

Fast Backfire Double Annealing Particle Swarm Optimization Algorithm for Parameter Identification of Permanent Magnet Synchronous Motor

Dingdou Wen, Chuandong Shi, Kaixian Liao, Jianhua Liu, and Yang Zhang*

Abstract—When particle swarm optimization (PSO) is used to identify the parameters of permanent magnet synchronous motor (PMSM), the movement of particles is not selective, which makes the algorithm easy to fall into the local optimum, and the accuracy is poor. The simulated annealing particle swarm optimization (SAPSO) improves the accuracy and evolution speed, but SAPSO has redundant iteration problems. To solve these problems, a motor parameter identification method based on fast backfire double anneal particle swarm optimization (FBDAPSO) is proposed. By reducing the optimization time and quickly tempering and annealing the “misunderstood” difference, the motor adjustable model and fitness function are designed, and the number of iterations is constantly reset to achieve the effect of online identification. Under different working conditions, simulated and experimental results show that the proposed method can quickly and accurately identify the four parameters of the motor’s stator, winding resistance, stator winding d -axis inductance, stator winding q -axis inductance, and permanent magnet flux linkage at the same time, compared with the traditional method of parameter identification, and it has better accuracy, rapidity, and robustness.

1. INTRODUCTION

Permanent magnet synchronous motor (PMSM) [1–3] is widely used by virtue of its high efficiency and high power density. PMSM is divided into interior permanent magnet synchronous motor (IPMSM) and surface permanent magnet synchronous motor (SPMSM) according to the different installation positions of permanent magnets of the motor. IPMSM has the advantages of high power density, high operating efficiency, and low failure rate, and it is widely used in industrial drives, electric vehicles, and new energy power generation [4–6]. However, IPMSM is a strongly coupled, multivariable, and non-linear system [7]. The parameters of the motor are easily affected by unpredictable factors. For example, the changes of temperature and frequency cause the variety of the stator resistance of the motor, and the d - q axis inductance and permanent magnet flux can be affected by current and magnetic saturation. These changes of parameters lead to the unsatisfactory effect of vector control and decrease the reliability of the system. In addition, the parameters setting of the PID controller is related to the parameters of the motor. When the parameters of the motor change, the parameters of the PID controller need to be re-tuned; otherwise, the high-precision control requirements will not be met. For this reason, scholars have proposed two kinds of parameter identification methods for PMSM parameter identification: offline and online ones [8]. Online parameter identification can meet the needs of real-time monitoring of motor parameter changes, and it is one of the hotspots of current research. Traditional online parameter methods mainly include: Extended Kalman filter [9, 10], model reference adaptive system [11, 12], recursive least squares [13, 14], etc. The authors of [9] and [10] propose an extended

Received 28 May 2021, Accepted 7 July 2021, Scheduled 3 August 2021

* Corresponding author: Yang Zhang (hut_zy@163.com).

The authors are with the College of Electrical and Information Engineering, Hunan University of Technology, Zhuzhou 412007, China.

Kalman filter (EKF) for parameter identification. EKF is suitable for nonlinear time-varying systems and can handle noise-sensitive problems in parameter identification. However, the selection of Q and R matrices depends on experience or experiment. Different Q and R matrices make the accuracy of identification uncertain, and EKF also needs complex matrix operation and accurate mathematical model. The model reference adaptive system (MRAS) in [11, 12] has good convergence and is easy to implement, but the design of its adaptive law is complicated. When multiple parameters are required to be identified at the same time, it is difficult to determine an appropriate adaptive law, and the selection of initial parameters has a great influence on the identification accuracy. The application of recursive least squares (RLS) method in [13, 14] has the advantages of simplicity and ease of realization for motor parameter identification, but RLS needs to process a large amount of data and is prone to data saturation problems. In addition, the linear parameter model requires derivation during simplification and approximation, and noise will have a greater impact on the derivation result, which leads to poor identification accuracy and robustness. When the above methods identify the parameters of the SPMSM, since the d -axis and q -axis inductance values of the SPMSM are equal, only three parameters, resistance, flux linkage, and excitation inductance, need to be identified, which is relatively simple to implement. But when the above methods are applied to the IPMSM with more parameters, the design of the parameter identification system is more difficult.

In view of the shortcomings of the above parameter identification methods, intelligent optimization algorithms with the advantages of low objective function requirements and high efficiency are widely used [15]. At present, scholars mostly use neural network [16, 17], genetic algorithm [18], particle swarm algorithm [19, 20], etc. In [16, 17], the neural network identification method was used to identify the resistance and flux linkage parameters of PMSM. The neural network algorithm has high accuracy, but the speed and stability of parameter identification require appropriate convergence factors. The authors of [18] proposed genetic algorithm to identify the parameters of IPMSM, and the identification error was small, but the genetic algorithm is easy to mature early, and the amount of calculation is large. Particle swarm optimization (PSO) [21] is widely used in motor parameter identification because of its simple algorithm and fast speed. But the particle movement of PSO is not selective, and the particle still updates the current position even if the fitness function of the next particle is very poor. The behavior of PSO leads to its poor local search ability. Simulated annealing (SA) is a kind of thermodynamic process to simulate the cooling of high temperature metal, which has the ability to jump out of the local optimum, and SA has low requirements for initial value and good robustness [22]. Therefore, some scholars combined PSO with SA to propose simulated annealing particle swarm optimization (SAPSO) [23, 24]. SAPSO improves the local search capability and evolution speed of PSO. At present, SAPSO has not been applied in the parameter identification of PMSM. This paper proposes for the first time that SAPSO is applied to identify the parameters of PMSM to improve the parameter identification performance, but it takes a long time to anneal all particles.

Aiming at the problems of PSO and SAPSO, a fast backfire double anneal particle swarm optimization (FBDAPSO) algorithm is proposed. FBDAPSO retains the advantages that SAPSO has better accuracy and robustness than PSO and performs better than SAPSO in terms of evolution speed.

The contributions of the paper are as follows.

(i) First application of SAPSO in PMSM parameter identification for its good robustness advantage avoids the defect that PSO is easy to fall into local optimum. The simulation and experimental results show that the parameter identification effect of SAPSO is better than PSO.

(ii) On the basis of SAPSO, a lower initial temperature is used to shorten the time to find the optimal solution during annealing, but it increases the risk of falling into local extremes. Therefore, the algorithm adds an improved tempering idea, which performs a rapid temperature recovery annealing operation on the promising bad solutions to ensure the accuracy.

(iii) Most improved PSO algorithms used the maximum number of iterations as the algorithm termination condition [25, 26], and the effect of online identification cannot be achieved because only the parameter values of some points are identified. This paper adopts the method of resetting the number of iterations to make the algorithm termination condition change with the change of the system running time, so that it can identify each sampling point and meet the requirements of online identification.

(iv) The results of simulation and RT-LAB hardware-in-the-loop simulations (HILS) show that the proposed method can quickly and accurately identify the four parameters of motor: stator resistance,

d -axis inductance, q -axis inductance, and permanent magnet flux linkage.

The rest of this paper is organized as follows. Section 2 introduces the model of PMSM. Section 3 introduces the proposed method. Section 4 designs the parameter identification scheme of FBDAPSO. Section 5 and Section 6 present the results of simulation and experiment to illustrate the superiority of FBDAPSO in parameter identification. Finally, Section 7 gives the conclusions.

2. PMSM MODEL

Assuming that there is no core saturation, no armature reaction during the commutation process, no damping effect, and the magnetic field is sinusoidally distributed. The voltage equation of IPMSM under the d - q coordinate system is expressed as follows:

$$\begin{bmatrix} u_d \\ u_q \end{bmatrix} = \begin{bmatrix} R_s & -\omega_e L_q \\ \omega_e L_d & R_s \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_d \\ \psi_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_e \psi_f \end{bmatrix} \quad (1)$$

The flux linkage equation is as follows:

$$\begin{bmatrix} \psi_d \\ \psi_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \psi_f \\ 0 \end{bmatrix} \quad (2)$$

where R_s is the stator resistance; u_d and u_q are the stator voltages of the d -axis and q -axis; i_d and i_q are the stator currents of the d -axis and q -axis; ψ_d and ψ_q are d -axis and q -axis stator flux linkage; L_d and L_q are the stator inductances of the d -axis and q -axis; ω_e is the speed, and ψ_f is the permanent magnet flux linkage.

