Computers, Materials & Continua
DOI: 10.32604/cmc.2023.033397
Article

Tech Science Press

check for updates

Double Update Intelligent Strategy for Permanent Magnet Synchronous Motor Parameter Identification

Shuai Zhou[1], Dazhi Wang[1,*], Mingtian Du[2], Ye Li[1] and Shuo Cao[3]

[1]School of Information Science and Engineering, Northeastern University, Shenyang, 110819, China
[2]Department of Mechanical Engineering, The University of Melbourne, Victoria, 3010, Australia
[3]College of Information Engineering, Shenyang Polytechnic College, Shenyang, 110045, China
*Corresponding Author: Dazhi Wang. Email: ProDZW@126.com
Received: 15 June 2022; Accepted: 08 September 2022

Abstract: The parameters of permanent magnet synchronous motor (PMSM) affect the performance of vector control servo system. Because of the complexity of nonlinear model of PMSM, it is very difficult to identify the parameters of PMSM. Aiming at the problems of large amount of data calculation, low identification accuracy and poor robustness in the process of multi parameter identification of permanent magnet synchronous motor, this paper proposes a weighted differential evolutionary particle swarm optimization algorithm based on double update strategy. By introducing adaptive judgment factor to control the proportion of weighted difference evolution (WDE) algorithm and particle swarm optimization (PSO) algorithm in each iteration process, and consider using PSO algorithm or WDE algorithm to update individuals according to the probability law. The individuals obtained from WDE operation are used to guide the individual evolution process in PSO operation through the information exchange mechanism. The proposed WDEPSO algorithm can ensure the diversity and effectiveness of the individual evolution of the population. The algorithm is applied to parameter identification of PMSM drive system. The simulation results show that the proposed algorithm has better convergence performance and has strong robustness, parameter identification of permanent magnet synchronous motor based on proposed method does not need to rely on more data sheet on the motor design value, can motor stator resistance identification at the same time, the rotor flux linkage, d/q-axis inductance and electrical parameters, and can effectively track the parameters value.

Keywords: PMSM; parameter identification; WDE; PSO; WDEPSO

1 Introduction

For Field-oriented control (FOC), the parameters of the current control loop will directly affect the overall performance of the system, with the stator resistance and stator inductance directly influencing the current control loop controller. In addition, the parameters of the speed and position loop controllers are also influenced by the current loop control parameters. In direct torque control (DTC), the electromagnetic torque and flux linkage are used as control variables, and the torque and chain deviations are directly controlled by a hysteresis comparator [1–4]. Numerous control techniques have been proposed to present high performance control of servo systems, and the parameters of magneto-synchronous motors can be affected by temperature, stator winding current and flux saturation during operation [3–5]. In addition, changes in electrical parameters are considered as an indication of servo system state changes; a short circuit between turns can lead to sudden changes in d/q-axis inductance and stator winding resistance [6], and demagnetization of the rotor permanent magnet can lead to a sudden decrease in the amplitude of the flux linkage [7]. Therefore, whether controlling speed, position, or torque control, the realization of high dynamic response and high precision control requires the use of accurate motor parameters, and obtaining accurate motor parameter values helps to improve the control performance of the whole servo system.

In order to obtain reliable permanent magnet synchronous motor parameters, suitable parameter identification methods are required, and the main methods for permanent magnet synchronous motor parameter identification are: model reference adaptive algorithm [8,9], recursive least squares method [10], extended Kalman filter algorithm [11], artificial neural network algorithm [12], particle swarm optimization [13,14]. With the development of machine learning [15,16], deterministic learning is applied to parameter identification of permanent magnet synchronous motor [17].

The reference [8] used the model reference adaptive algorithm to establish an adaptive observer and analyzed the large-signal convergence of the parameters using the Lyapunov second method and singular regression theory, and the designed method can improve the convergence speed and overall stability of the system but the complexity of the method implementation is high. In [9], a parameter identification method was designed using Popov's superstability theory, which can identify the stator resistance and flux linkage amplitude simultaneously with good identification accuracy, but the design of the adaptive observer and the selection of the adaptive law are more complicated. The standby algorithm is used to obtain the initial value of the inductor, and the running algorithm is used to continuously update the identification parameters. The reference [11] is based on the extended Kalman filter method for parameter identification, but the P and Q matrices are difficult to determine and are closely related to the system state. At the same time, the extended Kalman filter has some inherent drawbacks, such as susceptibility to noise, long running time, and difficulty in determining the algorithm design and objective function. In the reference [12], the stator resistance is first estimated offline, and the flux linkage and inductance are identified online using a neural network discriminator based on this estimation result, and then the obtained inductance and flux linkage values are used to further update the stator resistance value to realize the parameter identification decoupling, but this method requires a combination of offline identification and online identification, which is difficult to meet the high real-time requirements of the operating conditions; The reference [13,14] improved on the basis of the traditional particle swarm method and proposed a dynamic self-learning particle swarm optimization, which enhanced the ability of global search of the population and jumped out of the local optimum. Reference [18] proposed an improved differential evolution algorithm (DE), which introduced clonal selection and receptor editing mechanisms, to improve the population diversity and the global search ability of the algorithm. However, the algorithm was only applied to non-salient pole PMSM motor, and the parameter identification of salient pole PSMSM still needs further research.

Reference [19] proposed a parameter identification method based on Taylor series expansion of motor speed response under constant voltage input. The relationship between motor parameters and Taylor series coefficients is established by this method, but the method requires speed/position sensors and related calculation software to achieve.

In reference [20–22], particle swarm optimization algorithm, neural network algorithm and adaptive algorithm are combined for parameter identification. These methods combine the advantages of particle swarm optimization algorithm and other algorithms and have good identification accuracy in different parameter identification problems.

