



ARTICLE

## Improved Particle Swarm Optimization for Parameter Identification of Permanent Magnet Synchronous Motor

Shuai Zhou<sup>1</sup>, Dazhi Wang<sup>1,\*</sup>, Yongliang Ni<sup>2</sup>, Keling Song<sup>2</sup> and Yanming Li<sup>2</sup>

<sup>1</sup>School of Information Science and Engineering, Northeastern University, Shenyang, 110819, China

<sup>2</sup>China Northern Vehicle Research Institute, Beijing, 100072, China

\*Corresponding Author: Dazhi Wang. Email: ProDZW@126.com

Received: 20 December 2023 Accepted: 14 March 2024 Published: 15 May 2024

### ABSTRACT

In the process of identifying parameters for a permanent magnet synchronous motor, the particle swarm optimization method is prone to being stuck in local optima in the later stages of iteration, resulting in low parameter accuracy. This work proposes a fuzzy particle swarm optimization approach based on the transformation function and the filled function. This approach addresses the topic of particle swarm optimization in parameter identification from two perspectives. Firstly, the algorithm uses a transformation function to change the form of the fitness function without changing the position of the extreme point of the fitness function, making the extreme point of the fitness function more prominent and improving the algorithm's search ability while reducing the algorithm's computational burden. Secondly, on the basis of the multi-loop fuzzy control system based on multiple membership functions, it is merged with the filled function to improve the algorithm's capacity to skip out of the local optimal solution. This approach can be used to identify the parameters of permanent magnet synchronous motors by sampling only the stator current, voltage, and speed data. The simulation results show that the method can effectively identify the electrical parameters of a permanent magnet synchronous motor, and it has superior global convergence performance and robustness.

### KEYWORDS

Transformation function; filled function; fuzzy particle swarm optimization algorithm; permanent magnet synchronous motor; parameter identification

### Nomenclature

PMSM	Permanent magnet synchronous motor
PSO	Particle swarm optimization
CPSO	Convergent particle swarm optimization
FPSO	Fuzzy particle swarm optimization
CFPSO	Convergent fuzzy particle swarm optimization
FPSOTF	Fuzzy particle swarm optimization based on transformation function and filled function
IPMSM	Interior permanent magnet synchronous motor



SPMSM	Surface permanent magnet synchronous motor
DSCPSO	Chaotic particle swarm optimization with dynamic self-optimization
VSI	Voltage source inverter
DLIOBL	Dynamic lens imaging opposition-based learning
SOSAPSO	Self-optimizing simulated annealing particle swarm optimization
DOBL	Dynamic opposition-based learning
GOA	Greedy optimization algorithm
MRAS-SAPSO	Model reference adaptive system and simulated annealing particle swarm, optimization
QOBL	Quasi-opposition-based learning

## 1 Introduction

Because the stator resistance, d/q axis inductance, flux linkage, and other motor parameters are readily influenced by temperature, stator current, magnetic saturation, and other factors, the control performance of high-performance PMSM, reliability, and system dynamic and static performance are all negatively impacted. Therefore, precisely identifying the PMSM parameters is required to achieve high-performance PMSM control or reliable fault diagnosis [1–5].

Numerous scholars have undertaken comprehensive investigations into diverse techniques for ascertaining PMSM parameters. These techniques can be classified into two groups, namely offline identification and online identification, according to distinct identification procedures [6]. The two primary offline identification techniques for PMSM parameters are experimental measurement and finite element analysis. One approach that is frequently used to determine the PMSM parameters is finite element analysis [6].

The online identification approach is superior than the offline identification method because it allows for real-time modification of controller parameters for adaptive control in addition to real-time monitoring of the motor's operational state. As a result, researchers have undertaken a great deal of research in this area, producing successful findings and a variety of online identification techniques, including the Kalman filter algorithm [7,8], the recursive least square method [9–11], model reference adaptive algorithm [12,13], and artificial intelligence algorithm. These methods enable single-parameter or multi-parameter online identification of PMSM.

Although the recursive least squares algorithm is more susceptible to measurement noise, it has a reasonable computational complexity. The model reference adaptive algorithm can guarantee the asymptotic convergence of the detected parameters based on stability analysis. However, it is challenging to find an adaptive law that satisfies stability requirements when identifying numerous factors at once; this process involves a great deal of trial and error, and measurement noise can readily impact the identification results. In noisy situations, Extended Kalman filters (EKF) can offer state estimation in the sense of lowest variance. The parameters that need to be recognized must, nevertheless, be treated as state variables, which in some systems can be complicated.

In [14–16], PSO was adopted to achieve full-parameter identification of electrical parameters. In reference [17], an immune clone differential algorithm was proposed to identify four electrical parameters of PMSM. The identification accuracy and dynamic convergence speed of the algorithm are taken into account, and the algorithm has better global searching ability. An adaline neural network algorithm was proposed in reference [18] to identify multiple parameters simultaneously. However, the algorithm's calculation process was relatively complex, and it failed to account for the

impact of inductance changes brought about by injection current-induced changes in the motor's magnetic circuit on identification accuracy.

The method of parameter identification for PMSM outlined in reference [19] is based on chaotic particle swarm optimization with dynamic self-optimization (DSCPSO), which estimates the parameters and nonlinearity of the voltage source inverter (VSI) simultaneously in order to achieve real-time compensation of the VSI nonlinearity. The algorithm's capacity to investigate perhaps better regions is improved by incorporating tent chaos theory into the updating of the population, inertia weights, and learning variables of PSO. Additionally, the resilience of the simulated annealing algorithm (SA) is paired with the enhancement of population diversity, and the memory tempering annealing (MTA) strategy is employed to assure multiple learning of particles. Dynamic lens imaging opposition-based learning (DLIOBL) and domain optimization algorithms based on evolutionary information are meant to ensure a healthy balance between development and exploration. Reference [20] proposed a new multi-strategy self-optimizing simulated annealing particle swarm optimization technique called SOSAPSO. To prevent particle swarm monotonicity, this technique introduces dynamic opposition-based learning (DOBL) during the inertia weight update phase, simplifying the velocity term of PSO. Furthermore, to achieve self-learning in deep regions, a hybrid variation technique based on density and similarity between Cauchy and Gaussian processes is devised. Simulated annealing (SA) with memory and tempering mechanism is incorporated into SOSAPSO at the same time, and when the evolution of SOSAPSO reaches a standstill, the greedy optimization algorithm (GOA) is employed to improve the local fine exploitation capability. This approach has strong resilience and convergence speed, and it can successfully avoid the local convergence problem. However, due to the excessive improvement of PSO, the algorithm is too complicated, which greatly increases the computational difficulty. A directdrive permanent magnet synchronous generator (D-PMSG) parameter determination approach based on mean particle swarm optimization with extreme disturbance (EDMPSO)-EKF is proposed in reference [21]. Initially, a dual-thread identification model is constructed by analyzing the EKF principle. Subsequently, the particle swarm optimization (PSO) algorithm is refined to exit the local optimum by incorporating extremum interference and calculating the average extremum. Ultimately, the system executes parameter identification after getting the ideal noise matrix, and the enhanced PSO method is applied to the adaptive optimization of the EKF system noise matrix and measurement noise matrix. The calculation of this method is complicated and the calculation amount is large. A multi-parameter identification technique for permanent magnet synchronous motors based on simulated annealing particle swarm optimization (MRAS-SAPSO) and a model reference adaptive system was presented in reference [22]. First, the electrical parameters of a permanent magnet synchronous motor are determined using the model reference adaptive system approach. Second, the identification motor control system's electrical and mechanical parameters are further adjusted using the identification results as the initial population of PSO identification. The optimization results of the adaptive simulated annealing approach for multi-parameter identification are introduced in order to prevent premature convergence of PSO. This method simply uses the two algorithms successively, which not only does not avoid the shortcomings of the two algorithms, but also increases the complexity of the algorithm. A couple of enhanced PSOs have been suggested in the reference [23]. One is the combination of quasi-opposition-based learning (QOBL) with PSO, which helps to broaden and quicken the search actions. Second, the PSO's capacity for global search is further improved by the addition of a chaotic local search. The PSO is overly complex as a result of the algorithm's two improvements. PSO's research focus in PMSM parameter identification is on how to fully utilize its optimization potential without burdening the algorithm with more calculations. Since intelligent algorithms are very good at nonlinear optimization, they are employed extensively. However, such

algorithms have the problems of huge computational load and excessive data processing burden. Further research is needed to reduce the computational load of intelligent algorithms while ensuring identification accuracy.

