(Received 18 August 2023, Accepted 20 November 2023, Scheduled 3 December 2023)

PMSWG Parameter Identification Method Based on Improved Operator Genetic Algorithm

Zhun Cheng[1, *], Chao Zhang[2], and Yang Zhang[2]

[1] Hunan Railway Professional Technology College, Zhuzhou 412001, China
[2] Hunan University of Technology, Zhuzhou 412007, China

ABSTRACT: Permanent Magnet Synchronous Wind Generator (PMSWG) parameter identification method with improved operator genetic algorithm is proposed for the influence of perturbations caused by mechanical parameter changes on the dynamic performance of motor speed control system. Firstly, current with $i_d = 0$ and $i_d \neq 0$ are injected into axis d respectively to design the fitness function. Through quantum coding, the genetic algorithm can obtain better population and fitness in the early stage, and find better solutions in the search space. At the same time, the cross method of two random numbers is used to make the cross variable not restricted in a range, which enhances the global search ability. Finally, the update strategy of hybrid mutation composed of Gaussian mutation and Cauchy mutation is introduced to ensure the global search ability of the algorithm, and the accuracy of the optimization results is improved. Experiments show that the proposed method avoids local optimization and achieves global optimization, which can further improve the convergence speed and identification accuracy of the algorithm.

1. INTRODUCTION

PMSWG has the advantages of high efficiency, high reliability, small size, and light weight, which is widely used in wind power generation. The design of PMSWG high performance controller is related to the generator parameters, but the generator parameters are subject to temperature, load variation, voltage fluctuations, friction and mechanical vibrations, and magnetic field drift, resulting in degraded control performance [1–4]. For such issues, scholars have done a lot of research and proposed different parameter identification methods [5, 6].

According to the identification time in system operation, parameter identification methods are divided into offline identification and online identification [7, 8]. Offline identification method can be applied to different types of motors, but it is necessary to choose the appropriate method according to the actual situation of motors [9]. In [10], modal analysis is adopted to analyze the torsional vibration frequency response of the motor, find out the sound source of the motor, and obtain the vibration mode of the motor. By studying modal factors in detail, a mathematical method for calculating motor noise is presented. In [11], in single-phase AC experiment and no-load test, to reduce errors, discrete Fourier transform is adopted for signal processing to obtain the frequency components and amplitude of current and voltage signals, and then identify the stator rotor resistance, stator rotor leakage, and mutual inductance of the motor. Although the parameters can be accurately identified, the calculation is tedious. In [12] according to the characteristics of the stator current harmonic component change under the stator revolution condition, the sixth approximate coefficient is decomposed to construct the stator revolution short circuit, and a method based on discrete wavelet transform is proposed. In [13], finite element analysis was used to identify the field inductance and synchronous inductance of permanent magnet synchronous motor under rated load and overload. This method obtained electrical parameter identification values by solving multiple differential equations simultaneously, which was time-consuming and inefficient.

Online identification can collect and analyze the operating parameters of the motor in real-time, provide timely feedback on the status and performance index of the motor, and facilitate real-time monitoring and control of the motor [14, 15]. At present, the main methods for online identification of motor parameters are recursive least squares (RLS), extended Kalman filter (EKF), model reference adaptive (MRAS), and artificial intelligence algorithms. Reference [16] proposed a recursive least squares method for parameter identification based on LS, analyzed the factors affecting the identification accuracy, and the identification results were able to approach the actual values, but the increase in the amount of data led to the saturation of the data, which made the accuracy of the identification decrease. In [17, 18], a model reference based adaptive identification method is proposed, which is fast and easy to implement after improving the MRAS algorithm, but faces the problem of adaptive law selection, which has a great impact on the results of parameter identification, and the correct convergence speed of the parameters depends on the initial values of the parameters. References [19, 20] proposed an extended Kalman filter (EKF)-based discrimination method, which is a recursive filtering method, and although it improves the accuracy of the system and solves the noise sensitivity problem, the computation process requires a large number of matrix calculations, resulting in slow convergence. In [21], a method for determining

* Corresponding authors: Zhun Cheng (120277982@qq.com).

the electrical parameters of a three-phase asynchronous motor using a genetic algorithm technique was proposed, which successfully estimated the static current curve and the electromechanical conjugate current curve. In [22], an improved adaptive genetic algorithm was used for motor parameter identification, how to overcome the shortcomings of traditional genetic algorithm (GA), such as slow evolution and long computation time, when solving efficient electrical engineering problems. In [23], to solve the problem that the basic GA is prone to local extreme values, the improved adaptive GA is applied to model parameter identification, and polynomial fitting is used to describe the relationship between motor position and identification parameters.