In reference [23–25], differential evolution algorithm and particle swarm optimization algorithm are combined and applied to parameter identification in different backgrounds. The hybrid algorithm avoids the shortcomings of the two algorithms and has good accuracy and speed in identifying various parameters.

It provides a new idea for scientific research to combine the two methods to solve engineering problems by making full use of their respective advantages. In this paper, particle swarm optimization algorithm and weighted difference algorithm are combined for parameter identification of permanent magnet synchronous motor.

The main work of this paper:

(1) The advantages of the particle swarm optimization and the weighted differential evolution algorithm are combined, and a permanent magnet synchronous motor parameter identification method that combines the use of both algorithms is proposed.

Particle swarm optimization (PSO) algorithm has a fast convergence speed in the initial stage of solving the optimization problem, but in the later stage, because all particles are close to the optimal particle, the whole population loses diversity, and particles are easy to fall into local optimal. The weighted differential evolution (WDE) algorithm has the ability to maintain population diversity and explore local search, but it has no mechanism to store previous processes and use global information about the search space, so it can easily lead to waste of computing power. Therefore, in this paper, the advantages of PSO and WDE are combined to realize the identification of PMSM parameters.

(2) An adaptive judgment factor is proposed to control the ratio between the particle swarm optimization and the weighted differential evolution algorithm.

Adaptive judgment factors are introduced to control the proportion of particle swarm optimization and weighted differential evolution algorithm in each iteration process. According to the probability law, PSO algorithm or WDE algorithm is used to update individuals. The individuals obtained by WDE operation are used to guide the evolution process of individuals in PSO operation through information exchange mechanism, so as to ensure that the greater the crossover probability, the more information the new individuals inherit from the mutant individuals, and the richer the population diversity, thus ensuring the global solution accuracy and efficiency of the algorithm.

2 Mathematical Model of Permanent Magnet Synchronous Motor in Servo System

The permanent magnet synchronous motor has multivariable, nonlinear, and strongly coupled characteristics, and the mathematical model of the permanent magnet synchronous motor in the synchronous rotating coordinate system (d/q-axis coordinate system) can be expressed as (1) [26–28], after neglecting disturbances such as core saturation, harmonics, eddy currents, and losses caused by hysteresis.

$$\begin{cases} u_d = Ri_d + L_d \dfrac{di_d}{dt} - \omega_e L_q i_q \\ u_q = Ri_q + L_q \dfrac{di_q}{dt} + \omega_e \left(L_d i_d + \psi_f\right) \end{cases} \tag{1}$$

where u_d is the stator straight-axis voltage, u_q is the stator cross-axis voltage; R is the stator resistance; i_d is the stator straight-axis current, i_q is the stator cross-axis current; L_d is the stator straight-axis inductance, L_q is the stator cross-axis inductance; ω_e is the electric angular velocity; ψ_f is the flux linkage. When the servo system is in steady-state operation, the stator cross-axis and straight-axis currents and rotational speed of the permanent magnet synchronous motor vary very little, which is approximated as:

$$\frac{di_d}{dt} = \frac{di_q}{dt} = 0 \tag{2}$$

Eq. (1) changes to

$$\begin{cases} u_d = Ri_d - \omega_e L_q i_q \\ u_q = Ri_q + \omega_e \left(L_d i_d + \psi_f\right) \end{cases} \tag{3}$$

The order of the Eq. (3) is two, and there are four motor parameters to be identified, so the Eq. (3) is of non-full-rank type. The variables such as stator resistance, rotor chain and winding inductance change with load and temperature during motor operation, which may lead to inaccurate identification of the parameters to be identified. Permanent magnet synchronous motors generally use $i_d = 0$ control strategy to reduce motor losses and to improve the system power factor. To solve the problem of equation under-ranking in Eq. (3), a negative-sequence weak magnetic current strategy with $i_d \neq 0$ injected in the d-axis for a short time is usually used. A full-rank permanent magnet synchronous motor system model [12] can be obtained under both $i_d = 0$ and $i_d \neq 0$ strategies in the discrete form:

$$\begin{cases} u_{d0}(k) = -L_{q0}\omega_{e0}(k)\, i_{q0}(k) \\ u_{q0}(k) = Ri_{q0}(k) + \psi_m \omega_{e0}(k) \\ u_{d1}(k) = Ri_{d1}(k) - L_{q1}\omega_{e1}(k)\, i_{q1}(k) \\ u_{q1}(k) = Ri_{q1}(k) - L_{d1}\omega_{e1}(k)\, i_{d1}(k) + \psi_m \omega_{e1}(k) \end{cases} \tag{4}$$

the variables and parameters subscripted as "0" in Eq. (4) represent the sampled values in the $i_d = 0$ model, and the variables or parameters subscripted as "1" represent the sampled values in the $i_d \neq 0$ model.

3 Algorithm Design

3.1 Weighted Differential Evolutionary Algorithm

The differential evolution algorithm was proposed by Rainer Storn and Kenneth Price in 1995 while studying the Chebyshev polynomial fitting problem [29]. The differential evolution algorithm and process can be summarized as follows: population initialization, mutation operation, crossover operation, and selection operation. Assuming that the current population size is NP and the individual dimension is D the basic principle is as follows.