Three features of the technique presented in this research help to better detect the parameters of PMSM. Initially, the PSO algorithm is combined to create a new fuzzy controller, whose fuzzy rules help to enhance the algorithm's local and global search capabilities as well as the PSO's premature convergence defect. This allows each particle to have a distinct rule distribution. Secondly, a transformation function is suggested to streamline the fitness function while maintaining the control object's global optimal solution. This minimizes the likelihood of a local minimum during the optimization procedure. Thirdly, early in the optimization process, the algorithm may be able to break out of the local minimum point thanks to the freshly filled function.

## 2 Mathematical Model of PMSM

The PMSM possesses an ideal symmetrical structure. Neglecting the effects of magnetic field saturation, iron loss, and eddy current loss in PMSM, the axis equation of the rotating coordinate system d/q can be described as follows:

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

in the equation,  $u_d, u_q, i_q, i_d$  are respectively the d/q axis voltage and current,  $R, \omega_e, \psi_f$  are respectively the stator resistance, electrical angular speed, and flux linkage.

In a VSI-fed PMSM system, The voltage equation of PMSM considering VSI nonlinearity is provided in reference [14].

$$\begin{cases} u_d^* = Ri_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q - D_d(k) V_{dead} \\ u_q^* = Ri_q + L_q \frac{di_q}{dt} + \omega_e (L_d i_d + \psi_f) - D_q(k) V_{dead} \end{cases} \quad (2)$$

where  $D_d$  and  $D_q$  are the functions of the rotor position period:

$$\begin{bmatrix} D_d(k) \\ D_q(k) \end{bmatrix} = 2 \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{\pi}{3}\right) \end{bmatrix} \times \begin{bmatrix} \text{sign}(i_{as}) \\ \text{sign}(i_{bs}) \\ \text{sign}(i_{cs}) \end{bmatrix} \quad (3)$$

where  $i_{as}, i_{bs}, i_{cs}$  are the stator abc three-phase currents, respectively.

$$\text{sign} = \begin{cases} 1, & i \geq 0 \\ -1, & i < 0 \end{cases} \quad (4)$$

$V_{dead}$  can be represented as:

$$V_{dead} = \frac{T_{dead} + T_{on} - T_{off}}{T_s} (V_{dc} - V_{sat} + V_d) + \frac{V_{sat} + V_d}{2} \quad (5)$$

where  $T_{dead}$  is the dead-time period,  $T_{on}, T_{off}$  are turn-on, turn-off times of the switching device.

$V_{dc}$ ,  $V_{sat}$  and  $V_d$  are respectively the DC bus voltage, the saturation voltage drop of the active switch, and the forward voltage drop of the free wheel diode. When the motor is in steady state operation, Eq. (2) can be expressed as:

$$\begin{cases} u_d^* = Ri_d - \omega_e L_q i_q - D_d(k) V_{dead} \\ u_q^* = Ri_q + \omega_e (L_d i_d + \psi_f) - D_q(k) V_{dead} \end{cases} \quad (6)$$

There are five electrical parameters that need to be identified:  $R$ ,  $L_d$ ,  $L_q$ ,  $\psi_f$ ,  $V_{dead}$ . However, the order of Eq. (6) is two, so the motor equation of state is a non-full rank equation. The variables such as stator resistance, rotor flux and winding inductance change with load and temperature during motor operation, which easily lead to inaccurate identification results.

According to the principle introduced in reference [2], PMSM generally adopts  $i_d = 0$  strategy for decoupling flux and torque control. Under steady state conditions, by injecting a current of  $i_d \neq 0$  into the d-axis for a short time, Eq. (6) can be transformed into a system of equations of full rank [14]:

$$\begin{cases} u_{d0}^*(k) = -\omega_e(k) L_q i_{q0}(k) - D_{d0}(k) V_{dead} \\ u_{q0}^*(k) = Ri_{q0}(k) + \omega_e(k) \psi_f - D_{q0}(k) V_{dead} \\ u_{d1}^*(k) = Ri_{d1}(k) - \omega_e(k) L_q i_{q1}(k) - D_{d1}(k) V_{dead} \\ u_{q1}^*(k) = Ri_{q1}(k) + \omega_e(k) (L_d i_{d1}(k) + \psi_f) - D_{q1}(k) V_{dead} \\ \Delta u_q^*(k) = u_{q1}^*(k) - u_{q0}^*(k) = \omega_e(k) L_d i_{d1}(k) - D_{q1}(k) V_{dead} + D_{q0}(k) V_{dead} \end{cases} \quad (7)$$

in the equation, the lower corner of variables and parameters collected in mode  $i_d = 0$  is marked with “0”, and the lower corner of variables and parameters collected in mode  $i_d \neq 0$  is marked with “1”.

The PMSM parameter identification structure is shown in Fig. 1.

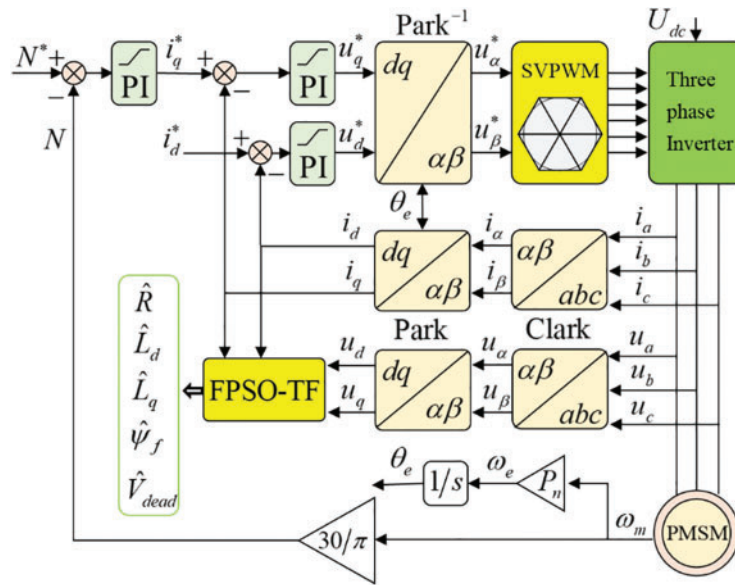
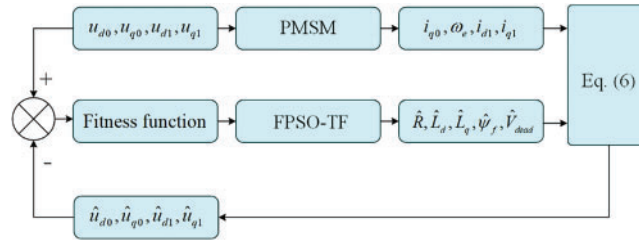


Figure 1: PMSM on-line parameter identification schematic diagram

### 3 Fitness Function

The identified value in the theoretical model must be continuously optimized and modified to bring it closer to the actual value until the fitness function value is the lowest. This process yields five parameters that need to be identified. The principle of parameter identification requires the output difference between the established theoretical model and the practical model that is currently in use, the calculation of the matching fitness function, and the calculation result as input to the intelligent algorithm. The schematic diagram for PMSM parameter identification based on FPSOTF is displayed in Fig. 2.



**Figure 2:** Principle of FPSOTF

The fitness function in this paper consists of five parts, which are defined as follows:

$$f_1(\hat{L}_q, \hat{V}_{dead}) = \frac{1}{n} \sum_{k=1}^n (u_{d0}^*(k) + D_{d0}(k) \hat{V}_{dead} - \hat{u}_{d0}(k))^2 \quad (8)$$

$$f_2(\hat{R}, \hat{\psi}_f, \hat{V}_{dead}) = \frac{1}{n} \sum_{k=1}^n (u_{q0}^*(k) + D_{q0}(k) \hat{V}_{dead} - \hat{u}_{q0}(k))^2 \quad (9)$$

$$f_3(\hat{R}, \hat{L}_q, \hat{V}_{dead}) = \frac{1}{n} \sum_{k=1}^n (u_{d1}^*(k) + D_{d1}(k) \hat{V}_{dead} - \hat{u}_{d1}(k))^2 \quad (10)$$

$$f_4(\hat{R}, \hat{\psi}_f, \hat{L}_q, \hat{V}_{dead}) = \frac{1}{n} \sum_{k=1}^n (u_{q1}^*(k) + D_{q1}(k) \hat{V}_{dead} - \hat{u}_{q1}(k))^2 \quad (11)$$

$$f_5(\hat{L}_d, \hat{V}_{dead}) = \frac{1}{n} \sum_{k=1}^n (\Delta u_q^*(k) - \hat{L}_d \omega_e(k) i_{d1}(k) + \hat{V}_{dead} (D_{q1}(k) - D_{q0}(k)))^2 \quad (12)$$

the fitness function is:

$$f = a_1 f_1 + a_2 f_2 + a_3 f_3 + a_4 f_4 + a_5 f_5 \quad (13)$$

where  $a_1, a_2, a_3, a_4, a_5$  are the weighting coefficients.