The current equation in the d - q coordinate system is obtained by Eqs. (1) and (2):

$$\frac{d}{dt} \begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & \omega_e \frac{L_q}{L_d} \\ -\omega_e \frac{L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} u_d \\ u_q \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{\omega_e \psi_f}{L_q} \end{bmatrix} \quad (3)$$

3. FAST BACKFIRE DOUBLE ANNEAL PARTICLE SWARM OPTIMIZATION ALGORITHM

3.1. Particle Swarm Optimization

The principle of PSO is to search for the optimal solution of a complex problem through the collaboration of particles. PSO first initializes the particles randomly, and each particle is a solution of the problem. In the iteration, the particles track the individual optimal solution P_{best} and the group optimal solution P_{gbest} through the memory function. The particles continuously iteratively share the position information of the individual and the group, and then the algorithm evaluates the pros and cons of the current position through the fitness function, so that the particles gradually approach the point with the smallest fitness value, and finally the algorithm obtains the optimal solution to the problem.

In an n -dimensional space, m particles form a population $X = (x_1, x_2, \dots, x_m)$. The particles have their own speed and position. The position of the particle i is $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$, and its speed is $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$. The individual optimal solution of the particle is $P_{best_i} = (P_{best_i1}, P_{best_i2}, \dots, P_{best_in})$, and the population optimal solution is $P_{gbest} = (P_{gbest_1}, P_{gbest_2}, \dots, P_{gbest_n})$.

According to the principle of approximating the optimal particle position, the particle x_i updates the velocity and position according to Eq. (4):

$$\begin{cases} v_{in}^{k+1} = w * v_{in}^k + c_1 r_1 (P_{best_in}^k - x_{in}^k) + c_2 r_2 (P_{gbest_n}^k - x_{in}^k) \\ x_{in}^{k+1} = x_{in}^k + v_{in}^{k+1} \end{cases} \quad (4)$$

where v_{in} is the component of flight velocity vector in n -dimensional of particle i ; x_{in} is the component of position vector in n -dimensional of particle i ; P_{best_in} is the best point found by the particle i in

the n -dimensional component; $P_{gbest,n}$ is the best point obtained by the particle swarm in the current n -dimensional component; k is the number of iterations; r_1 and r_2 are random numbers between 0 and 1; c_1 and c_2 are acceleration coefficients. The velocity of particles is between $[-v_{\max}, v_{\max}]$, and v_{\max} is the maximum velocity.

w is the inertia weighting factor, which adopts the method of linearly decreasing weight, and the expression is as follows:

$$w = w_{\max} - (w_{\max} - w_{\min}) \frac{k}{k_{\max}} \quad (5)$$

where w_{\max} is the initial inertia weight; w_{\min} is the inertia weight at the maximum number of iterations; k is the current number of iterations; k_{\max} is the maximum number of iterations.

The basic steps of PSO:

- 1) Initialize the particle swarm, and set the initial position and velocity randomly.
- 2) Update the speed and position according to Eq. (4).
- 3) Calculate the fitness value of the new position. If the fitness value of the new particle is better than the old particle, set the position of the new particle as the individual optimal solution P_{best} ; otherwise, keep the original state.
- 4) Find the group optimal solution P_{gbest} according to the individual optimal solution P_{best} .
- 5) After reaching the convergence criterion, output the P_{gbest} ; otherwise, go to 2).

3.2. Simulated Annealing

The core idea of SA is to accept the state of poor fitness with probability. At the temperature T , a new position j is generated from the current position i . The internal energies of the two positions are E_i and E_j . If $E_j < E_i$, the algorithm receives the new position; otherwise, it calculates the probability p_{ij} :

$$p_{ij} = e^{-\frac{E_j - E_i}{T}} \quad (6)$$

When $p_{ij} > \text{random}(0, 1)$, the algorithm receives the new position j ; on the contrary, it keeps the original position i , because it accepts the poor point with probability, which can jump out of the local optimal solution.

Basic steps of SA:

- 1) Set the initial temperature T_1 , and randomly generate the initial solution x_{old} .
- 2) Generate a new solution x_{new} near the initial solution x_{old} ; calculate the difference Δf between the fitness function $f(x_{new})$ and $f(x_{old})$; and receive the new state x_{new} according to the probability $\min\{1, \exp(-\Delta f/T)\} > r$, where r is a random number in the interval $(0, 1)$.
- 3) Annealing operation: $T = C_1 T$, where C_1 is between $(0, 1)$. When the convergence criterion is reached, the final accepted state is output; otherwise, go to 2).

3.3. Fast Backfire Double Anneal Particle Swarm Optimization Algorithm

SAPSO has absorbed the advantages of PSO and SA, and its accuracy and evolution speed have been improved, but SAPSO has the problem of redundant iteration.

FBDAPSO adopts double annealing form, which is divided into general annealing and tempering annealing: the general annealing operation works when the algorithm accepts a new state, and the selected initial temperature T_1 is lower than the conventional one, which will reduce the time to converge to a high-quality solution, but the risk of falling into a local extreme value will increase; the tempering annealing operation works when the algorithm receives a promising deteriorating solution, and the rapid annealing strategy proposed by *Ingber* is used for the increased temperature T_2 to maintain the probability and accuracy of the algorithm.

The local search of FBDAPSO is more purposeful; its optimization time is shorter; the rapid tempering double annealing process is convenient to implement; and the complexity of the algorithm will not be upgraded. The general annealing form is as follows:

$$T = C_1 T \quad (7)$$

The form of rapid tempering annealing is as follows:

$$T = \begin{cases} T & e^{-\frac{\Delta f}{T}} < \text{random}(0,1) \\ T + C_2^d T_2 & e^{-\frac{\Delta f}{T}} > \text{random}(0,1) \end{cases} \quad (8)$$

In this model, T is the current temperature; the initial value is the general annealing initial temperature T_1 ; C_1 is the general annealing coefficient; d is the number of times of receiving the difference; T_2 and C_2 are the temperature increased by tempering and the rapid annealing coefficient.

The basic steps of FBDSAPSO:

- 1) Initialize the position and velocity of the particles randomly, and set the initial annealing temperature and coefficient.
- 2) Set P_{best} of the particle to the current position, and set P_{gbest} to the position of the best particle in the particle swarm.
- 3) If the algorithm convergence criterion is satisfied, go to 11); otherwise, go to the next step.
- 4) Update the position and velocity of the particles according to Eq. (4).
- 5) If the fitness difference Δf between the new and old positions is less than 0, and the particle enters the new position, go to 7); otherwise, go to the next step.
- 6) Generate a random number r between (0, 1), and if $\min\{1, \exp(-\Delta f/T)\} > r$, the particle enters a new position; the temperature rises; $T = T + C_2^d T_2$; the algorithm jumps out of the local optimum; go to 7); otherwise, keep the original state, and go to 4).
- 7) If the fitness of the new particle is better than P_{best} , P_{best} is set to the current position; otherwise, P_{best} remains unchanged.
- 8) If the fitness of the new particle is better than P_{gbest} , P_{gbest} is set to the current position; otherwise, P_{gbest} remains unchanged.
- 9) Decrease temperature according to Eq. (7).
- 10) Judge whether the algorithm convergence criterion is satisfied. If satisfied, go to 11); otherwise go to 4).
- 11) Output P_{gbest} , and the algorithm operation ends.

In the process of particle optimization, the particle with better fitness is selected; on the contrary, the particle will not easily give up the difference and go directly to the next position, and instead, it will make a decision after probabilistic testing.

A lower initial temperature can increase the convergence speed of the algorithm, but it reduces the probability of probing. When the algorithm receives a promising poor point, the tempering annealing operation is performed to make up for the probability of trial and ensure the accuracy of the algorithm.