Step 1: population initialization. The current population can be described as follows:

$$P_t = \left\{X_{i,t} | X_{i,t} = \left(X_{i,t}{}^1, X_{i,t}{}^2, X_{i,t}{}^3 \cdots, X_{i,t}{}^D\right)^T\right\}, i \in [1, NP] \tag{5}$$

where t denotes the current evolutionary generation and $X_{i,t}$ denotes the i-th individual vector in the population.

$$X_i = \left(x_t{}^1, x_t{}^2 \cdots x_t{}^D\right) \tag{6}$$

$$\begin{cases} x_{i,0}{}^{j} = x_{\min}{}^{j} + rand\,(0,1)\,(x_{\max}{}^{j} - x_{\min}{}^{j}) \\ i \in [1, NP], j \in [1, D] \end{cases} \tag{7}$$

where, $rand\,(0,1)$ denotes a random small number that conforms to a uniform distribution in the range (0,1), and $X^{j}_{\max}$, $X^{j}_{\min}$ denote the maximum and minimum bounds of the j-th dimension of the individual, respectively.

Step 2: variant operation. Eq. (8) gives a simpler variation strategy:

$$V_{i,t} = X_{r1,t} + F\left(_{Xr2,t} - X_{r3,t}\right) \tag{8}$$

where r_1, r_2 , r_3 populations of three randomly selected individual numbers, and each of the three is not equal, F is the scaling factor, used to control the search step, and its value range (0,1).

Step 3: crossover operations. Eq. (9) gives a crossover operation to obtain variant individuals by comparing the crossover factor CR with a uniformly distributed random number in the range (0,1), the variable individual $u^{j}_{i,t}$ is obtained, and the value of CR ranges from 0 to 1.

$$u^{j}_{i,t} = \begin{cases} v^{j}_{i,t}, rand\,(0,1) < CR \\ x^{j}_{i,t}, \text{else} \end{cases} \tag{9}$$

Step 4: select operation. The differential evolution algorithm uses a greedy strategy to compare the fitness of offspring individuals over their parents, and the individuals with better fitness will be selected to enter the next generation population.

$$x^{k+1}_{i} = \begin{cases} u^{k+1}_{i}, f\left(u^{k+1}_{i}\right) < f\left(x^{k}_{i}\right) \\ x^{k}_{i}, \text{else} \end{cases} \tag{10}$$

where f is the fitness function.

The weighted differential evolution algorithm (WDE) [30] was proposed in 2018. WDE is based on a two-population variation strategy, and the weighting operation is introduced in the variation operator to achieve parameter-free adjustment of the optimization process. A better balance between local and global search is sought. The process of weighted difference evolution algorithm is as follows.

Step 1: population initialization. The initial WDE population contains $2NP$ individuals of D-dimensional variables, and the population P_{i_0,j_0} is initialized as:

$$P_{i_0,j_0} \sim U\left(\min_{j_0}, \min_{i_0}\right), i_0 \in [1, 2NP], j_0 \in [1, D] \tag{11}$$

In Eq. (11), NP is the number of variable individuals in the population in run mode, D is the dimension of the individuals, U is the initial parameter range, and $\min_{j_0}, \min_{i_0}$ are the lower and upper limit values of the parameters, respectively.

Step 2: weighted operation. WDE algorithm randomly selects NP subpopulation P_{sub1} from $2NP$ individuals, and the remaining individuals form subpopulation P_{sub2}. In the evolution process of each generation, each dimension of continuous variables in P_{sub1} is weighted to generate temporary population $x^{j}_{i(temp)}$ according to Eq. (12).

$$x^{j}_{i(temp)} = \sum_{i=1}^{NP} \omega_i x^{j}_{i\left(P_{sub1}\right)} \Big/ \sum_{i=1}^{NP} \omega_i, i \in [1, NP], j \in [1, D] \tag{12}$$

where ω_i is determined by Eq. (13).

$$\omega_i = [rand\,(0,1)]^3 \tag{13}$$

Step 3: cross-mutation operation. The crossover factor E_i^j of individuals in subpopulation P_{sub2} is 0, and the crossover factor of $D \times [rand\,(0,1)]^3$ variables in randomly selected individuals is set to 1, indicating that the crossover variation operation generates trial solutions.

One of the following two strategies is selected according to the probability of each pair to perform the variation operation.

Strategy 1. The variation factor is defined according to Eq. (14), for different dimension variables $x_i\,(i = 1, 2, 3 \cdots D)$ of the same individual, the variation factor CF^j takes the same value, and the operation of cross variation is carried out according to Eq. (15).

$$CF^j = [rand\,(0,1)]^3 \tag{14}$$

$$v^j_{i(P_{sub2})} = x^j_{i(P_{sub2})} + CF^j E_i^j \left(x^j_{i(temp)} - x^j_{i(P_{sub2})}\right) \tag{15}$$

Strategy 2. Generate different variation factors CF_i for each dimension of individual variable $x_i^j\,(j = 1, 2, 3 \cdots NP)$, respectively, by calling random numbers in Eq. (16), and then perform the cross-variation operation according to Eq. (17).

$$CF_i = [rand\,(0,1)]^3 \tag{16}$$

$$v^j_{i(P_{sub2})} = x^j_{i(P_{sub2})} + CF_i E_i^j \left(x^j_{i(temp)} - x^j_{i(P_{sub2})}\right) \tag{17}$$

Step 4: select operation. The differential evolution algorithm uses a greedy strategy to calculate the fitness values of individuals before and after the mutation, and selects the better solution with smaller fitness values to update and replace the initial values.

$$x_i^{k+1} = \begin{cases} v_i^{k+1}, f\left(v_i^{k+1}\right) < f\left(x_i^k\right) \\ x_i^k, \text{else} \end{cases} \tag{18}$$

where f is the fitness function.

From the algorithm steps, it can be seen that the weighting operation is an important difference between WDE and DE. Through the weighting operation of Eq. (12), individuals will synchronize the information of other individuals in each round of mutation is and increase the information exchange between individuals.