Considering the aforementioned issues, this paper presents a parameter identification method for PMSWG based on genetic algorithm with improved operators. The remainder of the paper is organized as follows. Section 2 designs the identification equations. Section 3 provides improvements the operators of the genetic algorithm. Then, Section 4 shows the experimental results on the RT-LAB platform to verify the correctness of the theoretical analysis and simulation analysis. Finally, the article draws conclusions in Section 5.

2. DESIGN OF THE PMSWG IDENTIFICATION EQUATION

2.1. Mathematical Model of PMSWG

$$\begin{cases} u_d = Ri_d + L_d\frac{di_d}{dt} - L_qP\omega i_q \\ u_q = Ri_q + L_q\frac{di_q}{dt} + L_dP\omega i_d + \varphi_f P\omega \\ T_e = \frac{3}{2}P\left[\varphi_f i_q + (L_d - L_q)\, i_d i_q\right] \\ J\frac{d\omega}{dt} = T_L - B\omega - T_{\mathrm{e}} \end{cases} \tag{1}$$

where P is the polar logarithm; ω is the mechanical angular velocity; u_d, u_q, i_d, i_q are the dq axis stator voltage and current; R, φ_f, L_d, L_q represent the motor winding resistance, motor magnetic chain, d axis and q axis inductance, respectively; mechanical parameters B, J are the coefficient of viscous friction, rotational inertia, respectively; T_{e} is the electromagnetic torque; T_L is the load torque.

Under no-load condition, that is, $T_L = 0$, $i_d = 0$, it can be simplified as:

$$J\frac{d\omega}{dt} = -\frac{3}{2}P\varphi_f i_q - B\omega \tag{2}$$

2.2. Electrical Parameter Design

Electrical parameter identification usually refers to the analysis of known input and output data to determine the unknown parameter values of a circuit or system. In parameter identification, $i_d = 0$ vector control can be used for electrical parameter design. When using $i_d = 0$ vector control for electrical parameter design, current and voltage sensors are used to measure the current and voltage of the motor. The symbols denoted by "0" or "1" in the subscript indicate that the d-axis is injected with $i_d = 0$ or $i_d \neq 0$, respectively. The reference model is as follows:

$$\begin{cases} u^*_{q0}(k) = Ri_{q0}(k) + \varphi_f\omega_0(k) \\ u^*_{q1}(k) = Ri_{q1}(k) + L_d\omega_1(k)\, i_{d1}(k) + \varphi_f\omega_1(k) \end{cases} \tag{3}$$

2.3. Mechanical Parameter Design

The estimation of mechanical parameters was achieved by designing the identification model for steady-state conditions and starting acceleration conditions. The steady-state estimation of B, with d/dt = 0, can be simplified as:

$$-\frac{3}{2}P\varphi_f i_q = B\omega \tag{4}$$

When d/dt=0, J does not appear in this equation. To obtain J so that d/dt $\neq$ 0, the motor is required to run at constant acceleration for a period of time during the start-up test, which can be discretized as follows:

$$J\frac{\omega(k+T_s) - \omega(k)}{T_s} = -\frac{3}{2}P\varphi_f i_q(k) - B\omega(k) \tag{5}$$

where T_s is the sampling time.

