropout to avoid over-fitting.

3.3 Model accuracy test

Another well-known machine learning modeling method called k-nearest neighbors (KNN) is used in comparative experiments, to prove the superiority of R-DNN modeling method. The accuracy test results for thrust and thrust fluctuation are shown in Fig. 6 and Fig. 7, respectively. This test is performed in Matlab.

An evaluation index called \( R^2 \) [19] is often used to evaluate the fitting accuracy of machine learning algorithms. The closer \( R^2 \) is to 1, the higher the accuracy of the corresponding model. The calculation formula of \( R^2 \) is as follows

\[
R^2 = 1 - \frac{\text{E}_{\text{tot}}}{\text{E}_{\text{sse}}} = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \]

where \( \bar{y} \) is the average of the real value, \( n \) is the number of test samples, \( \hat{y}_i \) is the predicted value, \( E_{\text{tot}} \) is the sum of the total dispersion squares, and \( E_{\text{sse}} \) is the residual sum of squares.

Fig. 8 shows an accuracy comparison between the R-DNN and KNN models. The results show that the fitting accuracy of the R-DDN model is higher than that of the (famous) KNN, and can meet the requirements for subsequent optimization design.

4 Modified cuckoo search algorithm

To achieve the optimization goal of this study, i.e., to improve the thrust and reduce the thrust fluctuations of a DSLM, a powerful global optimization algorithm
(the MCS algorithm) is proposed for optimizing the motor’s structural parameters.

A CS is a new heuristic optimization algorithm that imitates the parasitic breeding strategy of a cuckoo population as enhanced by Levy flight, rather than using a simple isotropic random walk. Some research work has been conducted to show that the CS algorithm is more effective than DE algorithms, GAs, and others [20-21]. The CS is simple and efficient, and has an excellent random search path. It has been successfully applied in engineering optimization, image processing, artificial intelligence, and other fields [22-25].

However, as a bionic algorithm based on population behavior, the CS algorithm still has the disadvantages in regards to a slow search rate, and difficulty in jumping out of a local optimum [26]. Considering these shortcomings, we modified the standard CS algorithm to purpose the MCS. In the MCS, the exploration and mining ability of CS algorithm is enhanced by introducing a non-linear decreasing inertia weight to modify the population renewal formula. The convergence speed of CS algorithm is improved by introducing Gauss disturbances to increase the change vitality of nest positions.

In the MSC algorithm, there is one egg per nest, and the cuckoo’s egg represents a new solution. The goal is to use new and potentially better solutions to replace worse solutions. Three rules are specified in the MCS algorithm, as follows [27].

1. The cuckoo lays only one egg at a time, representing a solution to the problem, and randomly places the eggs in a nest for incubation.

2. A part of the nests contains good-quality eggs, i.e., good solutions to problems. These nests will be retained for the next generation.

3. The total number of nests is certain. The probability of an alien egg being found is \( P_a \) (\( P_a \in [0,1] \)).

The flow chart of the MCS algorithm is shown in Fig. 9.

The main steps are as follows.

Step 1: Set the number of nests as \( n \), the dimensionality of the search space as \( d \), initialize the nest position \( P_0= [x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, \ldots, x_n^{(0)}]^{T} \), and select the optimal nest position \( X_b^{(0)} \) and its fitness value \( f_b \). The fitness value \( f \) is calculated as follows

\[
f = \frac{\bar{f}_{R-DNN}}{f_{\text{ripple}}_{R-DNN}} \tag{9}
\]

Step 2: Retain the best nest position \( x_b^{(i)} \) from the previous generation. Then, according to Eq. (10), the positions of the other nests are updated. Compare the nest position after being updated with that before, and retain the nest position with the better fitness value, so that a set of better nest positions \( g_i=[x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}]^{T} \) are obtained.

\[
x_i^{(i)} = w \times x_i^{(i-1)} + \alpha \odot L(\lambda) \quad i = 1, 2, \ldots, n \tag{10}
\]

where \( x_i^{(0)} \) represents the position of the \( i \)-th nest in the \( i \)-th generation, \( \alpha \) is the step size, \( L(\lambda) \) is the Levy random search path, and \( L \sim \mu=1^{-\lambda} (1<\lambda \leq 3) \), \( w \) is a non-linear decreasing inertia weight.

\[
w = \left( \frac{2}{\text{iter}} \right)^{0.3} \tag{11}
\]

where \( \text{iter} \) represents the number of iterations.