3.2 Particle Swarm Optimization Algorithm

The particle swarm optimization is inspired by observing the foraging messaging of a flock of birds. In the algorithm, each individual is a particle representing a feasible solution. Let Z_i be the position of the i-th particle, and the adaptation value of Z_i is calculated according to the previously set adaptation function, which is used to measure the merit of this particle position. V_i is the flight speed of particle i, i.e., the distance travelled by the particle per unit time. P_i is the optimal position found by particle i, and P_g is the optimal position found by the whole particle swarm [31].

In each iteration of the particle, its update equation is:

$$\begin{cases} V_i^{k+1} = \omega V_i^k + c_1 rand\,()\left(P_i - Z_i^k\right) + c_2 rand\,()\left(P_g - Z_i^k\right) \\ Z_i^{k+1} = Z_i^k + V_i^{k+1} \end{cases} \tag{19}$$

where K is the number of iterations; ω, c_1, c_2 are the inertia weight coefficients and learning factors, respectively; $rand\,()$ is a random number between 0 and 1. PSO has strong small data seeking ability and can solve more complex optimization problems, but with the increase of the number of iterations, it is easy to fall into local optimum.

3.3 WDEPSO Algorithm

The evolution criterion of the differential evolution algorithm is based on adaptive information and does not require additional conditions such as function derivability and continuity. In addition, differential evolution is inherently parallel and suitable for massively parallel distributed processing. However, differential evolution algorithms do not utilize individual prior knowledge, i.e., there is no mechanism to store prior processes and use global information about the search space. The particle swarm optimization algorithm, on the other hand, makes decisions based on its own and other particles' experience, so it can effectively compensate for the deficiencies of the differential evolution algorithm. The WDE algorithm and PSO algorithm are fused, and an adaptive judgment factor is introduced to control the ratio between the use of particle swarm optimization and differential evolution algorithms in each iteration, and the PSO algorithm or WDE algorithm is considered to update individuals according to the law of probability, which ensures that the greater the crossover probability, the more information the new individuals inherit from the mutant individuals, and the richer the population diversity. Judgment factor is used to select the update method of individuals, it is calculated as:

$$\lambda_G = \lambda_0 \frac{2}{\pi} \operatorname{arccot} \frac{G_{\max} - G + 1}{G^2} \tag{20}$$

where $G_{\max}$ is the total number of iterations; G is the current evolutionary generation and $G \in [1, G_{\max}]$; λ_0 is a constant.

Then when the i-th individual generates a new individual, first a random number $S_i \in [0, 1]$ is generated, and if S_i is smaller than λ_G, the individual is updated using the WDE algorithm, otherwise the particle swarm optimization algorithm is used for updating.

According to Eq. (20), the judgment factor gradually increases with the number of iterations, so that the solution space is mainly searched by the PSO algorithm during the preliminary search, while the WDE algorithm is mainly used for fine search at the later stage of the search to find the non-inferior solution set with high accuracy. In addition, the WDE algorithm in the preliminary stage of S_i search can perform variational operations on individuals to prevent the particle swarm optimization algorithm from falling into local optimal solutions when searching. Eq. (20) combined with the role of random numbers makes it possible to update individuals in a probabilistic way, ensuring that the ratio of the two algorithms used in each iteration better conforms to the statistical law, and giving full play to the role of the judgment factor in regulating the intelligent evolutionary algorithm of the population. Fig. 1a is the change curve of λ_G, Fig. 1b is Parameter identification block diagram of permanent magnet synchronous motor based on WDEPSO.

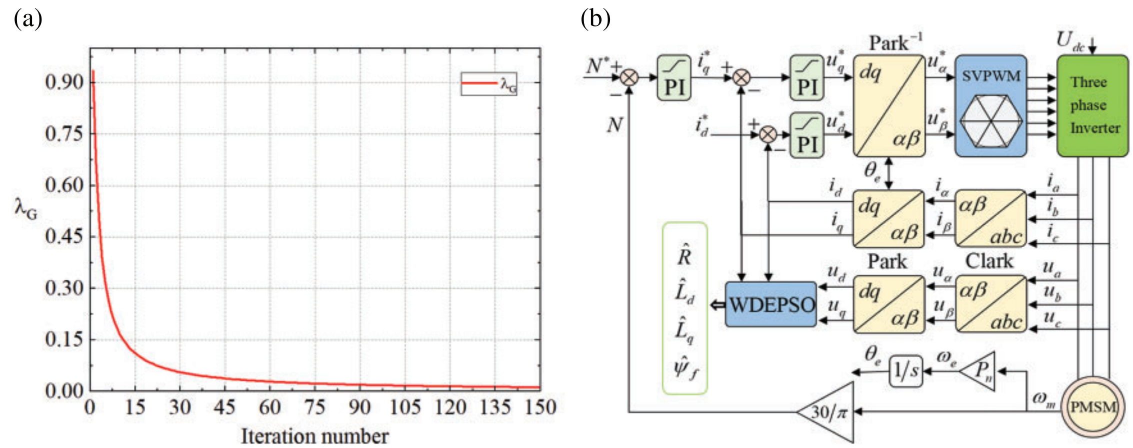

Figure 1: (a) curve of λ_G; (b) PMSM parameter identification based on WDEPSO

3.4 Fitness Function

The basic idea is that according to the difference between the actual output of the system and the output of the adjustable model, the parameters of the model to be identified are continuously adjusted by the identification algorithm to minimize the value of the error adaptation function between the actual output value and the adjustable model, the smaller the adaptation value, the closer the input of the identification model and the measured input, and the closer the parameters to be identified and the real value.