4. REALIZATION OF PARAMETER IDENTIFICATION THROUGH FBDAPSO

4.1. Principle of Parameter Identification

When the structure of the mathematical model of IPMSM is known, the parameter identification of the motor can be transformed into an optimization problem. The basic idea of parameter identification of IPMSM based on FBDAPSO is to optimize the error of the output between the reference model and the adjustable model of motor. The identified parameters of the adjustable motor model continuously are modified through FBDAPSO, until the maximum number of iterations or accuracy requirements of the algorithm are reached, and the finally obtained optimal solution is used as the actual parameters of the motor.

A mathematical model with the same structure and adjustable parameters is designed as follows:

$$\frac{d}{dt} \begin{bmatrix} \hat{i}_d \\ \hat{i}_q \end{bmatrix} = \begin{bmatrix} -\frac{\hat{R}_s}{\hat{L}_d} & \omega_e \frac{\hat{L}_q}{\hat{L}_d} \\ -\omega_e \frac{\hat{L}_d}{\hat{L}_q} & -\frac{\hat{R}_s}{\hat{L}_q} \end{bmatrix} \begin{bmatrix} \hat{i}_d \\ \hat{i}_q \end{bmatrix} + \begin{bmatrix} \frac{1}{\hat{L}_d} & 0 \\ 0 & \frac{1}{\hat{L}_q} \end{bmatrix} \begin{bmatrix} u_d \\ u_q \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{\omega_e \hat{\psi}_f}{\hat{L}_q} \end{bmatrix} \quad (9)$$

where \hat{R}_s , \hat{L}_d , \hat{L}_q , and $\hat{\psi}_f$ are the adjustable motor parameters to be identified; \hat{i}_d and \hat{i}_q are the currents of the d -axis and q -axis output by the adjustable model; ω_e is the speed.

The first-order forward Euler method is commonly used in engineering for discretization, which is as follows:

$$\frac{di}{dt} = \frac{i(k+1) - i(k)}{T_s} \quad (10)$$

where T_s is the sampling time; T_s is small enough, and the difference is similar to the differential.

Then the discretization of Eq. (9) is as follows:

$$\begin{bmatrix} \hat{i}_d(k+1) \\ \hat{i}_q(k+1) \end{bmatrix} = \begin{bmatrix} 1 - \frac{\hat{R}_s(k)T_s}{\hat{L}_d(k)} & \omega_e(k) \frac{\hat{L}_q(k)T_s}{\hat{L}_d(k)} \\ -\omega_e(k) \frac{\hat{L}_d(k)T_s}{\hat{L}_q(k)} & 1 - \frac{\hat{R}_s(k)T_s}{\hat{L}_q(k)} \end{bmatrix} \begin{bmatrix} \hat{i}_d(k) \\ \hat{i}_q(k) \end{bmatrix} + \begin{bmatrix} \frac{T_s}{\hat{L}_d(k)} & 0 \\ 0 & \frac{T_s}{\hat{L}_q(k)} \end{bmatrix} \begin{bmatrix} u_d(k) \\ u_q(k) - \omega_e(k)\hat{\psi}_f(k) \end{bmatrix} \quad (11)$$

The similarity between the reference model of the motor and the adjustable model needs to be measured by designing an appropriate fitness function. In order to characterize the pros and cons of the result of parameter identification, the fitness function is defined as:

$$f(R_s, L_d, L_q, \psi_f) = \frac{1}{k} \left(i_d(k) - \hat{i}_d(k) \right)^2 + \left(i_q(k) - \hat{i}_q(k) \right)^2 \quad (12)$$

where k is the current number of iterations.

The principle block diagram of identifying the parameters of IPMSM by FBDAPSO is shown in Fig. 1.

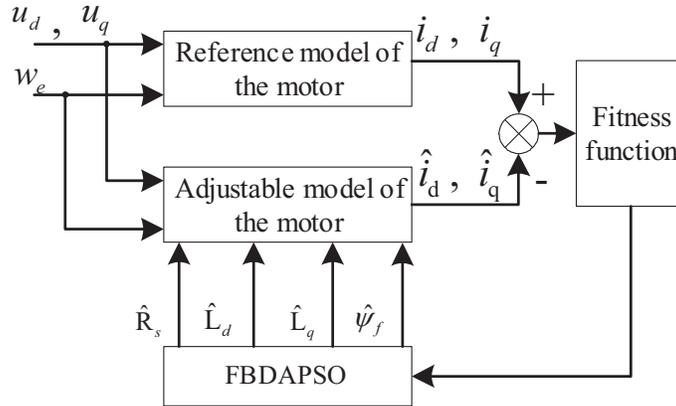


Figure 1. Principle block diagram of parameter identification.

In Fig. 1, the inputs of the adjustable motor model and the reference model both are d -axis voltage u_d , q -axis voltage u_q , and speed ω_e . The error of the current output from the reference model and the adjustable model of the motor is used as the input of the fitness function, and the fitness value of the particle is obtained from the fitness function.

The FBDAPSO is to continuously screen better parameter values to be identified, so that the error between reference model and adjustable model tends to 0, or the fitness function value tends to 0. The smaller the fitness value is, the more accurate the identification value is.

At present, the collected data are first saved in the workplace, and then the algorithm is run to obtain the motor parameter values. The identification is only the value at a certain sampling point, which is offline identification [27].

In order to obtain the parameters of IPMSM in real time, the termination condition of the identification algorithm should be followed by the running time of the system. In this paper, the proposed algorithm adopts the method of resetting after every 50 iterations to realize the identification of each sampling point, which can meet the requirements of online identification. The voltage and current signals of the direct axis and quadrature axis of the motor and speed collected in real time are used as the input of the parameter identification system, and then the real-time parameters of the motor are obtained through algorithm iteration.

4.2. Realization of Parameter Identification

The steps of motor parameter identification are as follows:

1) Given the initial values of the parameters $\hat{R}_s(k-1)$, $\hat{L}_d(k-1)$, $\hat{L}_q(k-1)$ and $\hat{\psi}_f(k-1)$ randomly, and the initial values of the speeds of $V \cdot \hat{R}_s(k-1)$, $V \cdot \hat{L}_d(k-1)$, $V \cdot \hat{L}_q(k-1)$ and $V \cdot \hat{\psi}_f(k-1)$, set the general annealing temperature T_1 and coefficient C_1 , tempering temperature T_2 and annealing coefficient C_2 .

2) Collect electrical signals, and obtain the output $\hat{i}_d(k-1)$ and $\hat{i}_q(k-1)$ of the adjustable model from Eq. (11).

3) Get the fitness value of the initial parameters:

$$f_{k-1} [R_s(k-1), L_d(k-1), L_q(k-1), \psi_f(k-1)] = [i_d(k-1) - \hat{i}_d(k-1)]^2 + [i_q(k-1) - \hat{i}_q(k-1)]^2 \tag{13}$$

4) Determine the current individual optimal parameter value and group optimal parameter value, and update the parameter value according to Eq. (4), for example:

$$\begin{cases} V \cdot \hat{R}_s(k) = w * V \cdot \hat{R}_s(k-1) + c_1 r_1 [\hat{R}_{pbest} - \hat{R}_s(k-1)] + c_2 r_2 [\hat{R}_{gbest} - \hat{R}_s(k-1)] \\ \hat{R}_s(k) = \hat{R}_s(k-1) + V \cdot \hat{R}_s(k) \end{cases} \tag{14}$$

where \hat{R}_{gbest} is the optimal resistance value of the individual; \hat{R}_{gbest} is the optimal resistance value of the group. In the same way, $\hat{L}_d(k)$, $\hat{L}_q(k)$, and $\hat{\psi}_f(k)$ can be obtained.

5) $\hat{i}_d(k)$ and $\hat{i}_q(k)$ corresponding to the new parameters are obtained by Eq. (11), and then the corresponding fitness value is calculated.

6) The principle of *Metropolis* is used to judge whether the new parameter is accepted or not:

$$\Delta f = f_k - f_{k-1} \tag{15}$$

$$p = \begin{cases} 1 & \Delta f \leq 0 \\ e^{-\frac{\Delta f}{T}} & \Delta f > 0 \end{cases} \tag{16}$$

When the fitness deviation Δf is less than 0 or $e^{-\frac{\Delta f}{T}}$ greater than a random number between (0, 1), the algorithm updates to the new parameter value; otherwise, the original parameter value is maintained.