## 4 Algorithm Design

### 4.1 Multi Loop Fuzzy Control System

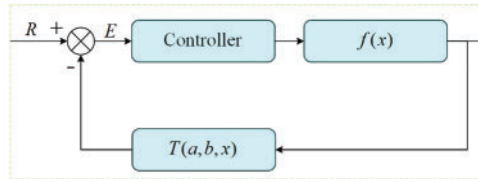
The optimization problem considered in this paper is:

$$\min f(x), x \in \Omega \subset R^n \quad (14)$$

where  $f(x)$  is a differentiable continuous function.

### 4.1.1 Single Loop Control System

The single-loop control system of problem (13) is shown in Fig. 3, where  $f(x)$  is the controlled object and  $R$  is the constant input. Here  $T(a, b, x)$  is used to change the expression form of fitness function without changing the position of the extreme point of fitness function, so that the optimal value of fitness function can be separated from the local minimum, reducing the possibility of the calculation process falling into the local optimal, and making the optimization problem fast and simple. The introduction of the transformation function must ensure that the position of the extreme point before and after the fitness function transformation remains unchanged.



**Figure 3:** The single loop control system

The transformation function is:

$$T(a, b, x) = a(x - b) / \sqrt{1 + (a(x - b))^2}, a > 0 \quad (15)$$

Eq. (15) is a differentiable continuous function. The extreme point of the fitness function is the solution of  $\partial f(x) / \partial x = 0$ , the transformed function is:

$$T(f(x)) = a(f(x) - b) / \sqrt{1 + (a(f(x) - b))^2}, a > 0 \quad (16)$$

The extreme point of Eq. (16) is the solution of Eq. (17).

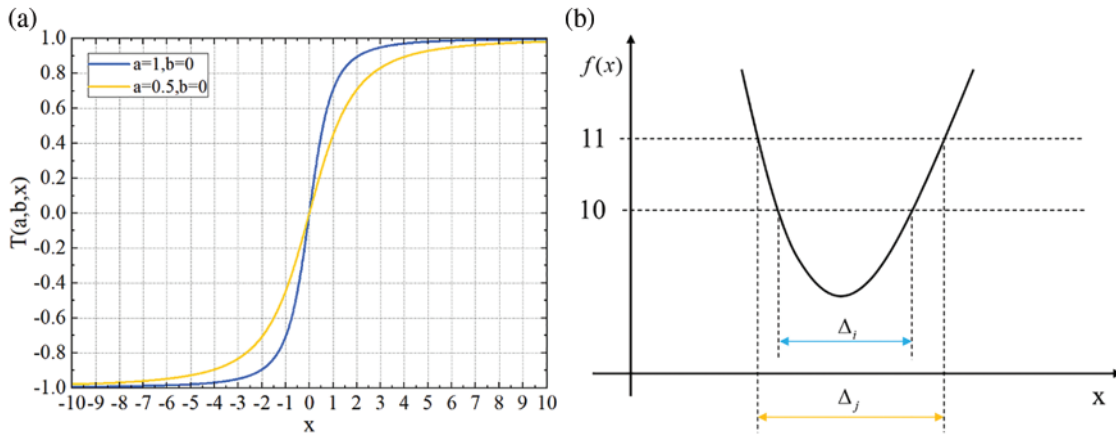
$$\partial T(f(x)) / \partial x = (\partial T / \partial f) \times (\partial f(x) / \partial x) = 0 \quad (17)$$

Eq. (17) can be further expressed as:

$$\partial T / \partial f = \partial \left[ a(f - b) / \sqrt{1 + (a(f - b))^2} \right] / \partial f = a / \left[ [1 + (a(f - b))^2] \sqrt{1 + (a(f - b))^2} \right] > 0 \quad (18)$$

It can be seen from Eq. (18),  $\partial T / \partial f > 0$ ,  $\partial T[f(x)] / \partial x = (\partial T / \partial x) \times (\partial f(x) / \partial x) = 0$ ,  $\partial f(x) / \partial x = 0$ , that is,  $T[f(x)]$  and  $f(x)$  have the same extreme point, and the transformation function does not change the extreme point position of the original fitness function.

As shown in Fig. 4a, when  $b = 0$ , the transformation function  $T(a, b, x)$  is basically close to the proportional function near the zero point, and the value of the regional transformation function far from the zero point tends to 1. When the transformation function acts on the fitness function, the shape of the fitness function curve before and after the action changes little near zero, while the fitness function curve with the fitness function value far from zero tends to 1, and its shape changes greatly and is horizontal. The new fitness function obtained by the transformation has a global minimum point near zero, and the transformed function can reduce the possibility of falling into the local minimum point in the algorithm calculation process. The larger  $a$  is, the smaller the effective search range is. Conversely, the smaller  $a$  is, the larger the effective search range is.



**Figure 4:** (a) Transformation function curve; (b) The relationship between  $f(x)$  and  $l_i$

As shown in Fig. 4a, when  $a = 1$ , if the independent variable of the transformation function is greater than 10, the value of the transformation function tends to 1, so the effective change interval of the independent variable is  $[-10,10]$ . As shown in Fig. 4b, when  $b = 0$ ,  $T[f(x)] = f(x)/\sqrt{1+f(x)^2}$ ,  $l_i$  of function  $f(x) = 10$  has two points of intersection with function  $f(x)$ , in this case, the effective search area should be  $\Delta_i$ , when  $b = 1$ ,  $T[f(x)] = f(x)/\sqrt{1+(f(x)-1)^2}$ ,  $f(x) - 1 = 10$ ,  $l_i$  of function  $f(x) = 11$  has two points of intersection with function  $f(x)$ , in this case, the effective search area should be  $\Delta_j$ , as shown in Fig. 4b,  $\Delta_i < \Delta_j$ , when  $a$  is held constant, the greater the value of  $b$ , the larger the effective search area  $\Delta_j$  becomes; otherwise, when  $b$  decreases, so does the effective search area  $\Delta_i$ .

When  $a = 1, b = 0$ ,  $T(a, b, x) = T(1, 0, x) = x/\sqrt{1+x^2}$ .

The comparison diagram after T transformation is shown in Fig. 5, in which Figs. 5a–5c are three functions (19)–(21). These three functions have multiple local extremum points, and the algorithm is easy to fall into local extremum points in the calculation process. Figs. 6a–6c are the corresponding  $T(f(x))$  images after T transformation. It can be seen from the function image after transformation that many local extreme values in the original function are reduced, while the optimal value after transformation is better highlighted.

$$f_1 = \frac{\pi}{2} (10 \sin^2(\pi(1.25 + 0.25x_1)) + (0.25 + 0.25x_2)^2 + (0.25 + 0.25x_1)(1 + 10 \sin^2(\pi(1.25 + 0.25x_2)))) \quad (19)$$

$$f_2 = x_1^2 + x_2^2 - 10 \cos(2\pi x_1) - 10 \cos(2\pi x_2) + 20 \quad (20)$$

$$f_3 = \frac{1}{4000} (x_1^2 + x_2^2) - \cos(x_1) \cos\left(\frac{x_2}{\sqrt{2}}\right) + 1 \quad (21)$$

#### 4.1.2 Multi Loop Fuzzy Control System

The multi loop fuzzy control system is shown in Fig. 7. A subsystem is equivalent to a particle, FNN1, ..., FNNm is the fuzzy controller distinguished from m different membership functions, and at the same time reflects the global search ability and local search ability of the system,  $f(x)$  is the fitness function.  $Best(f(x))$  is the optimal value of the subsystem, each subsystem shares the optimal



value of all subsystems, and quickly adjusts the optimization direction of the subsystem according to the shared information.  $\min(f(x))$  is the optimal value of the entire system.