The fitness function is:

$$f = \sum_{k=1}^{n} \left[a_1 \left(u_{d0}(k) - \hat{u}_{d0}(k) \right)^2 + a_2 \left(u_{d1}(k) - \hat{u}_{d1}(k) \right)^2 + a_3 \left(u_{q0}(k) - \hat{u}_{q0}(k) \right)^2 + a_4 \left(u_{q1}(k) - \hat{u}_{q1}(k) \right)^2 \right] \tag{21}$$

where a_1, a_2, a_3, a_4 are the weight coefficients; $\hat{u}_{d0}(k)$, $\hat{u}_{d1}(k)$ are the d-axis output voltage values of the stator of the identification model; $\hat{u}_{q0}(k)$, $\hat{u}_{q1}(k)$ are the q-axis output voltage values of the stator of the identification model. Under the same adaptation value, different weighting coefficients can change the accuracy of each parameter to be identified. Fig. 2a shows the identification principle of WDEPSO.

3.5 PMSM Parameter Identification Based on WDEPSO

Parameter identification steps of permanent magnet synchronous motor based on WDEPSO:

Step 1: sampling current and angular velocity.

Step 2: initialize the two algorithms.

Step 3: judge the relationship between λ_G and S_i.

Step 4: select one of the two algorithms for iteration according to the results of step 3.

Step 5: if the maximum number of iterations is not reached, re-enter step 3 for circulation.

Step 6: reach the maximum number of iterations and output identification parameters.

The algorithm flow is shown in Fig. 2b.

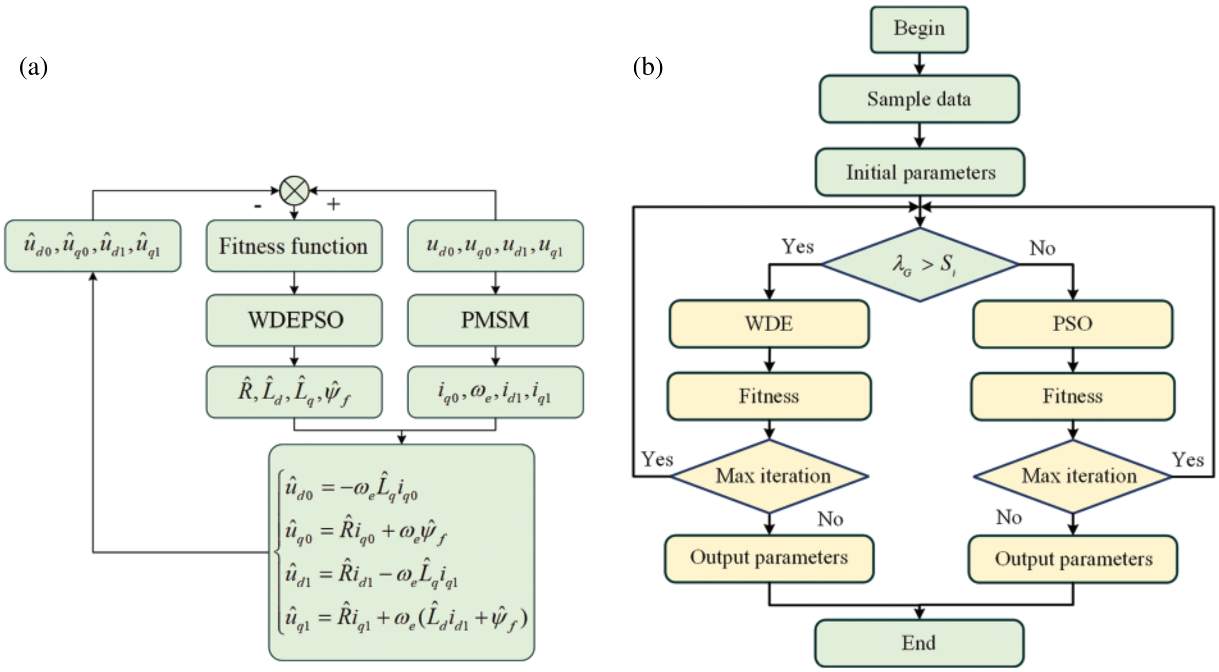

Figure 2: (a) Principle of WDEPSO; (b) Flow chart of the WDEPSO

4 Simulation Results and Analysis

In this simulation, a simulation model based on the WDEPSO is built on the MATLAB/Simulink platform for the parameter identification of permanent magnet synchronous motor. The parameters of the simulation model are set as in Table 1.

Table 1: Design parameters and specification of PMSM

Symbol	Quantity	Value
P	Rated power	1.0 kW
N	Rated speed	2000 r · min^{-1}
T	Nominal torque	4 N · m
U	Rated line voltage	220 V
R	Stator resistance	1.454 Ω
L_d	d-axis inductance	7.53 mH
L_q	q-axis inductance	13.25 mH
ψ_f	Flux linkage	0.224 Wb
p_n	Number of pole pairs	4

To make the results comparable, the WDEPSO is compared with the WDE, DE, PSO, and particle swarm optimization with inertial weight variation (LPSO). To ensure the rationality of simulation, the

WDEPSO, PSO, LPSO are set with the same parameters, and the WDE and DE algorithms are set with the same crossover factor and variation factor. In order to test the performance of the algorithms in global search, the initial domain of all parameters to be identified is set to (−2, 10), far from the real values set in the simulation. The simulations were run 50 times independently in order to reduce the testing error caused by the randomness of the single algorithm.

In order to fully verify the computational performance of various algorithms, the effectiveness of algorithm recognition is tested under changing working conditions. The algorithms are tested under two conditions: load change and speed change. The speed of the permanent magnet synchronous motor is designed to first accelerate to the rated speed, then run steadily, and finally decelerate to 20% of the rated speed until the end. Each speed change strategy accounts for one-third of the total sampling time, and the motor load increases from 0 to the rated value during stable operation, and decreases to 0 when the motor starts to decelerate.

The simulation data are presented in the following Table 2.