When the fitness deviation Δf is 0, although the new parameter value is accepted, the fitness value does not decrease in fact, and this situation occurs many times. In order to avoid the resulting non-convergence of the algorithm, when $\Delta f = 0$, Δf is regarded as a very small number, and it takes 0.001. This change will not affect the accuracy of the algorithm. The new acceptance rule is:

$$p = \begin{cases} 1 & \Delta f < 0 \\ e^{-\frac{0.001}{T}} & \Delta f = 0 \\ e^{-\frac{\Delta f}{T}} & \Delta f > 0 \end{cases} \tag{17}$$

7) The rapid tempering annealing operation is carried out according to Eq. (8).

8) The general annealing operation is carried out according to Eq. (7), $k = k + 1$.

9) If the system is over, output the optimal parameters; otherwise, go to 4) to continue iterating.

A reasonable cooling schedule can guarantee the performance of the algorithm. In the early stage, the probability of accepting the difference should be close to 1. In this paper, the acceptance probability λ is selected as 0.9999, and the initial temperature is obtained through the formula proposed by *Aarts et al.* [28]:

$$T = \frac{\overline{\Delta f+}}{\ln \frac{m_2}{m_2 \lambda - m_1 (1 - \lambda)}} \tag{18}$$

where $\overline{\Delta f+}$ represents the average increment of m_2 transformations that make the fitness value increase, and m_1 represents the number of transformations that make the fitness function value smaller.

The average value of random $(0, 1)$ is 0.5, which is equal to $\exp(-0.69)$, which shows that the new state can be adopted when Δf is within 0.69 times of T . When the final fitness deviation is controlled within 0.01, the cooling coefficient can be obtained according to the initial temperature and the number of iterations.

5. SIMULATION ANALYSIS

In order to verify the performance of the algorithm, the vector control system of IPMSM is established in *Matlab/Simulink*, and the $i_d = 0$ current control strategy is adopted in the simulation. Its principle block diagram is shown in Fig. 2.

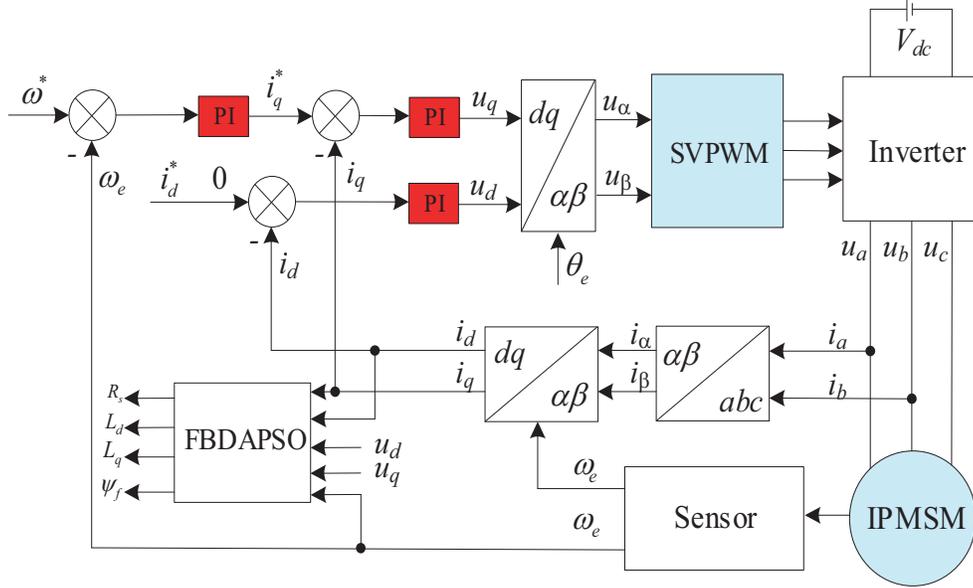


Figure 2. Vector control system block diagram.

The parameters of motor used in the simulation are shown in Table 1.

Table 1. Motor parameter table.

Parameter	Value	Unit
Number of pole pairs	4	pairs
Stator resistance	2.875	Ω
D -axis inductance	4.5	mH
Q -axis inductance	13.5	mH
Permanent magnet flux	0.17858	Wb
Rated speed	1200	rpm
Rated torque	20	$N \cdot m$

For the sake of generality, the initial parameters of the algorithm are all set: the population number of the particle swarm algorithm is 20; the number of iterations is reset every 50 times, and it is equal to the running time divided by the sampling time; the acceleration factors c_1 and c_2 are both set to 1.6, and w linearly decreases from 0.9 to 0.4. SAPSO selects the initial temperature T_1 of 1000 and the annealing coefficient C_1 of 0.8; the initial temperatures T_1 and T_2 of FBDAPSO are 800 and 500, respectively, and the annealing coefficients C_1 and C_2 of FBDAPSO are 0.8 and 0.9, respectively. In addition, two methods, EKF and RLS, are added for comparison.

The simulation system runs for 0.5 s; the sampling time is 1e-6s; and the initial values of the parameters to be identified are completely consistent and far away from the actual parameter values. The actual system will be disturbed by uncertain factors. In order to verify the performance of the proposed method, we run FBDAPSO, SAPSO, EKF, RLS, and PSO under different working conditions to independently identify the motor parameters for 5 times, and take their average value as the final output value.

1) Working condition 1.

The identification results and errors in the operating state with the torque of 10 N · m and the speed of 1000 r/min are shown in Table 2.

Table 2. Average result of motor parameter identification under condition 1.

Parameter	EKF	RLS	PSO	SAPSO	FBDAPSO
$R_s(\Omega)$	2.76839	2.75130	2.90530	2.88690	2.88541
Error (%)	-3.70817	-4.30261	1.05390	0.41392	0.36327
L_d (mH)	4.28985	4.69609	4.59225	4.52501	4.52121
Error (%)	-4.66988	4.35756	2.05010	0.55588	0.47227
L_q (mH)	12.92879	13.4078	13.6435	13.5757	13.5794
Error (%)	-4.23112	-0.68319	1.06340	0.56122	0.58811
ψ_f (Wb)	0.17394	0.17451	0.18208	0.17915	0.17930
Error (%)	-2.59600	-2.27909	1.96549	0.32198	0.40439
Recognition time (s)	0.160	0.193	0.176	0.116	0.094
Fitness value	~	~	2.1	0.85	0.83

The data show that FBDAPSO, SAPSO, EKF, RLS, and PSO can identify motor parameters, and the identified parameters all converge near the actual values. When the identified parameters tend to be stable, the error of EKF is 4.66988%, and the identification time is 0.16 s. The error of RLS in 0.193 s is 4.35756%; the identification error of FBDAPSO converges to less than 1% in 0.094 s; its fitness value is the smallest; SAPSO needs 0.116 s to complete the identification. Therefore, the identification speed of FBDAPSO is 19% higher than SAPSO. Because PSO is trapped in the local optimum, its identification error can be approached within 2.5% in 0.176 s. The results prove that FBDAPSO has higher credibility and speed.

2) Working condition 2.

The identification results and errors in the operating state with the torque of 20 N · m and the speed of 1200 r/min are shown in Table 3.

Table 3. Average result of motor parameter identification under condition 2.