Step 3: A random number \( r \in [0,1] \) which obeys a uniform distribution is used as the probability of alien eggs being found. Compared with \( P_a \), the nest position with a low probability of being found is retained,
whereas a nest position with high probability of being found is changed according to Eq. (12). Compare the nest position after the change with that before the change, and retain the nest position with the better fitness value, so that a new and better group of nest positions $k_t=[x^{(t)}_1, x^{(t)}_2, \ldots, x^{(t)}_n]^T$ is obtained.

$$x_i^{(t)} = x_i^{(t-1)} + r(x_j^{(t-1)} - x_k^{(t-1)}) \quad i = 1, 2, \ldots, n$$  \hspace{1cm} (12)

where $r$ is the scaling factor. $x^{(t-1)}_i$ and $x^{(t-1)}_k$ represent two random different solutions.

Step 4: According to Eq. (13), a Gaussian perturbation is applied to the nest position $k_t$ to obtain a new set of nest positions, $p'_t$. Each nest in $p'_t$ is compared with each corresponding nest in $k_t$ to retain the nest position with the better fitness value. Finally, a better nest position $p_t$ is obtained.

$$p'_t = k_t + a \odot \mathbf{e}$$  \hspace{1cm} (13)

In the above, $\mathbf{e}$ is a random matrix of the same order as $k_t$, and $a$ is a constant.

Step 5: The optimal nest position $x^{(t)}_b$ and its fitness value $f_b$ are selected. If the iteration stop condition is reached, the optimal fitness value $f_b$ and the corresponding optimal position $x^{(t)}_b$ are output. Otherwise, the process returns to Step 2 to continue with the iterative update.

Fig. 10 compares the iteration curve of the MCS with standard CS, PSO, and GA algorithms under the same maximum number of iterations. These algorithms are implemented in Matlab.

The results show that the MCS algorithm has the fastest search speed and highest fitness value corresponding to the solution. It further shows that the MCS algorithm is more suitable for solving the optimal structural parameters of complex motors. The optimal structural parameters found by MCS algorithm are shown in Tab. 3.

**Tab. 3 Optimal variable value obtained by modified cuckoo search (MCS)**

| Design variables          | Optimal value |
|---------------------------|---------------|
| Polar distance $\tau$/mm  | 19.53         |
| Magnet thickness $h$/mm   | 4.03          |
| Magnet width $z$/mm       | 14.45         |
| Air gap height $\delta$/mm| 10.62         |
| Coil width $d$/mm         | 6.83          |

5 Experiment

To prove the feasibility and effectiveness of the motor optimization method used in this study, an optimized prototype is manufactured and tested on an experimental platform, as shown in Fig. 11. The thrust curve comparison test results of the initial motor and optimized motor are presented in Fig. 12.

In this study, the average thrust and thrust fluctuation rate are selected as the optimal evaluation indexes. Their evaluation expressions are shown in Eq. (14) and Eq. (15), respectively.

$$\overline{F} = \frac{\sum F_i}{N}$$  \hspace{1cm} (14)
where $F_i$ is the thrust of each sampling point, and $n$ is the number of sampling points.

The calculated results show that the average thrust of the initial motor is 58.28 N, and the thrust fluctuation is 3.54%. The average thrust of the optimized motor is 63.19 N, and the thrust fluctuation is 1.95%. Therefore, by applying the optimization strategy proposed in this study, the average thrust of the motor is increased by 8.42%, and the thrust fluctuation is reduced by 44.92%.