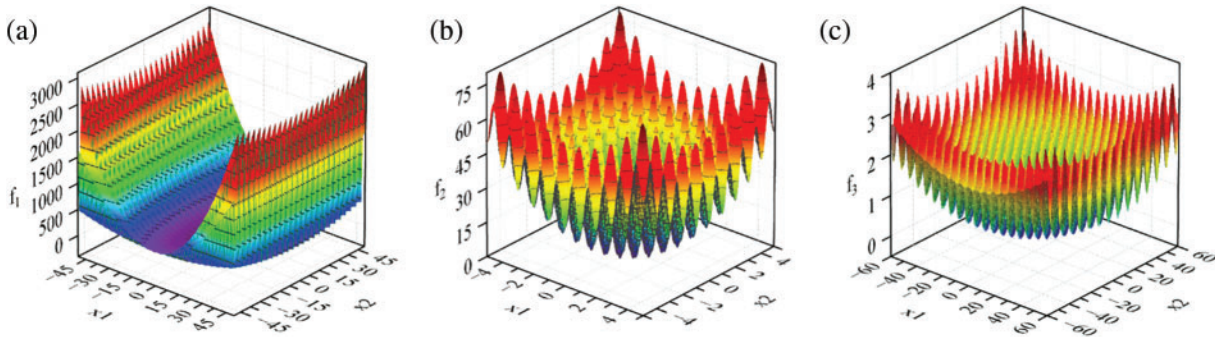


Figure 5: (a) Function of  $f_1$ ; (b) Function of  $f_2$ ; (c) Function of  $f_3$

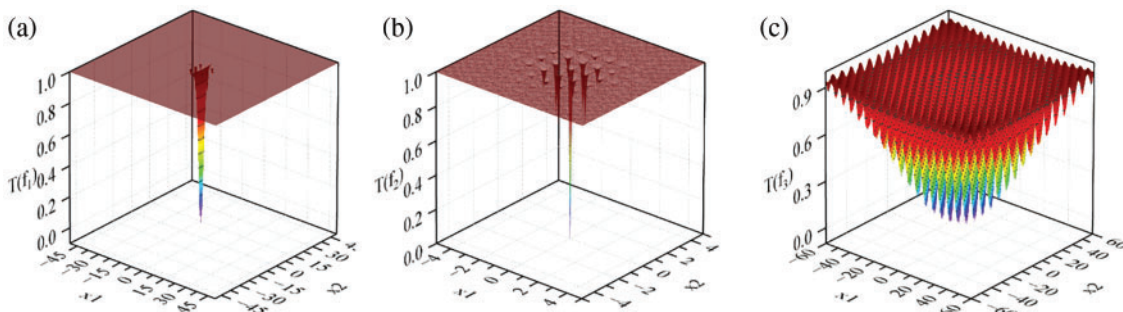


Figure 6: (a) Function of  $T(f_1)$ ; (b) Function of  $T(f_2)$ ; (c) Function of  $T(f_3)$

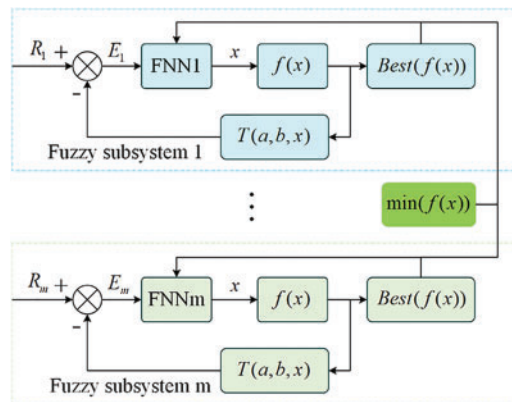


Figure 7: Multi-loop fuzzy control system block diagram

## 4.2 Fuzzy Particle Swarm Optimization

### 4.2.1 PSO Algorithm

In the standard PSO algorithm, it is assumed that in an n-dimensional problem space, there are m particles, each particle is a feasible solution to the problem to be optimized in the search space, and the

optimal solution of the problem is found through cooperation and competition between particles. The process of optimization problem can be seen as a process of continuous particle update. Each particle adjusts its flight speed and direction according to Eq. (22) with the current speed, individual optimal solution and global optimal solution. The evolution equation is as follows:

$$\begin{cases} v_i(t+1) = \omega v_i(t) + c_1 r_1 (p_i - x_i) + c_2 r_2 (p_g - x_i) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \quad (22)$$

where,  $x_i$  and  $v_i$  represent the current position and velocity of particle  $i$ , respectively,  $p_i$  and  $p_g$  represent the best value searched by particle  $i$  and the best position searched by all particles in the population, respectively.  $r_1$  and  $r_2$  are random numbers in  $[0,1]$ ,  $c_1$  and  $c_2$  are learning factors that are constants, The  $\omega$  is called the inertia weight and is generally taken as a number between 0.4 and 0.9 [24,25].

#### 4.2.2 Fuzzy Particle Swarm Optimization Algorithm

The conventional particle swarm optimization (PSO) approach is not able to find the best solution and is prone to premature convergence. Eq. (22) can be used to get the global optimal solution of the fitness function if there are a sufficient number of particles and their initial position distributions are completely scattered. However, this process takes a long time to compute. Reducing the number of particles will solve this issue by ensuring that the finite particles' starting locations are as far apart as possible. This work enhances particle swarm optimization, the algorithm's local and global search capabilities, and its usage of fuzzy rules in fuzzy controllers. According to the Takagi-Sugeno fuzzy model, the fuzzy rules of the fuzzy control system in this paper are adopted as follows:

$R_i^l$ : If  $\xi_1$  is  $M_{i1}^l \wedge \dots \wedge \xi_i$  is  $M_{ip}^l$  then

$$\begin{cases} x_1 = \omega_0(i, l) + \omega_1(i, l) x_i(t) \\ x_2 = \omega_2(i, l) (x_i(t) - x_i(t-1)) \\ x_3 = \omega_3(i, l) r_1 (x_{ibest}(t) - x_i(t-1)) \\ x_4 = \omega_3(i, l) r_1 (x_{allbest}(t) - x_i(t-1)) \\ \dot{x}_i(t) = x_1 + x_2 + x_3 + x_4 \end{cases} \quad (23)$$

where,  $R_i^l$  represents the  $l$ -th fuzzy rule of the  $i$ -th fuzzy subsystem,  $N_i$  represents the number of fuzzy rules of the  $i$ -th fuzzy subsystem,  $p$  represents the number of fuzzy variables of the  $i$ -th fuzzy subsystem,  $x_{ibest}$  is the best position vector for the  $i$ -th fuzzy subsystem,  $x_{allbest}$  is the best position vector for all fuzzy subsystems,  $\omega_0$ ,  $\omega_1$ ,  $\omega_2$ ,  $\omega_3$  and  $\omega_4$  are the speed regulation parameters of the  $i$ -th fuzzy subsystem. Using the weighted average method to defuzzify, the global model of the  $i$ -th fuzzy subsystem can be described as:

$$\begin{cases} x_1 = \omega_0(i, l) + \omega_1(i, l) x_i(t) \\ x_2 = \omega_2(i, l) (x_i(t) - x_i(t-1)) \\ x_3 = \omega_3(i, l) r_1 (x_{ibest}(t) - x_i(t-1)) \\ x_4 = \omega_4(i, l) r_1 (x_{allbest}(t) - x_i(t-1)) \\ \dot{x}_i(t) = \sum_l^{N_i} (x_1 + x_2 + x_3 + x_4) h_{il}(\xi) \end{cases} \quad (24)$$

where,  $h_{il}(\xi) = \frac{\prod_{k=1}^p \mu_{M_{ip}^l}}{\sum_{l=1}^{N_i} \prod_{k=1}^p \mu_{M_{ip}^l}}$ ,  $\sum_{l=1}^{N_i} h_{il}(\xi) = 1$ ,  $\mu_{M_{ip}^l}$  is the membership function of the  $i$ -th fuzzy variable,  $\mu_{M_{ip}^l} = R_i - T(a_{il}, b_{il}, f)$ ,  $R_i$  is the input to subsystem  $i$ .

The design idea of membership function is expounded in combination with Fig. 8: Taking the commonly used triangular membership function as an example, the membership function width interval in Fig. 8a is denoted as  $i_k$ , the width interval of the dependency function in Fig. 8b is denoted as  $i_m$ , if  $i_k < i_m$ , the search range  $\Delta_k$  for the same number of membership functions is less than  $\Delta_m$ . The smaller the membership function width interval  $i_k$ , the stronger the local search ability and the higher the precision of the fuzzy control system. On the contrary, the larger the membership function width interval  $i_m$ , the stronger the global search capability and the larger the search scope of the system. According to this idea, this paper sets  $n$  membership functions for  $n$  fuzzy control subsystems, as shown in Fig. 9. The membership function width interval of the  $k$ th fuzzy control subsystem is  $i_k$ , and the membership function width interval of the  $m$ th fuzzy control subsystem is  $i_m$ ,  $i_k < i_m$  ( $m, k = 1, 2, \dots, n$ ). In this way, the membership function width interval of each fuzzy control subsystem is different, and the control system with narrow membership function width interval has strong local search ability and high precision. The control system with wide membership function width has strong global search ability and large search range, which makes up for the defects of traditional particle swarm optimization.