Parameter	EKF	RLS	PSO	SAPSO	FBDAPSO
$R_s(\Omega)$	2.69591	2.74335	2.96140	2.89047	2.88963
Error (%)	-6.22939	-4.57891	3.00590	0.53821	0.50876
L_d (mH)	4.28073	4.70934	4.68753	4.53053	4.52614
Error (%)	-4.72684	4.65214	4.16733	0.67833	0.58096
L_q (mH)	12.84220	13.39842	13.7207	13.6063	13.5958
Error (%)	-4.87256	-0.75241	1.63459	0.78763	0.71005
ψ_f (Wb)	0.17061	0.17414	0.18644	0.17977	0.17989
Error (%)	-4.46333	-2.48562	4.39868	0.66797	0.73388
Recognition time (s)	0.211	0.220	0.178	0.116	0.094
Fitness value	~	~	4.6	1.29	1.26

Increasing torque and speed, the identification error of FBDAPSO can be still kept within 1%, and the convergence speed of FBDAPSO is still about 19% faster than SAPSO, while the identification error of PSO has exceeded 2.5% and is within 5%. The error of EKF is close to 6.5%, and its convergence speed becomes slow. The error of RLS also fluctuates, and the convergence speed is slower than before. The results show that the parameter identification of FBDAPSO has better accuracy, robustness, and convergence speed. The reason that it is better is that the lower initial temperature improves the algorithm speed, and the rapid tempering annealing operation guarantees the probability of the algorithm sudden jump. Even if the fitness function value changes very little, it can jump out of the local optimum.

The results show that the proposed method retains the advantages of SAPSO and has a better convergence speed, and even the accuracy of individual parameters has been improved.

6. EXPERIMENTAL VERIFICATION

In order to further verify the correctness of the proposed method, this paper uses RT-LAB to implement the hardware in the loop simulation (HILS) of the PMSM drive system. The RT-LAB experiment platform is shown in Fig. 3. The RT-LAB hardware-in-the-loop system configuration diagram of PMSM is shown in Fig. 4. The model of the DSP controller is TMS320F2812, which runs the algorithm, and RT-LAB (OP5600) is used to construct other parts of the system such as PMSM and inverter.

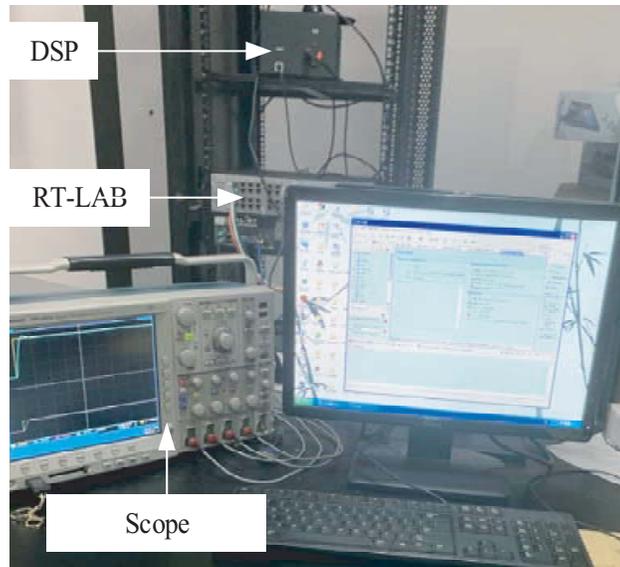


Figure 3. RT-LAB experiment platform.

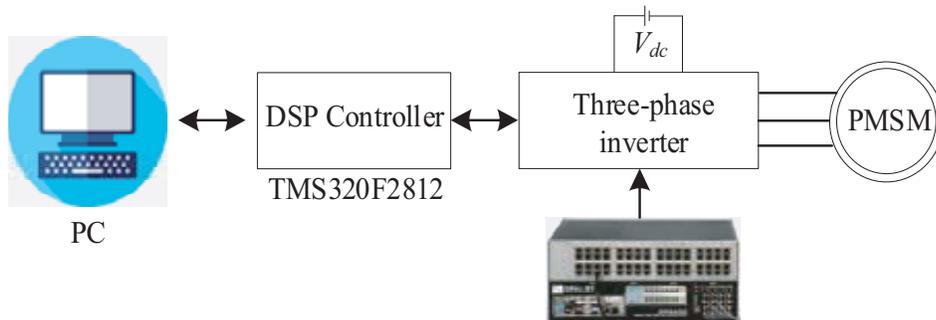


Figure 4. RT-LAB hardware-in-the-loop system configuration.

The parameters of PMSM are shown in Table 1. The display range of the scope is fixed. In the experiment, in order to better observe the effect of parameter identification, the d -axis inductance, q -axis inductance, and permanent magnet flux linkage values are enlarged by 1000, 500, and 10 times, and the fitness value is reduced by 6 times.

1) Working condition 1.

Parameter identification is carried out under the operating state with torque of 10 N·m and rotation speed of 1000 r/min. The curves of parameter identification and the curves of fitness function are shown in Figs. 5–7.

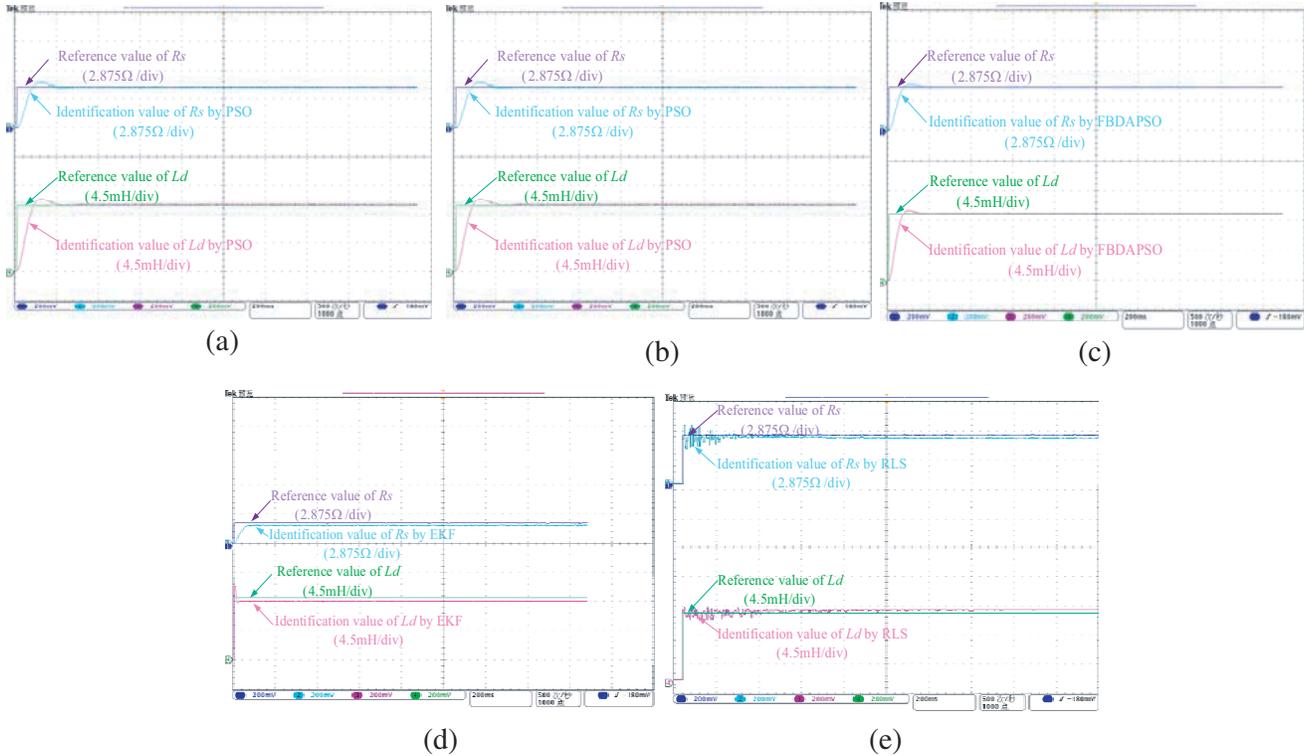


Figure 5. Identification curve of stator resistance and direct axis inductance under condition 1. (a) PSO. (b) SAPSO. (c) FBDAPSO. (d) EKF. (e) RLS.

In Fig. 5, the blue and pink curves are the identification curves of the stator resistance and d -axis inductance, respectively. The purple and green curves represent the actual value curves of the corresponding parameters. EKF completes the parameters identification of the motor in 0.165 s, and its identification error is 4.72%; RLS completes the parameters identification of the motor in 0.395 s, and its identification error is 4.39%; PSO completes the parameters identification of the motor in 0.264 s, and its identification error is 2.1%; SAPSO completes the parameter identification in 0.208 s, and its error converges to 0.65%, while FBDAPSO only needs 0.172 s to complete the identification, and its identification speed is 17.5% ahead of SAPSO.