6 Conclusions

Given that an analytical model is not accurate enough to meet the needs of a motor’s optimal design, we proposed an effective alternative method, R-DNN. Compared with a traditional machine learning modeling method (KNN), R-DNN has stronger modeling capability. As general optimization algorithms have slow convergence rates and it is difficult to jump out of local optimal solutions, an efficient global optimization algorithm, i.e., the MCS, is proposed. Compared with GAs, PSOs, and standard CSs, the MCS algorithm has the strongest search capability and fastest search speed. Finally, the experimental results from the optimized prototype are evidently better than those of the original motor are, proving the effectiveness of the optimization design work performed in this study.

References

[1] M Wang, L Li, R Yang. Overview of thrust ripple suppression technique for linear motors. Chinese Journal of Electrical Engineering, 2017, 2(1): 77-84.
[2] M Li, J Zhao, Y Hu. Active disturbance rejection position servo control of PMSLM based on reduced-order extended state observer. Chinese Journal of Electrical Engineering, 2020, 6(2): 30-41.
[3] J Si, S Zhao, H Feng, et al. Multi-objective optimization of surface-mounted and interior permanent magnet synchronous motor based on Taguchi method and response surface method. Chinese Journal of Electrical Engineering, 2018, 4(2): 67-73.
[4] R Cao, Y Jin, M Lu, et al. Quantitative comparison of linear flux-switching permanent magnet motor with linear induction motor for electromagnetic launch system. IEEE Transactions on Industrial Electronics, 2018, 65(9): 7569-7578.
[5] S U Chung, J W Kim, B C Woo, et al. Force ripple and magnetic unbalance reduction design for doubly salient permanent magnet linear synchronous motor. IEEE Transactions on Magnetics, 2011, 47(10): 4207-4210.
[6] M H Ravanji, Z Nasiri-Gheidiari. Design optimization of a ladder secondary single-sided linear induction motor for improved performance. IEEE Transactions on Energy Conversion, 2015, 30(4): 1595-1603.
[7] E Öksützepe. In-wheel switched reluctance motor design for electric vehicles by using a pareto-based multiobjective differential evolution algorithm. IEEE Transactions on Vehicular Technology, 2017, 66(6): 4706-4715.
[8] F Dong, J Zhao, J Song, et al. Optimal design of permanent magnet linear synchronous motors at multispeed based on particle swarm optimization combined with SN ratio method. IEEE Transactions on Energy Conversion, 2018, 33(4): 1943-1954.
[9] J Song, J Zhao, F Dong, et al. A novel regression modeling method for PMSLM structural design optimization using a distance-weighted KNN algorithm. IEEE Transactions on Industry Applications, 2018, 54(99): 4198-4206.
[10] C C Hwang, L Y Lü, C T Liu, et al. Optimal design of an SPM motor using genetic algorithms and Taguchi method. IEEE Transactions on Magnetics, 2008, 44(11): 4325-4328.
[11] M Wang, L Li, D Pan. Detent force compensation for PMLSM systems based on structural design and control method combination. IEEE Transactions on Industrial Electronics, 2015, 62(11): 6845-6854.
[12] Y Zhang, Z Yang, M Yu, et al. Analysis and design of double-sided air core linear servo motor with trapezoidal permanent magnets. IEEE Transactions on Magnetics, 2011, 47(10): 3236-3239.
[13] L Li, Y Tang, D Pan. Design optimization of air-cored PMLSM with overlapping windings by multiple population genetic algorithm. IEEE Transactions on Magnetics, 2014, 50(11): 1-5.
[14] B H Lee, K S Kim, J P Hong, et al. Optimum shape design of single-sided linear induction motors using response surface methodology and finite-element method. IEEE Transactions on Magnetics, 2011, 47(10): 3657-3660.
[15] S Doi, H Sasaki, H Igarashi. Multi-objective topology
optimization of rotating machines using deep learning. *IEEE Transactions on Magnetics*, 2019, 55(6): 1-5.

[16] M Khodayar, O Kaynak, M E Khodayar. Rough deep neural architecture for short-term wind speed forecasting. *IEEE Trans. Ind. Inform.*, 2017, 13(6): 2770-2779.