Table 2: Result comparisons among five algorithms

Identified parameter	WDE	DE	LPSO	PSO	WDEPSO
R/Ω	1.447	1.544	1.438	1.438	1.453
L_d/mH	6.829	7.554	7.484	7.484	7.541
L_q/mH	17.05	12.70	13.50	14.89	13.43
ψ_f/Wb	0.2299	0.2227	0.2295	0.2273	0.2242
Fitness	15210	20570	5659	11005	45.9

From the data in the Table 2, it can be seen that the WDEPSO has the highest recognition accuracy and the average value of the fitness function is smaller, and the algorithm performs better than several other algorithms.

The results of the five algorithms to identify each parameter of the permanent magnet synchronous motor are shown in Figs. 3∼5.

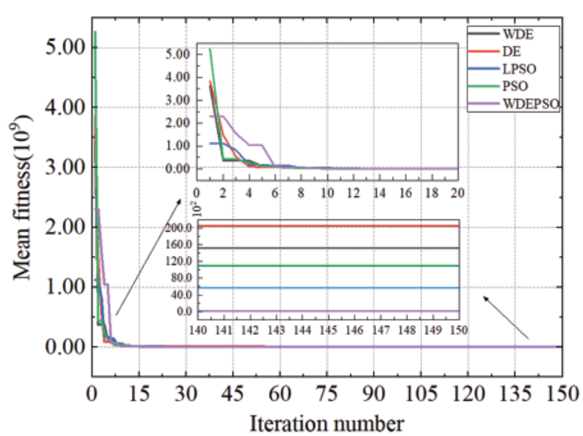

Figure 3: Plot of mean fitness

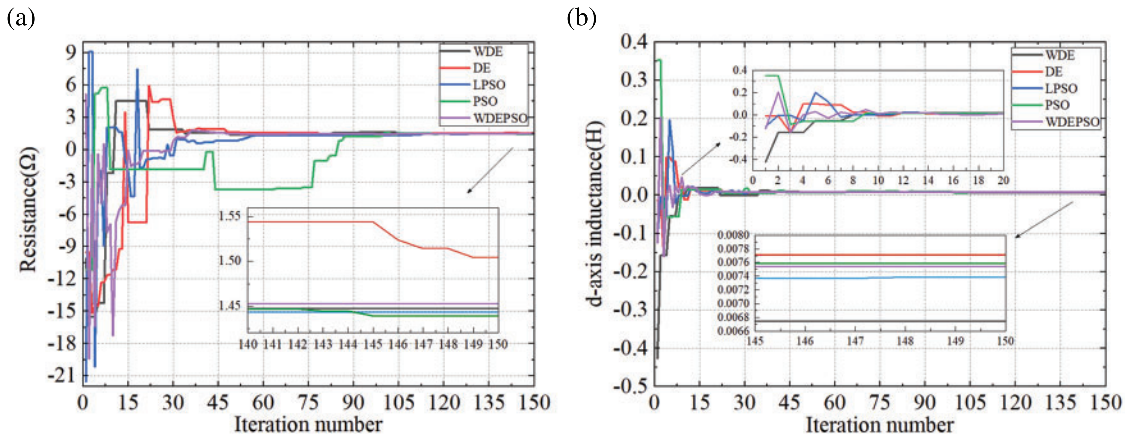

Figure 4: (a) Plot of stator resistance; (b) Plot of d-axis inductance

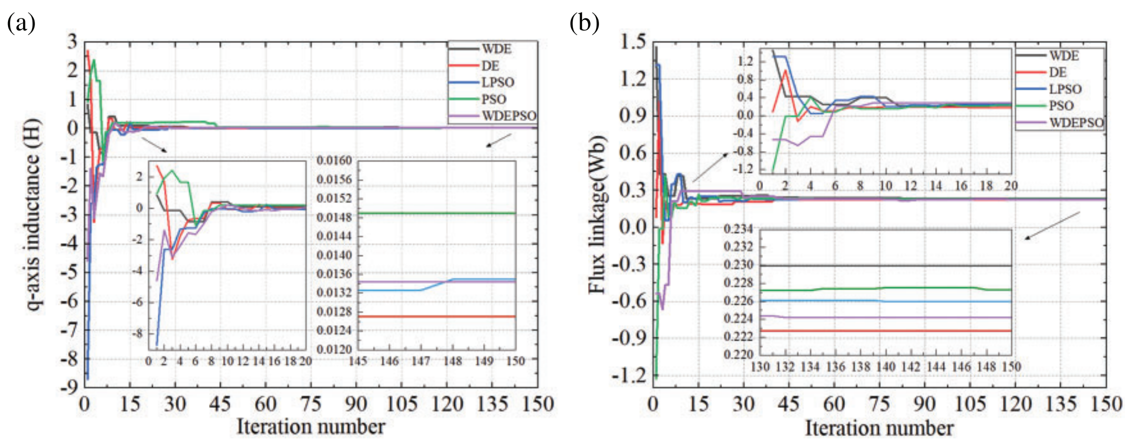

Figure 5: (a) Plot of q-axis inductance; (b) Plot of flux linkage

From Fig. 3, it can be seen that WDEPSO enters the local search around the 15th iteration, and the other four algorithms start to enter the local search around the 5th generation. From the convergence speed, the other four algorithms are faster, but from the data after the final iteration is completed, it can be seen that the other four algorithms converge faster, but the final error is larger, because the other four algorithms fall into the local search and cannot jump out. The slow convergence of the WDEPSO algorithm at the beginning of the iteration is to avoid getting into local optimum. This verifies the better global search performance of the WDEPSO algorithm and the ability to avoid getting into local optimal solutions.