It can be seen from Fig. 6 that EKF, RLS, PSO, SAPSO, and FBDAPSO can identify quadrature-axis inductance and permanent magnet flux linkage. The accuracy of SAPSO is close to FBDAPSO, but the identification speed of FBDAPSO is faster than SAPSO.

The curves of the change of fitness value are shown in Fig. 7. The fitness value of SAPSO converges to around 1 at 0.208 s, while FBDAPSO only takes about 0.172 s to complete it, so its convergence speed is 17.5% higher than that of SAPSO. Because PSO is trapped in local optimum, its fitness function value curve converges to around 2.5 in 0.264 s, and the error fluctuation of PSO is larger.

2) Working condition 2.

Parameter identification is carried out under the operating state with the torque of 20 N·m and

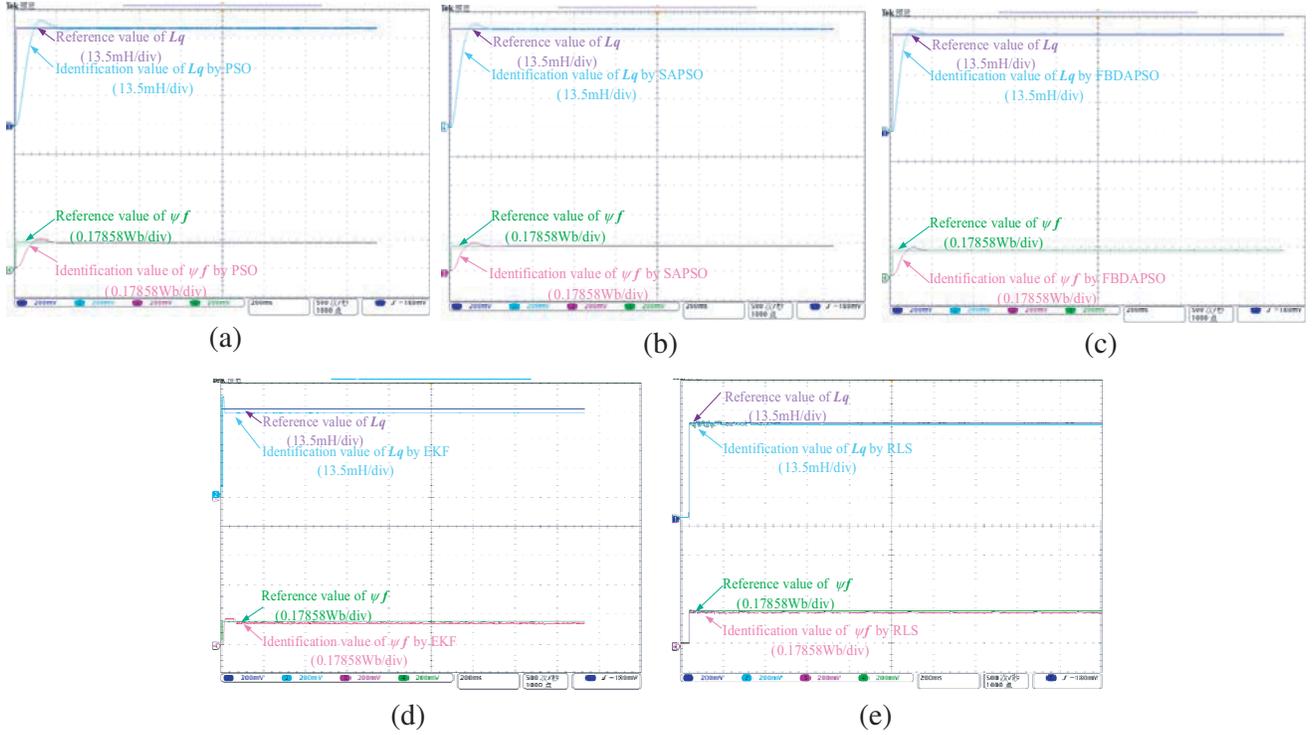


Figure 6. Identification curve of quadrature axis inductance and permanent magnet flux linkage under condition 1. (a) PSO. (b) SAPSO. (c) FBDAPSO. (d) EKF. (e) RLS.

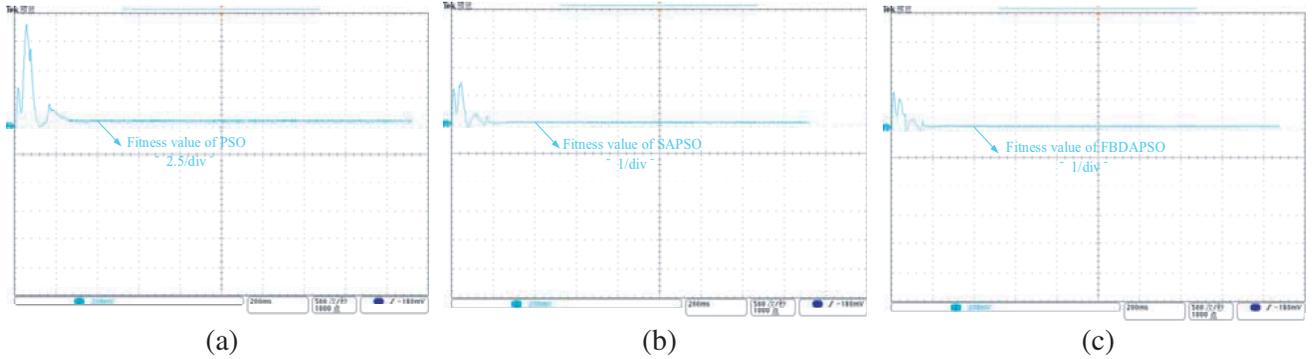


Figure 7. Fitness function value curve under condition 1. (a) PSO. (b) SAPSO. (c) FBDAPSO.

the speed of 1200 r/min. The curve of parameter identification and the curve of fitness function are shown in Figs. 8–10.

The identification curves of the stator resistance and the direct-axis inductance of the three algorithms in a more severe environment are shown in Fig. 8. EKF fluctuates the most, and its error is 6.9%. RLS also has some fluctuations, and its error becomes 4.96%. The fluctuation of error of the PSO is significantly increased, and its error becomes 4.7%. The identification curves of SAPSO and FBDAPSO show good robustness, and their error is kept within 1%, which is 0.82%, but the speed of parameter identification of FBDAPSO is still 17.5% faster than SAPSO.

The identification curves of quadrature axis inductance and permanent magnet flux linkage of the five algorithms are shown in Fig. 9. The curves of SAPSO and FBDAPSO are smoother than EKF, RLS, and PSO, which means that their fluctuations of error are smaller, and it can be seen that the speed of parameter identification of FBDAPSO is the fastest.