[17] C Shen, B Wang, K Xu, et al. Turn-to-turn short circuit of motor stator fault diagnosis in continuous state based on deep auto-encoder. *IET Electric Power Applications*, 2019, 13(10): 2770-2779.

[18] W Li, H Li, Q Wu, et al. Simultaneously detecting and counting dense vehicles from drone images. *IEEE Transactions on Industrial Electronics*, 2019, 66(12): 9651-9662.

[19] X Liu, W N Fu. A dynamic dual-response-surface methodology for optimal design of a permanent-magnet motor using finite-element method. *IEEE Transactions on Magnetics*, 2016, 52(3): 1-4.

[20] H R Massrur, T Niknam, M Mardaneh, et al. Harmonic elimination in multilevel inverters under unbalanced voltages and switching deviation using a new stochastic strategy. *IEEE Transactions on Industrial Informatics*, 2016, 12(2): 716-725.

[21] A H Gandomi, X S Yang, A H Alavi. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers*, 2013, 29(1): 17-35.

[22] Z C Cao, C R Lin, M C Zhou, et al. Scheduling semiconductor testing facility by using cuckoo search algorithm with reinforcement learning and surrogate modeling. *IEEE Transactions on Automation Science and Engineering*, 2019, 16(2): 825-837.

[23] N J Cheung, X M Ding, H B Shen. A Nonhomogeneous cuckoo search algorithm based on quantum mechanism for real parameter optimization. *IEEE Transactions on Cybernetics*, 2017, 47(2): 391-402.

[24] A A El-Fergany, A Y Abdelaziz. Capacitor allocations in radial distribution networks using cuckoo search algorithm. *IET Generation Transmission & Distribution*, 2014, 8(2): 223-232.

[25] D N Vo, P Schegner, W Ongsakul. Cuckoo search algorithm for non-convex economic dispatch. *IET Generation Transmission & Distribution*, 2013, 7(6): 645-654.

[26] W Han, J Xu, M Zhou, et al. Cuckoo search and particle filter-based inversing approach to estimating defects via magnetic flux leakage signals. *IEEE Transactions on Magnetics*, 2016, 52(4): 1-11.

[27] N Agrawal, A Kumar, V Bajaj, et al. Design of bandpass and bandstop infinite impulse response filters using fractional derivative. *IEEE Transactions on Industrial Electronics*, 2019, 66(2): 1285-1295.

Weitao Wang was born in Fuyang, China. He received a B.S. degree in electrical engineering and automation from Anhui University, Hefei, China, in 2018. Since 2018, he has been working toward an M.S. degree in detection technology and automatic equipment at the School of Electrical Engineering, Anhui University, Hefei, China.

His current research interests include the design, analysis, and optimization of linear motors.

Jiwen Zhao was born in Danshan, China. He received his Ph.D. degree in 2005 from the University of Science and Technology of China, Hefei, China. Since 2019, he has been working at the School of Electrical and Automation Engineering at Hefei University of Technology, Hefei, China. He has authored/coauthored more than 50 papers in international journals, such as IEEE Transactions on Industrial Electronics, IEEE Transactions on Industrial Informatics, and IEEE Transactions on Industrial Applications. He is a reviewer for more than ten SCI/SCIE journals.

His research interests include linear motor optimization design, linear motor control, and photoelectric detection technology.

Yang Zhou was born in Lu’an, China. He received a B.S. degree in electrical engineering and automation while minoring in software engineering in 2018 from Anhui University, Hefei, China. Since 2018, he has been working toward an M.S. degree in control engineering with the School of Electrical Engineering and Automation, Anhui University, Hefei, China.

His current research interests include electrical machines, image processing, and photoelectric detection technology.

Fei Dong was born in Yuncheng, China. She received a B.S. degree in mechanical engineering and an M.S. degree in mechatronic engineering from Chang’an University, Xi’an, China, in 2006 and 2011, respectively. Since 2011, she has been a lecturer with the School of Electrical Engineering and Automation, Anhui University, Hefei, China.

Her research interests include the design, analysis, and optimization of permanent magnet synchronous motors.