Fig. 4a shows the identification curve of stator resistance. From the figure, it can be seen that the five algorithms fluctuate more at the beginning of the identification process, because the stator resistance is more affected after the motor operation, and the algorithm takes longer time to carry out the identification. Compared with the other four algorithms, WDEPSO converges faster and starts to converge at the 50th iteration, and the resistance identification value fluctuates less during the iteration, and the final identification result is closest to the set value, which verifies the ability of WDEPSO algorithm to find the optimal solution.

Fig. 4b shows the d-axis inductance identification curve. From the Fig. 4b, it can be seen that the five algorithms have less curve fluctuation and converge faster during the identification process, which indicates two things: firstly, the d-axis inductance is less affected when the motor is running, so it is easier for the five algorithms to find the optimal solution; secondly, the d-axis inductance is more affected by the operating conditions of the motor, but the algorithms have stronger identification ability and can accurately identify the inductance value. The WDEPSO does not converge faster than the other algorithms, but the final identification results show that the WDEPSO algorithm is closest to the set value. This confirms that WDEPSO has better solving ability.

Fig. 5a shows the identification curve of q-axis inductance values. From the Fig. 5a, it can be seen that the WDEPSO is gradually approaching the optimal solution from the beginning of the iteration to entering the local search process, and the identification results do not fluctuate significantly, while other algorithms fluctuate more in the beginning of the search process, which fully verifies the better global search capability of the WDEPSO.

Fig. 5b shows the flux linkage identification curve. From the Fig. 5b, it can be seen that the WDEPSO starts to converge at about the 40th iteration and then refines the search near the set value. Compared with several other algorithms, the convergence speed of WDEPSO has no obvious advantage in flux linkage identification, but the final identification result is closest to the set value, which verifies the stronger identification accuracy of the WDEPSO.

Parameter identification of PMSM is subject to large fluctuations or mis-convergence due to the high degree of nonlinearity of the PMSM model and the local extreme value points of the objective function, which makes it difficult to find the optimal solution for algorithms with low search degree, small solution space and poor stability. All five algorithms in this paper can converge quickly to near the true value, which indicates that all five algorithms have good global search performance. When the identified values of all five algorithms are close to the stable values, it can be seen that the WDEPSO converges faster than the other algorithms, and the identification accuracy is higher than the other algorithms, and it can be seen from the data in Table I that the identified values of the WDEPSO are closer to the design values, the average fitness value is smaller, and the WDEPSO can better track the motor in speed and load torque mutation. This shows that the WDEPSO has good robustness and convergence, which verifies the superiority of the algorithm.

5 Conclusion

In this paper, based on vector control, the motor electromechanical mathematical model is brought to full rank by injecting weak magnetic negative sequence current with $i_d \neq 0$. The double update strategy combining weighted differential evolution algorithm based on initial parameters and particle swarm optimization is used to accurately identify four parameters, namely, stator winding resistance, d-axis inductance, q-axis inductance and flux linkage, an adaptive judgment factor is introduced to control the ratio of WDE and PSO algorithms during each iteration. The adaptive judgment factor is introduced to control the proportion of WDE and PSO algorithm calls in each iteration, and the PSO algorithm or WDE algorithm is considered to update the individuals according to the probability law to ensure the diversity and effectiveness of the evolution of individuals in the population. By comparing the algorithm with other four algorithms under different operating conditions, it is proved that the algorithm has good robustness and high computational accuracy in the parameter identification of permanent magnet synchronous motor. When PMSM runs at high speed or low speed, the resistance identification results are affected by the voltage-source-inverter (VSI) nonlinearity, while the d-q axis inductance and flux linkage are affected when the motor runs at low speed. In the next step, the voltage-source-inverter (VSI) nonlinearity on PMSM parameter identification will be considered.

Funding Statement: This work was supported in part by the Natural Science Foundation of China under Grant 52077027, and in part by the Liaoning Province Science and Technology Major Project No.2020JH1/10100020.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Z. Q. Zhu, D. Liang and K. Liu, "Online parameter estimation for permanent magnet synchronous machines: An overview," *IEEE Access*, vol. 9, no. 8, pp. 59059–59084, 2021.
[2] Z. Wang, J. Chai, X. Xiang, X. Sun and H. Lu, "A novel online parameter identification algorithm designed for deadbeat current control of the permanent-magnet synchronous motor," *IEEE Transactions on Industry Applications*, vol. 58, no. 2, pp. 2029–2041, March–April 2022.
[3] Y. Yu, X. Huang, Z. Li, M. Wu, T. Shi *et al.,* "Full parameter estimation for permanent magnet synchronous motors," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 5, pp. 4376–4386, May 2022.
[4] M. S. Rafaq and J. Jung, "A comprehensive review of state-of-the-art parameter estimation techniques for permanent magnet synchronous motors in wide speed range," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4747–4758, July 2020.
[5] H. Ahn, H. Park, C. Kim and H. Lee, "A review of state-of-the-art techniques for PMSM parameter identification," *Journal of Electrical Engineering and Technology*, vol. 15, no. 3, pp. 1177–1187, 2020.
[6] Y. Xu, Z. Zhang, Y. Jiang, J. Huang and W. Jiang, "Numerical analysis of turn-to-turn short circuit current mitigation for concentrated winding permanent magnet machines with series and parallel connected windings," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 11, pp. 9101–9111, 2020.
[7] K. C. Kim, K. Kim, H. J. Kim and J. Lee, "Demagnetization analysis of permanent magnets according to rotor types of interior permanent magnet synchronous motor," *IEEE Transactions on Magnetics*, vol. 45, no. 6, pp. 2799–2802, Jun. 2009.
[8] T. Boileau, N. Leboeuf, B. Nahid-Mobarakeh and F. Meibody-Tabar, "Online identification of PMSM parameters: Parameter identifiability and estimator comparative study," *IEEE Transactions on Industry Applications*, vol. 47, no. 4, pp. 1944–1957, Jul. 2011.
[9] K. H. Kim, S. K. Chung, G. W. Moon, I. C. Baik and M. -J. Youn, "Parameter estimation and control for permanent magnet synchronous motor drive using model reference adaptive technique," in *Proc. of IECON '95-21st Annual Conf. on IEEE Industrial Electronics*, Orlando, FL, USA, vol. 1, pp. 387–392, 1995.
[10] S. Kallio, J. Karttunen, P. Peltoniemi, P. Silventoinen and O. Pyrhönen, "Online estimation of double-star IPM machine parameters using RLS algorithm," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 9, pp. 4519–4530, Sep. 2014.
[11] Y. Shi, K. Sun, L. Huang and Y. Li, "Online identification of permanent magnet flux based on extended Kalman filter for IPMSM drive with position sensorless control," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4169–4178, 2012.
[12] K. Liu, Q. Zhang, J. Chen, Z. Q. Zhu and J. Zhang, "Online multiparameter estimation of nonsalient-pole PM synchronous machines with temperature variation tracking," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1776–1788, 2011.
[13] Z. H. Liu, H. L. Wei, Q. C. Zhong, K. Liu and X. -S. Xiao, "Parameter estimation for VSI-fed PMSM based on a dynamic PSO with learning strategies," *IEEE Transactions on Power Electronics*, vol. 32, no. 4, pp. 3154–3165, Apr. 2017.