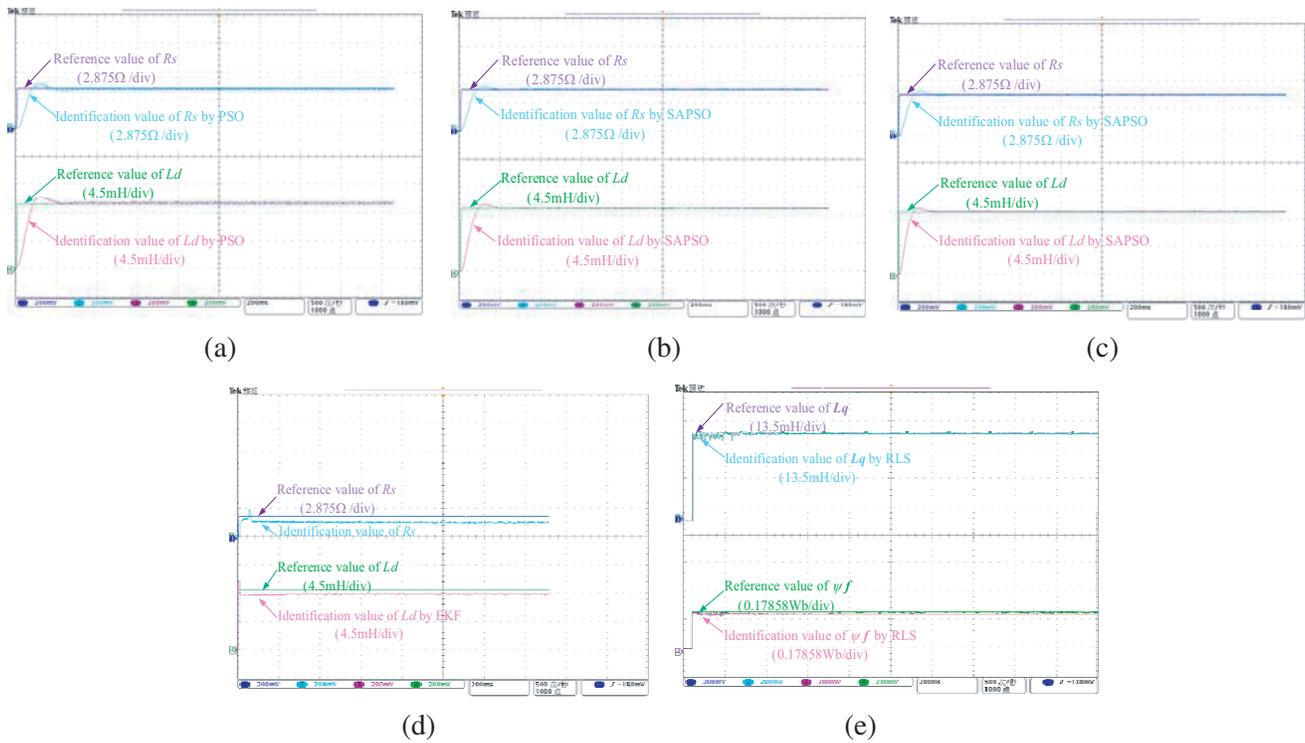


Figure 8. Identification curve of stator resistance and direct axis inductance under condition 2. (a) PSO. (b) SAPSO. (c) FBDAPSO. (d) EKF. (e) RLS.

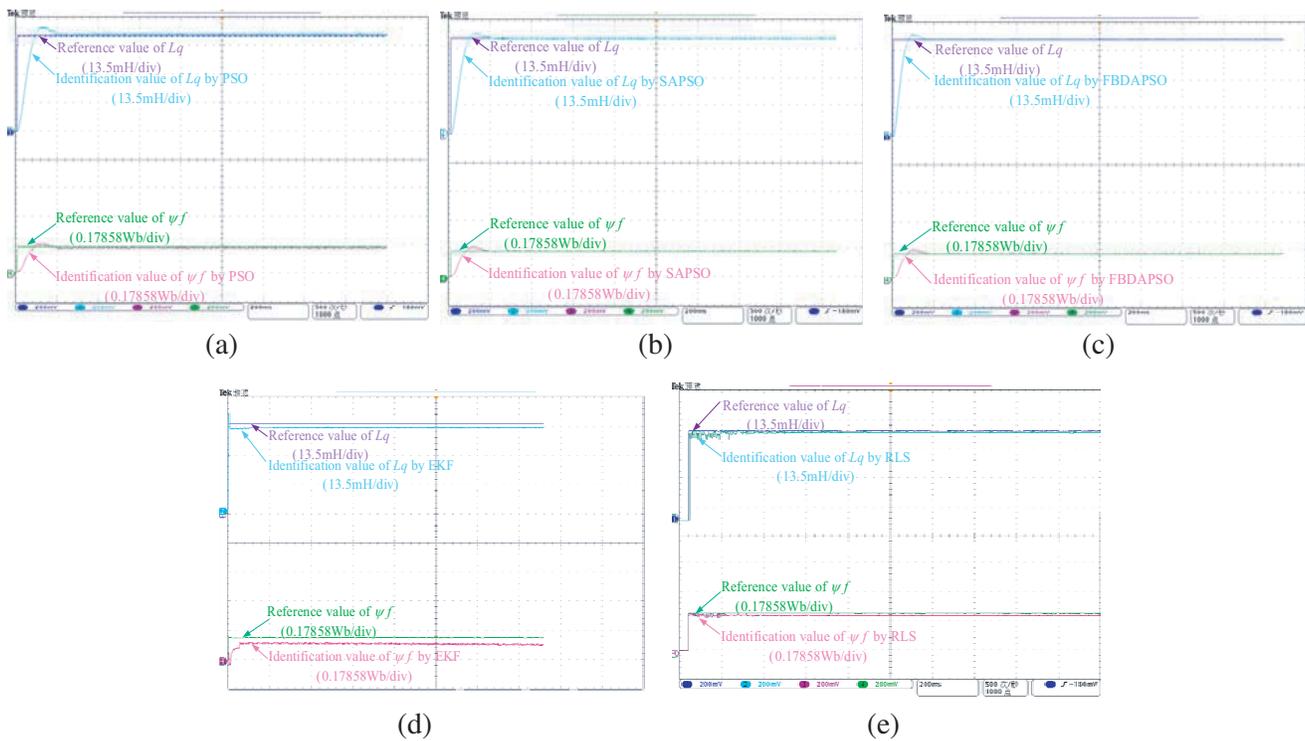


Figure 9. Identification curve of quadrature axis inductance and permanent magnet flux linkage under condition 2. (a) PSO. (b) SAPSO. (c) FBDAPSO. (d) EKF. (e) RLS.

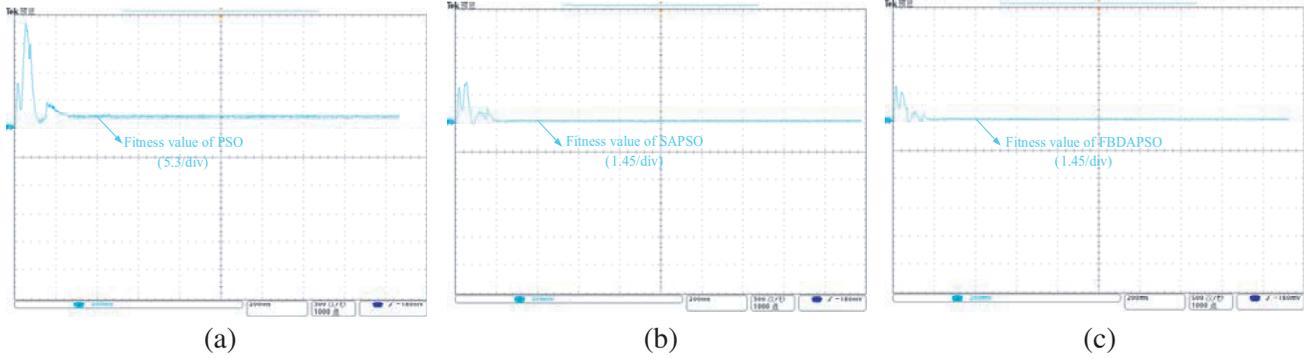


Figure 10. Fitness function value curve under condition 2. (a) PSO. (b) SAPSO. (c) FBDAPSO.

It can be seen from Fig. 10 that the curve of fitness function of PSO fluctuates greatly with the increase of disturbance, while that of FBDAPSO has less fluctuation, and the convergence speed of fitness function curve is 17.5% faster than that of SAPSO.

The above analysis shows that the accuracies and speeds of EKF, RLS, and PSO are poor in the application of parameter identification, and they need to make some improvements. The accuracy of SAPSO is close to FBDAPSO, but FBDAPSO has better convergence rate. In the case of speed and load torque changes, FBDAPSO has higher accuracy and robustness, and its identification speed is the fastest.

The identification speed and accuracy in the experiment are slightly worse than the simulation results. The main reason is that the noise and electromagnetic interference in the experiment and the errors caused by connection and transmission will have a certain impact on the accuracy and rapidity of parameter identification. In practice, electromagnetic interference is directional. Placing two devices perpendicular to each other can eliminate a lot of electromagnetic interference, then extend the distance between the devices, and add shielding devices to better eliminate electromagnetic interference.