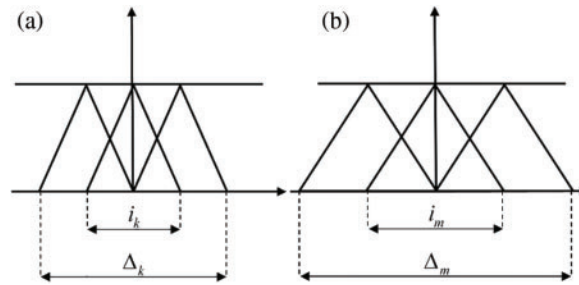


Figure 8: (a) Membership function of  $\mu_{M_{ikp}^l}$  (b) Membership function of  $\mu_{M_{imp}^l}$

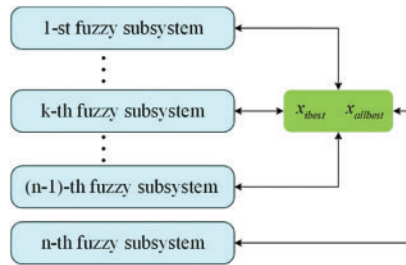
It can be seen that for  $n$  fuzzy control subsystems, the width interval of the membership function can be changed only by setting the parameter  $a$  of the transformation function, that is,  $a_{ikl} < a_{iml}$ , then  $i_k < i_m$ . At this time, the setting of  $n$  fuzzy control subsystems is to set  $a$  as a  $n$  row matrix. The larger the value of  $a_k$ , the narrower the width interval of the membership function of the  $k$ -th subsystem, the stronger the local search ability and the higher the precision of the fuzzy control subsystem. On the contrary, the smaller the value of  $a_m$ , the wider the membership function of the  $m$ -th subsystem, the stronger the global search ability of the fuzzy subsystem, and the larger the search range. When a very small positive number is taken from the walk length, the discretization of Eq. (24) is described as follows:

$$\begin{cases} x_0 = x_i(k) \\ x_1 = \omega_0(i, l) + \omega_1(i, l) x_i(k) \\ x_2 = \omega_2(i, l) (x_i(k) - x_i(k-1)) \\ x_3 = \omega_3(i, l) r_1 (x_{ibest}(k) - x_i(k)) \\ x_4 = \omega_4(i, l) r_1 (x_{allbest}(k) - x_i(k)) \\ x_i(k+1) = x_0 + \eta_4 \sum_{l=1}^{N_i} (x_1 + x_2 + x_3 + x_4) h_{il}(\xi) \end{cases} \quad (25)$$

here  $\eta_4$  is the length from the walk, a very small positive number.  $\omega_0, \omega_1, \omega_2, \omega_3, \omega_4$  corrected as follows:

$$\begin{cases} \omega_0(i, l) = -\eta_1 (\partial f(x_i) / \partial x_i(t) + \eta_2 (x_i(t) - x_{allbest}(t))) h_{il} \\ \omega_1(i, l) = -\eta_1 (\partial f(x_i) / \partial x_i(t) + \eta_2 (x_i(t) - x_{allbest}(t))) x_i(t) h_{il} \\ \omega_2(i, l) = -\eta_1 (\partial f(x_i) / \partial x_i(t) + \eta_2 (x_i(t) - x_{allbest}(t))) (x_i(t) - x_i(t-1)) h_{il} \\ \omega_3(i, l) = -\eta_1 (\partial f(x_i) / \partial x_i(t) + \eta_2 (x_i(t) - x_{allbest}(t))) (x_{allbest}(t) - x_i(t)) h_{il} \\ \omega_4(i, l) = -\eta_1 (\partial f(x_i) / \partial x_i(t) + \eta_2 (x_i(t) - x_{allbest}(t))) (x_{ibest}(t) - x_i(t)) h_{il} \end{cases} \quad (26)$$

where,  $i = 1, 2, \dots, n, l = 1, 2, \dots, N_i, \eta_1, \eta_2$  are small positive numbers,  $n$  is the number of subsystems.



**Figure 9:** The relationship with all subsystems

#### 4.2.3 Filled Function

Fuzzy particle swarm optimization (FPSO) improves the global search ability of particle swarm optimization, but it also has the possibility of falling into local extreme points. Therefore, a filled function is constructed to make the algorithm have the ability to jump out of the local extreme point to a smaller point, so as to search for the global optimal value.

The main logic of the filled function method is that if a local minimum  $x_1$  has been found, but it is not a global minimum, construct a filled function at  $x_1$  so that the iteration point leaves the valley where  $x_1$  is located, and find a better solution for  $x_2$  (i.e., the fitness function value at  $x_2$  is smaller than the fitness function value at  $x_1$ ). As shown in Fig. 10, construct a function  $P(x)$ , called the filled function, from a known minimum point  $x_1$  of the fitness function  $f(x)$ , which takes a maximum value at  $x_1$ , there is no minimum point or saddle point in any basin higher than  $B_1$  of  $x_1$ , but there is a minimum point or saddle point  $x^*$  in some basin lower than  $x_1$ , then minimize  $f(x)$  with  $x^*$  as the initial point and finds a new minimum  $x_2$ , making  $F(x_2) \leq F(x_1)$ , replacing  $x_2$  with  $x_1$ , repeat the process until the global minimum of  $f(x)$  is found [26,27]. Common filled functions are as follows:

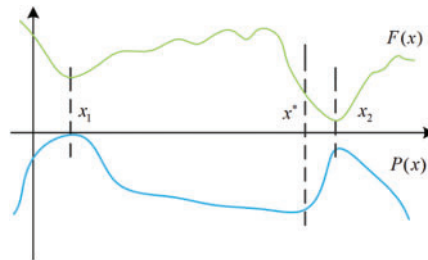
$$G(x, x_1, r, \rho) = -\rho^2 \log(x + f(x)) - \|x - x_1\|^2 \quad (27)$$

$$Q(x, x_1, a) = -(f(x) - f(x_1)) \exp(a \|x - x_1\|^2) \quad (28)$$

where,  $r, \rho, a$  are the parameters that need to be adjusted during the calculation, Eq. (27) is a two-parameter filled function, which is troublesome and time-consuming to adjust, while Eq. (28) is a single-parameter filled function, however, only when  $a$  is large, it can become a filled function at  $x_1$ , because the parameter  $a$  is on the exponential, when  $a$  is large, the calculation performance is not ideal. This paper constructs a new filled function:

$$\begin{cases} P(x, x_1) = -\text{sgn}(f(x) - f(x_1)) (\|x - x_1\| / (1 + \|x - x_1\|)) \\ \text{sgn}(t) = \begin{cases} 1, & t \geq 0 \\ -1, & t < 0 \end{cases} \end{cases} \quad (29)$$

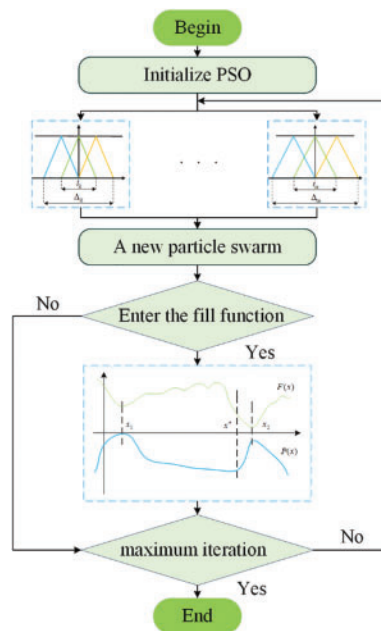
the filled function is parameterless, which can save the long calculation steps and the time of adjusting parameters, so as to improve the efficiency of the algorithm.



**Figure 10:** Filled function diagram

### 4.3 FPSOTF Algorithm Flow

The flow of fuzzy particle swarm optimization algorithm based on filled function and transformation function (FPSOTF) is shown in Fig. 11.



**Figure 11:** Flowchart of FPSO-TF algorithm

**Algorithm:** FPSOTF algorithm for all parameters identification

**Step 1:** Given variable  $Xn \times m$ , system input  $R$ , fuzzy rule number  $N\sigma$ , parameter  $a$  of transformation function, initial value of learning coefficient  $\eta_1, \eta_2, \eta_3, \eta_4$ , maximum number of algorithm iterations  $MN$ , maximum number of entering filled function  $TT$ , the maximum degree  $FMN$  of the variable iteration in the filled function, and the filled function iteration parameter  $d_1$  ( $0 < d_1 < 1E - 5$ ),  $d_2$  ( $0 < d_2 < 1$ ),  $d_3$  ( $1 < d_3$ ),  $d_4$  ( $0 < d_4 < 1$ ),  $d_5$  ( $1 < d_5$ ), let  $L = 0, j = 0, k = 1$ .

**Step 2:** Make  $h(x) = f(x)$ , by Eq. (6) update  $Xn \times m, j = j + 1$ , and calculate the  $f(x) = [f_1, f_2, \dots, f_m]$  ( $f_\sigma(x) = T_\sigma(f(x)), \sigma = 1, 2, \dots, m$ ).

if  $f_\sigma < f_{\sigma best}$  ( $\sigma = 1, 2, \dots, m$ )

then  $x_{\sigma best} = x_\sigma, f_{\sigma best} = f_\sigma$

if  $f_{\sigma best} < f_{allbest}$  ( $\sigma = 1, 2, \dots, m$ )

then  $x_{allbest} = x_{\sigma best}, f_{allbest} = f_{\sigma best}$ .