[14] Z. H. Liu, H. L. Wei, X. H. Li, K. Liu and Q. C. Zhong, "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, pp. 10858–10871, 2018.

[15] H. Sun and R. Grishman, "Lexicalized dependency paths based supervised learning for relation extraction," *Computer Systems Science and Engineering*, vol. 43, no. 3, pp. 861–870, 2022.

[16] H. Sun and R. Grishman, "Employing lexicalized dependency paths for active learning of relation extraction," *Intelligent Automation & Soft Computing*, vol. 34, no. 3, pp. 1415–1423, 2022.

[17] W. Yu, H. Liang, X. Dong and Y. Luo, "Modeling and identification of permanent magnet synchronous motor via deterministic learning," *IEEE Access*, vol. 8, no. 6, pp. 168516–168525, 2020.

[18] K. Lam, G. Feng and Y. Han, "Identification of PMSM parameters based on differential evolution algorithm with multiple evolutionary strategies," in *2021 China Automation Congress (CAC)*, Beijing, China, pp. 4797–4802, 2021.

[19] W. Wu, "DC motor identification using speed step responses," in *Proc. of the 2010 American Control Conf.*, Baltimore, MD, USA, pp. 1937–1941, 2010.

[20] L. Nikakhtar, S. Zare, H. M. Nasirabad and B. Ferdosi, "Application of ANN-PSO algorithm based on FDM numerical modeling for back analysis of EPB TBM tunneling parameters," *European Journal of Environmental and Civil Engineering*, vol. 26, no. 8, pp. 3169–3186, 2020.

[21] F. Zha, W. Sheng, W. Guo, S. Qiu, J. Deng *et al.,* "Dynamic parameter identification of a lower extremity exoskeleton using RLS-PSO," *Applied Sciences*, vol. 9, no. 2, pp. 324–341, 2019.

[22] J. Chrouta, A. Zaafouri and M. Jemli, "System identification using gray-based adaptive heterogeneous multi-swarm PSO algorithm: Application to an irrigation station," *Journal of Circuits, Systems and Computers*, vol. 27, no. 4, pp. 1850059–1850086, 2017.

[23] H. Yang, J. Li, B. Liu and L. Chen, "Identification of source information for sudden hazardous chemical leakage accidents in surface water on the basis of particle swarm optimisation, differential evolution and metropolis-hastings sampling," *Environmental Science and Pollution Research*, vol. 28, no. 47, pp. 67292–67309, 2021.

[24] S. Patel, T. Eldho and A. Rastogi, "Hybrid-metaheuristics based inverse groundwater modelling to estimate hydraulic conductivity in a nonlinear real-field large aquifer system," *Water Resources Management*, vol. 34, no. 6, pp. 2011–2028, 2020.

[25] B. Pourasghar, M. A. Ahandani and H. Kharrati, "Parameter identification of engineering problems using a differential shuffled complex evolution," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2749–2782, 2020.

[26] T. Sebastian, G. Slemon and M. Rahman, "Modelling of permanent magnet synchronous motors," *IEEE Transactions on Magnetics*, vol. 22, no. 5, pp. 1069–1071, 2015.

[27] J. Y. Lee, S. H. Lee, G. H. Lee, J. P. Hong and J. Hur, "Determination of parameters considering magnetic nonlinearity in an interior permanent magnet synchronous motor," *IEEE Transactions on Magnetics*, vol. 42, no. 4, pp. 1303–1306, Apr. 2006.

[28] F. Fernandez-Bernal, A. Garcia-Cerrada and R. Faure, "Determination of parameters in interior permanent-magnet synchronous motors with iron losses without torque measurement," *IEEE Transactions on Industry Applications*, vol. 37, no. 5, pp. 1265–1272, Sep. 2001.

[29] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[30] P. Civicioglu, E. Besdok, M. A. Gunen and U. H. Atasever, "Weighted differential evolution algorithm for numerical function optimization: A comparative study with cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms," *Neural Computing and Applications*, vol. 32, no. 8, pp. 3923–3937, Apr. 2020.

[31] J. Kennedy and R. Eberhart, "Particleswarm optimization," in *Proc. of ICNN'95-Int. Conf. on Neural Networks*, Perth, WA, Australia, vol. 4, pp. 1942–1948, 1995.