7. CONCLUSION

Aiming at the problem of poor accuracy of PSO and long time-consuming of SAPSO, a method of motor parameter identification based on FBDAPSO is proposed. This method injects the randomness of SA into PSO and overcomes the defect that PSO is easy to fall into local optimality. In order to improve the evolution speed of the algorithm, the operation of reducing the initial annealing temperature of SAPSO is adopted, and then it performs fast tempering and annealing operations on the promising inferior solutions to ensure the accuracy of the algorithm. The algorithm termination condition follows the system running time to achieve the goal of real-time online identification of parameters. The following conclusions are drawn through simulations and experiments under different working conditions:

1) The algorithm termination condition changes with the change of the system running time, and the four parameters, motor resistance, d -axis inductance, q -axis inductance, and permanent magnet flux, can be identified online in real time.

2) For the first time to apply SAPSO to parameter identification of the PMSM, it has better motor parameter identification accuracy, evolution speed, and robustness than traditional PSO.

3) On the basis of SAPSO, the proposed FBDAPSO can accurately identify the parameters of the motor with an error less than 1%; its convergence speed is 17.5% faster than SAPSO; and it retains the accuracy and robustness of SAPSO.

ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of Hunan Province of China under Grant Number 2019JJ50119, National Natural Science Foundation of China under Grant Number

51907061 and National Engineering Laboratory of UHV Engineering Technology under Grant Number NEL202008.

REFERENCES

1. Li, X., L. Zhang, H. Ying, et al., "Study of suppression of vibration and noise of PMSM for electric vehicles," *IET Electric Power Applications*, Vol. 14, No. 7, 1274–1282, 2020.
2. Yi, P., Z. Sun, and X. Wang, "Research on PMSM harmonic coupling models based on magnetic co-energy," *IET Electric Power Applications*, Vol. 13, No. 4, 571–579, 2019.
3. Shi, T., Y. Xu, M. Xiao, et al., "VSP predictive torque control of PMSM," *IET Electric Power Applications*, Vol. 13, No. 4, 463–471, 2019.
4. Chen, Q., L. Gu, Z. Lin, et al., "Extension of space-vector-signal-injection-based MTPA control into SVPWM fault-tolerant operation for five-phase IPMSM," *IEEE Transactions on Industrial Electronics*, Vol. 67, No. 9, 7321–7333, 2020.
5. Du, J., X. Wang, and H. Lv, "Optimization of magnet shape based on efficiency map of IPMSM for EVs," *IEEE Transactions on Applied Superconductivity*, Vol. 26, No. 7, 1–7, 2016.
6. Li, L. and Q. Liu, "Research on IPMSM drive system control technology for electric vehicle energy consumption," *IEEE Access*, Vol. 7, No. 1, 186201–186210, 2019.
7. He, C. and T. Wu, "Analysis and design of surface permanent magnet synchronous motor and generator," *CES Transactions on Electrical Machines and Systems*, Vol. 3, No. 1, 94–100, 2019.
8. Perdukova, D., P. Palacky, P. Fedor, et al., "Dynamic identification of rotor magnetic flux, torque and rotor resistance of induction motor," *IEEE Access*, Vol. 8, No. 1, 142003–142015, 2020.
9. Li, X. and R. Kennel, "General formulation of Kalman-filter-based online parameter identification methods for VSI-fed PMSM," *IEEE Transactions on Industrial Electronics*, Vol. 68, No. 4, 2856–2864, 2021.
10. Yang, H., R. Yang, W. Hu, et al., "FPGA-based sensorless speed control of PMSM using enhanced performance controller based on the reduced-order EKF," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, Vol. 9, No. 7, 289–301, 2021.
11. Loria, A., E. Panteley, and M. Maghenem, "Strict lyapunov functions for model reference adaptive control: Application to lagrangian systems," *IEEE Transactions on Automatic Control*, Vol. 64, No. 7, 3040–3045, 2019.
12. Zhao, L., J. Huang, H. Liu, et al., "Second-order sliding-mode observer with online parameter identification for sensorless induction motor drives," *IEEE Transactions on Industrial Electronics (1982)*, Vol. 61, No. 10, 5280–5289, Jan. 1, 2014.
13. Raouti, D., S. Flazi, and D. Benyoucef, "Modeling and identification of electrical parameters of positive DC point-to-plane corona discharge in dry air using RLS method," *IEEE Transactions on Plasma Science*, Vol. 44, No. 7, 1144–1149, 2016.
14. Zhao, H., H. H. Eldeeb, J. Wang, et al., "Parameter identification based online noninvasive estimation of rotor temperature in induction motors," *IEEE Transactions on Industry Applications*, Vol. 57, No. 1, 417–426, 2021.
15. Shao, M., J. Huang, S. Wei, et al., "An improved PSO algorithm for parameter identification of Bouc-Wen model for piezoelectric actuator," *Proceedings of the 39th Chinese Control Conference, Shenyang, China*, 1070–1074, 2020.
16. Liu, K. and Z. Q. Zhu, "Position-offset-based parameter estimation using the adaline NN for condition monitoring of permanent-magnet synchronous machines," *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 4, 2372–2383, 2015.
17. Ortombina, L., D. Pasqualotto, F. Tinazzi, et al., "Magnetic model identification of synchronous motors considering speed and load transients," *IEEE Transactions on Industry Applications*, Vol. 56, No. 5, 4945–4954, 2020.

18. Accetta, A., F. Alonge, M. Cirrincione, et al., “GA-based off-line parameter estimation of the induction motor model including magnetic saturation and iron losses,” *IEEE Open Journal of Industry Applications*, Vol. 1, No. 5, 135–147, 2020.
19. Liu, Z., H. Wei, Q. Zhong, et al., “GPU implementation of DPSO-RE algorithm for parameters identification of surface PMSM considering VSI nonlinearity,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, Vol. 5, No. 3, 1334–1345, 2017.
20. Liu, Z., H. Wei, X. Li, et al., “Global identification of electrical and mechanical parameters in PMSM drive based on dynamic self-learning PSO,” *IEEE Transactions on Power Electronics*, Vol. 33, No. 12, 10858–10871, 2018.
21. Calvini, M., M. Carpita, A. Formentini, et al., “PSO-based self-commissioning of electrical motor drives,” *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 2, 768–776, 2015.
22. Srivastava, V., A. Stein, D. G. Rossiter, et al., “Simulated annealing with variogram-based optimization to quantify spatial patterns of trees extracted from high-resolution images,” *IEEE Geoscience and Remote Sensing Letters*, Vol. 13, No. 8, 1084–1088, 2016.
23. Pan, X., L. Xue, Y. Lu, et al., “Hybrid particle swarm optimization with simulated annealing,” *Multimedia Tools and Applications*, Vol. 78, No. 8, 29921–29936, 2019.
24. Ma, H., L. Chu, J. Guo, et al., “Cooperative adaptive cruise control strategy optimization for electric vehicles based on SA-PSO with model predictive control,” *IEEE Access*, Vol. 8, No. 21, 225745–225756, 2020.
25. Wang, J. L., Y. Li, and A. An, “Dynamic parameter identification of upper-limb rehabilitation robot system based on variable parameter particle swarm optimisation,” *IET Cyber-systems and Robotics*, Vol. 2, No. 3, 140–148, Jan. 1, 2020.
26. Liu, Z., H. Wei, Q. Zhong, et al., “Parameter estimation for VSI-fed PMSM based on a dynamic PSO with learning strategies,” *IEEE Transactions on Power Electronics*, Vol. 32, No. 4, 3154–3165, 2017.
27. Sandre-Hernandez, O., R. Morales-Caporal, J. Rangel-Magdaleno, et al., “Parameter identification of PMSMs using experimental measurements and a PSO algorithm,” *IEEE Transactions on Instrumentation and Measurement*, Vol. 64, No. 8, 2146–2154, 2015.
28. Dekkers, A. and E. H. L. Aarts, “Global optimization and simulated annealing,” *Mathematical Programming*, Vol. 50, No. 3, 367–393, 1991.