**Step 3:** If  $|h_\sigma(x) - f_\sigma(x)| < d_1, k \leq FMN$  and  $f_\sigma(x) \geq f_{allbest}$

for  $k = k + 1$

$x_\sigma(i, k) = x_\sigma(i, k) + d_2(\text{rand}(1) - 0.5) \exp(d_3(k - 1) / FMN), \sigma = 1, 2, \dots, m, i = 1, 2, \dots, n,$

$x_\sigma(i, k) = x_\sigma(i, k) + d_4(\text{rand}(1) + 1) \exp(d_5(k - 1) / FMN) (x_\sigma(i, k) - x_{allbest}),$  and calculate  $f(x) = [f_1, f_2, \dots, f_m]$  ( $f_\sigma(x) = T_\sigma(f(x)), \sigma = 1, 2, \dots, m$ ),

if  $f_\sigma < f_{\sigma best}$  ( $\sigma = 1, 2, \dots, m$ )

then  $x_{\sigma best} = x_\sigma, f_{\sigma best} = f_\sigma$

if  $f_{\sigma best} < f_{allbest}$  ( $\sigma = 1, 2, \dots, m$ )

then  $x_{allbest} = x_{\sigma best}, f_{allbest} = f_{\sigma best}$

re-enter step 3 for circulation, or else,  $k \geq FMN$ , then  $L = L + 1$ , go to step 4.

**Step 4:** Update parameters  $\omega_0, \omega_1, \omega_2, \omega_3, \omega_4$  according to Eq. (7)

if  $j > MN$  or  $L > TT$

then output  $f_{allbest}(x_{allbest})$ , the algorithm ends

otherwise, go to Step 2.

## 5 Simulation Results and Analysis

To parameterize PMSM in this simulation, a MATLAB/Simulink platform simulation model based on the FPSOTF is constructed. The simulation model's parameters are configured as shown 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 <sup>-1</sup>
T	Nominal torque	4 N·m
U	Rated line voltage	220 V
R	Stator resistance	2.76 Ω
L <sub>d</sub>	d-axis inductance	6.4 mH
L <sub>q</sub>	q-axis inductance	6.4 mH

(Continued)

**Table 1 (continued)**

Symbol	Quantity	Value
$\psi_f$	Flux linkage	0.204 Wb
$p_n$	Number of pole pairs	4

In order to make the simulation results scientific and comparable, FPOTF was compared with four algorithms, FPSO, CPSO and CFPSO, and the simulation parameters of these five algorithms were set to the same value, and the initial values of all five parameters to be identified were set in the range of  $(-20,20)$ . The initial values of the five parameters to be identified differ greatly from the real values set in the simulation. In order to reduce the error caused by the randomness of the algorithm, each algorithm is independently run 100 times, and the final result is the average value of the 100 simulation results.

The simulation results are shown in [Table 2](#).

**Table 2:** The results of five parameters identified by five algorithms

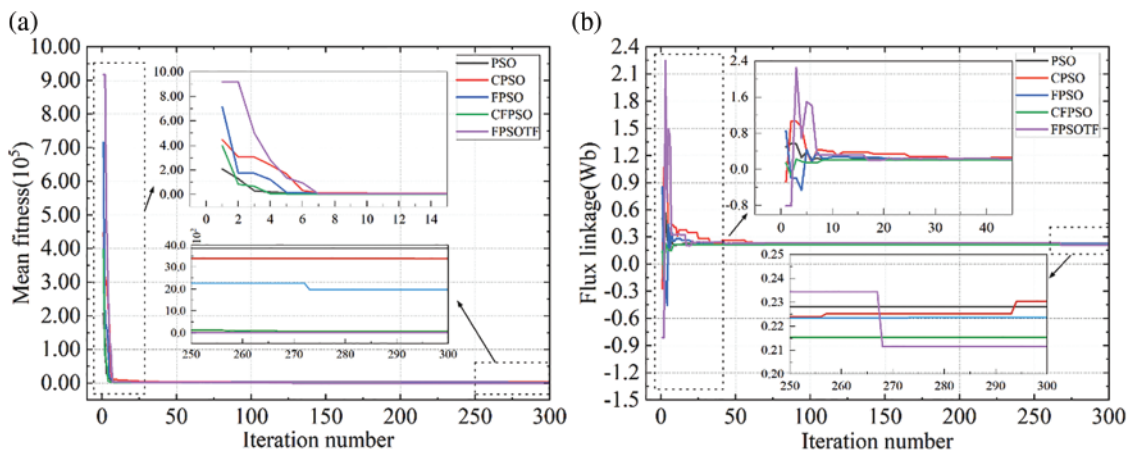
Identified parameter		PSO	CPSO	FPSO	CFPSO	FPSOTF
$R/\Omega$		5.642	4.852	4.144	3.005	2.802
$L_d/mH$		8.228	6.244	7.941	6.891	6.557
$L_q/mH$		14.887	14.356	11.588	8.365	6.512
$\psi_f/Wb$		0.2281	0.2304	0.2236	0.2153	0.2114
	Mean	3861	3397	1975	68.7	12.1
Fitness	Std.dev	14.660	10.635	9.532	7.001	2.237
	t-value	2582.39	3099.02	1610.62	76.62	0
$V_{dead}/V$		-0.9359	-1.1996	-0.9235	-0.7544	-0.6978
Time/s		55.5	58.6	60.2	64.5	59.4

The data in [Tables 2](#) and [1](#) show that, with the least amount of error and average fitness function value, the identification results of the five FPSOTF algorithm parameters are closest to the set values. [Table 2](#)'s t-value indicates that the FPSOTF in this study and the other four variation PSO algorithms all have t-values higher than 2.06, indicating that FPSOTF is significantly more effective than the other four variant PSO algorithms for PMSM multi-parameter identification. 98 percent of the time. The algorithm's running time in [Table 2](#) shows that, in comparison to the PSO algorithm, the FPSOTF algorithm runs longer, increasing by 7%, decreasing by 1.3%, and increasing by 1.4% when compared to the CPSO algorithm. In contrast, the CFPSO algorithm runs slower, decreasing by 8%. This demonstrates that the FPSOTF algorithm's complexity rises when compared to the PSO algorithm but falls when compared to the FPSO and CFPSO algorithms. This demonstrates how well the FPSOTF algorithm strikes a compromise between computational complexity and balance identification accuracy. The comparison's findings demonstrate that the FPSOTF algorithm performs better than the other methods.

Table 3 presents a comparison of the identification accuracy of PMSM parameters using the algorithm suggested in this paper with reference 7. The table shows that for stator resistance and d-axis inductance, the identification accuracy of the technique suggested in this study is higher than that of reference 7, but for d-axis inductance and flux linkage, it is marginally worse. This demonstrates that the method presented in this study has advanced to a satisfactory level; however, more research is needed to properly balance the identification accuracy across different parameters. The identification process of five parameters of PMSM by five algorithms is shown in Figs. 12–14.

**Table 3:** Comparison of identification accuracy of FPSOTF and reference 7

Identified parameter	Reference 7			FPSOTF		
	Set value	Identified value	Error (%)	Set value	Identified value	Error (%)
$R/\Omega$	0.065	0.068	4.16	2.76	2.802	1.52
$L_d/\text{mH}$	0.0373	0.0380	1.87	6.4	6.557	2.45
$L_q/\text{mH}$	0.0488	0.0500	2.45	6.4	6.512	1.75
$\psi_f/\text{Wb}$	0.020	0.0195	-2.5	0.204	0.2114	3.62

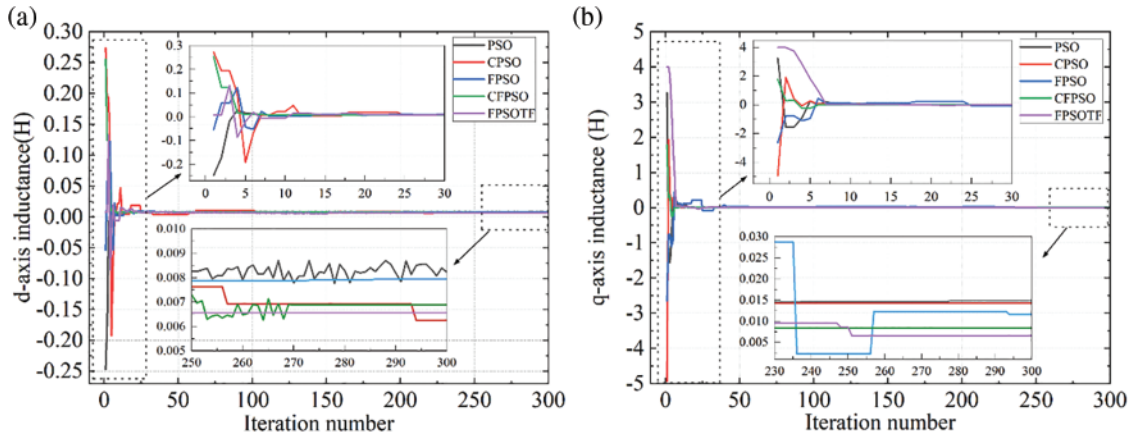


**Figure 12:** (a) Mean fitness value curve; (b) Flux linkage curve

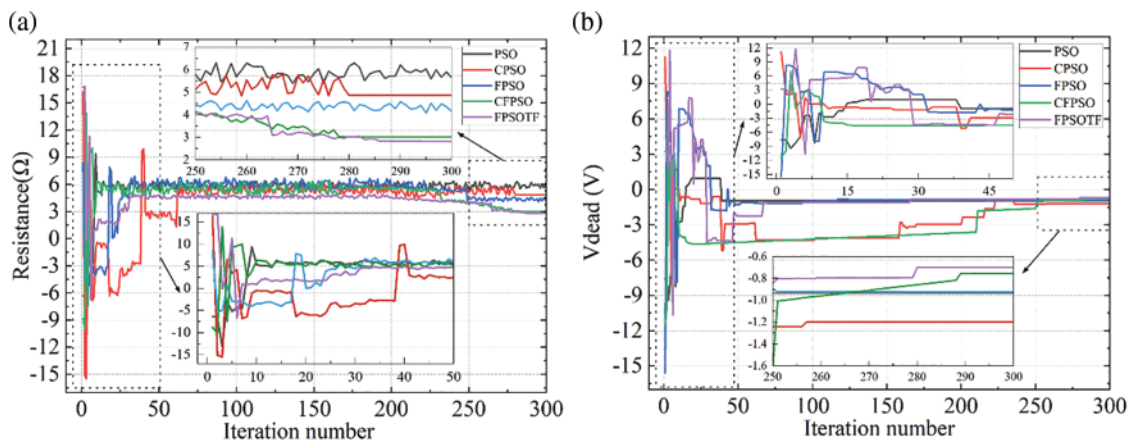
The mean fitness value curve in the process of calculating various algorithms is shown in Fig. 12a. The FPSOTF method suggested in this research enters the stable search phase more slowly than the other four algorithms, as seen in the figure, suggesting that the FPSOTF algorithm has superior global search capabilities. While the fitness values of CFPSO and FPSOTF are small—FPSOTF being the smallest—and thus suggest that these two algorithms have superior local searching ability, the fitness values of PSO, CPSO, and FPSO are always large, suggesting that these three algorithms are trapped in local optimal solutions. The flux linkage convergence curve for the five algorithms’ computation is shown in Fig. 12b. The figure shows that all five algorithms have essentially finished the global search in less than 50 iterations, and there is little variation in the convergence speed of the algorithms. Each algorithm has reached the stable search phase after 50 iterations, and the chart shows that all five algorithms have reached the local search phase after 270 iterations. The other three algorithms’ poor search accuracy suggests that they are stuck in the local optimal solution and are unable to escape,



whereas CFPSO and FPSOTF exhibit excellent search accuracy. According to the final outcome, FPFOTF has the best search accuracy.



**Figure 13:** (a) D-axis inductance curve; (b) Q-axis inductance curve



**Figure 14:** (a) Stator resistance curve; (b) VSI disturbance voltage curve

Fig. 13 is the convergence curve of the d/q axis inductance. It can be seen from the figure that all five algorithms have completed convergence within 30 iterations, that is, global search has been realized, and the five algorithms can complete global search in a relatively short time, indicating that inductance is an easy parameter to identify, and that the calculation speed of the five algorithms in the early search process is basically the same. However, in the refined search process after 270 iterations, only CFPSO and FPSOTF have better search accuracy, among which FPSOTF algorithm can better stabilize near the set value, which indicates that the FPSOTF algorithm has better search ability compared with other algorithms.

The convergence curve for stator resistance during the five algorithms' computation process is shown in Fig. 14a. The figure shows that the resistance has not converged well and fluctuates significantly over the entire computation procedure, suggesting that the resistance is a difficult-to-identify quantity. Different algorithms can do fine search more effectively after 280 iterations. Nevertheless, within a specific numerical range, the three algorithms—PSO, CPSO, and FPSO—still

fail to stabilize the identification findings. The algorithm's search capacity is demonstrated by the nearly ideal final search results of CFPSO and FPSOTF, as well as the identification results of FPSOTF that are closest to the set value.

The disrupted voltage curve of VSI is displayed in Fig. 14b. The figure illustrates how this parameter's convergence process takes a while. The fact that the CPSO and CFPSO algorithms needed 280 iterations to do fine searches suggests that they are not very good at conducting global searches to find this parameter. The FPSOTF algorithm's identification result is the one that comes the closest to the set value in the final result. The FPSOTF method outperforms the other four algorithms in terms of calculation performance and has good global and local search capabilities throughout the identification process. At present, in the recent research, two improved PSO algorithms are proposed in reference 29 and 30, respectively, SOSAPSO (a novel multi-strategy self-optimizing simulated annealing particle swarm optimization) and EDMPSO-EKF (mean particle swarm optimization with extreme disturbance combined with extended Kalman filter), The algorithm proposed in this paper is compared with these two algorithms, and the results are as follows.

Tables 4 and 5 show that, in comparison to the FPSOTF algorithm, the SOSAPSO algorithm's identification errors for flux linkage are smaller, but the algorithm's identification errors for stator resistance, d-axis inductance, and q-axis inductance are marginally higher. For all four parameters, the MDEPSO-EKF algorithm's identification errors are larger than the FPSOTF algorithm's. Overall, the four parameters' identification accuracy is higher for the FPSOTF method, suggesting that the algorithm presented in this study is more capable of identification. On the other hand, the comparison with the SOSAPSO method reveals that the FPSOTF algorithm has a significant inaccuracy in flux linkage identification, so the future research should balance the identification accuracy of the four parameters.

**Table 4:** Comparison of identification accuracy of FPSOTF and SOSAPSO

Identified parameter	SOSAPSO			FPSOTF		
	Set value	Identified value	Error (%)	Set value	Identified value	Error (%)
$R/\Omega$	2.76	2.821	2.21	2.76	2.802	1.52
$L_d/\text{mH}$	6.4	6.573	2.70	6.4	6.557	2.45
$L_q/\text{mH}$	6.4	6.551	2.34	6.4	6.512	1.75
$\psi_f/\text{Wb}$	0.204	0.20806	1.99	0.204	0.2114	3.62

**Table 5:** Comparison of identification accuracy of FPSOTF and EDMPSO-EKF

Identified parameter	EDMPSO-EKF			FPSOTF		
	Set value	Identified value	Error (%)	Set value	Identified value	Error (%)
$R/\Omega$	2.76	2.813	1.92	2.76	2.802	1.52
$L_d/\text{mH}$	6.4	6.561	2.52	6.4	6.557	2.45
$L_q/\text{mH}$	6.4	6.513	1.77	6.4	6.512	1.75
$\psi_f/\text{Wb}$	0.204	0.21201	3.92	0.204	0.2114	3.62

Due to the substantial nonlinearity of the PMSM model and the abundance of local extremum points in the fitness function, the technique exhibits significant fluctuation or non-convergence in parameter identification of PMSM. This makes finding the optimal solution difficult for algorithms with limited processing power. The FPSOTF algorithm outperforms the other algorithms in terms of identification accuracy, as demonstrated by the identification procedure and results. Additionally, the data in the table demonstrates that the FPSOTF method has the lowest average fitness value and an identification value that is closer to the design value, suggesting that the algorithm is better and has strong robustness and convergence.

## 6 Conclusion and Future Directions

The study develops a thorough mathematical model of PMSM that takes into consideration the voltage source-inverter's (VSI) nonlinearity and how it affects parameter identification. Furthermore, by combining FPSO with the transformation function and filled function, a brand-new algorithm known as FPSOTF is presented. The filled function improves PSO's local search capabilities, while the transformation function makes the process of finding fitness functions less difficult. Consequently, five parameters are correctly identified by this algorithm. This approach provides strong anti-interference ability and computing performance during parameter identification of PMSM, as demonstrated by a comparative comparison with four other algorithms. Because of their powerful nonlinear optimization capabilities, artificial intelligence algorithms have drawn a lot of attention. However, these algorithms also have significant computational and data processing overhead. Consequently, more study is required to decrease the computational burden of artificial intelligence algorithms and increase their identification accuracy.

**Acknowledgement:** We are grateful for the valuable comments provided by the anonymous reviewers, which have helped us to improve the content of this paper and make it even better.

**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.

**Author Contributions:** The authors confirm contribution to the paper as follows: Study conception and design: Shuai Zhou and Dazhi Wang; data collection: Yongliang Ni, Keling Song and Yanming Li; analysis and interpretation of results: Shuai Zhou; draft manuscript preparation: Shuai Zhou. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** Data not available due to [ethical/legal/commercial] restrictions. Due to the nature of this research, participants of this study did not agree for their data to be shared publicly, so supporting data is not available.

**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. 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 Trans. Ind. Applicat.*, vol. 58, no. 2, pp. 2029–2041, Apr. 2022. doi: [10.1109/TIA.2021.3136807](https://doi.org/10.1109/TIA.2021.3136807).

- [2] Z. Q. Zhu, D. Liang, and K. Liu, "Online parameter estimation for permanent magnet synchronous machines: An overview," *IEEE Access*, vol. 9, pp. 59059–59084, Oct. 2021. doi: [10.1109/ACCESS.2021.3072959](https://doi.org/10.1109/ACCESS.2021.3072959).
- [3] Y. L. Yu, X. Y. Huang, and Z. K. Li, "Overall electrical parameters identification for IPMSMs using current derivative to avoid rank deficiency," *IEEE Trans. Ind. Electron.*, vol. 70, no. 7, pp. 7515–7520, Jul. 2023. doi: [10.1109/TIE.2022.3203751](https://doi.org/10.1109/TIE.2022.3203751).
- [4] W. Feng, W. J. Zhang, and S. D. Huang, "A novel parameter estimation method for PMSM by using chaotic particle swarm optimization with dynamic self-optimization," *IEEE Trans. Veh. Technol.*, vol. 72, no. 7, pp. 8424–8432, Jul. 2023. doi: [10.1109/TVT.2023.3247729](https://doi.org/10.1109/TVT.2023.3247729).
- [5] Y. L. Yu *et al.*, "Full parameter estimation for permanent magnet synchronous motors," *IEEE Trans. Ind. Electron.*, vol. 69, no. 5, pp. 4376–4386, May 2022. doi: [10.1109/TIE.2021.3078391](https://doi.org/10.1109/TIE.2021.3078391).
- [6] S. A. Odhano, P. Pescetto, H. A. A. Awan, M. Hinkkanen, G. Pellegrino and R. Bojoi, "Parameter identification and self-commissioning in AC motor drives: A technology status review," *IEEE Trans. Power Electron.*, vol. 4, no. 34, pp. 3603–3614, Apr. 2019.
- [7] Z. W. Wang *et al.*, "UKF-based parameter estimation and identification for permanent magnet synchronous motor," *Front. Energy Res.*, vol. 10, no. 1, pp. 855649–855658, 2022.
- [8] S. Xiao and A. Griffio, "Online thermal parameter identification for permanent magnet synchronous machines," *IET Electr. Power Appl.*, vol. 14, no. 12, pp. 2340–2347, 2020.
- [9] M. Koc and O. E. Ozciflikci, "Precise torque control for interior mounted permanent magnet synchronous motors with recursive least squares algorithm based parameter estimations," *Eng. Sci. Technol. Int. J.*, vol. 34, no. 1, pp. 101087–101093, 2022.
- [10] H. Yu, J. Wang, and Z. Xin, "Model predictive control for PMSM based on discrete space vector modulation with RLS parameter identification," *Energies*, vol. 15, no. 11, pp. 4041–4057, 2022.
- [11] C. Y. Yang, L. C. Bu, and B. Chen, "Energy modeling and online parameter identification for permanent magnet synchronous motor driven belt conveyors," *Measurement*, vol. 178, no. 1, pp. 109342–109350, 2021.
- [12] X. Qi, C. Sheng, Y. Guo, T. Su, and H. Wang, "Parameter identification of a permanent magnet synchronous motor based on the model reference adaptive system with improved active disturbance rejection control adaptive law," *Appl. Sci.*, vol. 13, no. 21, pp. 12076–12090, Jan. 2023. doi: [10.3390/app132112076](https://doi.org/10.3390/app132112076).
- [13] J. She, L. Wu, C. K. Zhang, Z. T. Liu, and Y. H. Xiong, "Identification of moment of inertia for PMSM using improved modelreference adaptive system," *Int. J. Control Autom. Syst.*, vol. 20, no. 1, pp. 13–23, 2022. doi: [10.1007/s12555-020-0549-8](https://doi.org/10.1007/s12555-020-0549-8).
- [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 Trans. Power Electron.*, vol. 33, no. 12, pp. 10858–10871, Dec. 2018. doi: [10.1109/TPEL.2018.2801331](https://doi.org/10.1109/TPEL.2018.2801331).
- [15] R. A. El-Wanis and M. A. Hamida, "Parameter estimation of permanent magnet synchronous machines using particle swarm optimization algorithm," *RRST-EE*, vol. 67, no. 4, pp. 377–382, Dec. 2022.
- [16] J. He and Z. H. Liu, "Estimation of stator resistance and rotor flux linkage in SPMSM using CLPSO with opposition-based-learning strategy," *J. Control Sci. Eng.*, vol. 2016, no. 5781467, 2016. doi: [10.1155/2016/5781467](https://doi.org/10.1155/2016/5781467).
- [17] G. Lin, J. Zhang, and Z. Liu, "Parameter identification of PMSM using immune clonal selection differential evolution algorithm," *Math. Probl. Eng.*, vol. 2014, no. 6, pp. 1–10, 2014. doi: [10.1155/2014/160685](https://doi.org/10.1155/2014/160685).
- [18] K. Liu and Z. Q. Zhu, "Position-offset-based parameter estimation using the adaline NN for condition monitoring of permanent-magnet synchronous machines," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2372–2383, Apr. 2015. doi: [10.1109/TIE.2014.2360145](https://doi.org/10.1109/TIE.2014.2360145).
- [19] W. Feng, W. Zhang, and S. Huang, "A novel parameter estimation method for PMSM by using chaotic particle swarm optimization with dynamic self-optimization," *IEEE Trans. Veh. Technol.*, vol. 72, no. 7, pp. 8424–8432, Jul. 2023. doi: [10.1109/TVT.2023.3247729](https://doi.org/10.1109/TVT.2023.3247729).
- [20] D. D. Wen *et al.*, "A novel multi-strategy self-optimizing SAPSO algorithm for PMSM parameter identification," *IET Power Electron.*, vol. 16, no. 2, pp. 305–319, Aug. 2023. doi: [10.1049/pel2.12385](https://doi.org/10.1049/pel2.12385).

- [21] Q. Xiao, K. Liao, C. Shi, and Y. Zhang, "Parameter identification of direct-drive permanent magnet synchronous generator based on EDMPSO-EKF," *IET Renew. Power Gen.*, vol. 16, no. 5, pp. 1073–1086, Jan. 2022. doi: [10.1049/rpg2.12415](https://doi.org/10.1049/rpg2.12415).
- [22] G. Su, P. Wang, Y. Guo, G. Cheng, S. Wang and D. Zhao, "Multiparameter identification of permanent magnet synchronous motor based on model reference adaptive system—simulated annealing particle swarm optimization algorithm," *Electronics*, vol. 11, no. 1, pp. 159–176, Jan. 2022. doi: [10.3390/electronics11010159](https://doi.org/10.3390/electronics11010159).
- [23] M. A. Ahandani, J. Abbasfam, and H. Kharrati, "Parameter identification of permanent magnet synchronous motors using quasi-opposition-based particle swarm optimization and hybrid chaotic particle swarm optimization algorithms," *Appl. Intell.*, vol. 52, no. 11, pp. 13082–13096, Jan. 2022. doi: [10.1007/s10489-022-03223-x](https://doi.org/10.1007/s10489-022-03223-x).
- [24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN'95-Int. Conf. Neural Netw.*, Perth, WA, Australia, 1995, pp. 1942–1948.
- [25] Z. H. Zhan, J. Zhang, Y. Li, and Y. H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Computat.*, vol. 15, no. 6, pp. 832–847, Dec. 2011. doi: [10.1109/TEVC.2010.2052054](https://doi.org/10.1109/TEVC.2010.2052054).
- [26] A. I. Ahmed, "A new parameter free filled function for solving unconstrained global optimization problems," *Int. J. Comput. Math.*, vol. 98, no. 1, pp. 106–119, Jan. 2021. doi: [10.1080/00207160.2020.1731484](https://doi.org/10.1080/00207160.2020.1731484).
- [27] A. Sahiner, T. Ermis, and M. W. Awan, "A new filled function for global optimization," *Analele Stiintifice ale Universitatii Ovidius Constanta, Seria Matematica*, vol. 31, no. 3, pp. 207–220, Sep. 2023.