2.4. Fitness Function Design

$$\begin{cases} f_1\left(\hat{R}, \hat{\varphi}_f\right) = \frac{1}{n}\sum\limits_{k=1}^{n}\left|u^*_{q0}(k) - \hat{u}_{q0}(k)\right| \\ f_2\left(\hat{R}, \hat{\varphi}_f\right) = \frac{1}{n}\sum\limits_{k=1}^{n}\left|u^*_{q1}(k) - \hat{u}_{q1}(k)\right| \\ f_3\left(\hat{\varphi}_f, \hat{B}\right) = \frac{1}{n}\sum\limits_{k=1}^{n}\left|\omega^*(k) - \hat{\omega}(k)\right| \\ \qquad\qquad = \frac{1}{n}\sum\limits_{k=1}^{n}\left|\omega^*(k) + \frac{3}{2}p\hat{\varphi}_f i_q/\hat{B}\right| \\ f_4\left(\hat{J}, \hat{B}, \hat{\varphi}_f\right) = \frac{1}{M}\sum\limits_{k=1}^{m}\left|iq^*(k) - \hat{i}q(k)\right| \\ \qquad\qquad = \frac{1}{m}\sum\limits_{k=1}^{m}\left|iq^*(k) + \frac{\hat{J}\frac{\omega(k+Ts)-\omega(k)}{Ts} + \hat{B}\omega(k)}{1.5p\varphi_f}\right| \end{cases} \tag{6}$$

where $u^*_{q0}(k)$ is the data sampled at the kth time; $u^*_{q1}(k)$ is the data sampled at the kth time; $\omega^*(k)$, $iq^*(k)$ are the data sampled at the kth time.

Let $\hat{\theta} = (\hat{R}, \hat{\varphi}_f, \hat{B}, \hat{J})$, and the required parameters can be solved by calculating the fitness function value.

$$f\left(\hat{\theta}\right) = \sum\nolimits_{i=1}^{4} a_i f_i \tag{7}$$

where a_i is the weighting coefficient. Since the identification parameters are all important, the weighting coefficient is 1/4.

3. PMSWG PARAMETER IDENTIFICATION METHOD BASED ON IOGA

3.1. Using Quantum Bits for Coding

Traditional genetic algorithms generally use binary coding, and although the binary coding is simple and intuitive, there are

mapping errors in the discretization of continuous functions. If the individual is short, it may not meet the accuracy requirements. However, if the individual encoding length is long, it can improve the accuracy, but increase the difficulty of decoding.

In this paper, quantum genetic coding will be used to encode the solved parameters, and the relationship between parameters and fitness function is represented by one qubit. Quantum genetic algorithm is carried out on multiple populations simultaneously, and each population corresponds to a solution. When the population size is larger, the genetic complexity of the population is higher, and the possibility of obtaining the optimal solution is greater. The specific implementation can be adjusted according to the size of the problem to be solved. The quantum chromosome code is denoted as U, as shown in Equation (8).

$$U = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \dots & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & \dots & \beta_n \end{bmatrix} \tag{8}$$

Qubits differ from classical bits in that they can be in a superposition of two quantum states at the same time, such as:

$$|\varphi\rangle = \alpha|0\rangle \beta|1\rangle \tag{9}$$

(α, β) is two amplitude constants.

$$|\alpha|^2 + |\beta|^2 = 1 \tag{10}$$

where $|0\rangle$ and $|1\rangle$ represent spin-down and spin-up states, respectively. A qubit can contain both state and state information.

3.2. Linear Sorting Selection

In the linear sorting selection method, the individuals are first sorted according to their fitness values. The best individual is ranked N and the worst individual ranked 1. Then the selection probability is distributed linearly according to the individual's rank.

$$P_i = \frac{1}{N}\left(n^- + \left(n^+ - n^-\right)\frac{i-1}{N-1}\right); \quad i \in \{1, \dots, N\} \tag{11}$$

where P_i is the probability of selecting an individual, n^- the probability of selecting the worst individual, and n^+ the probability of selecting the best individual. Even if the probabilities are the same, each individual gets a different rank. The ranking process is divided into two steps; in the first step, the overall is ranked, and in the second step, the ranks are assigned in the order corresponding to the proportional choices.

3.3. Based on the Crossover of Two Random Numbers

The traditional genetic algorithm generates a pair of offspring by simulating binary crossover using crossover of each selected pair of parents. The offspring is as follows:

$$\begin{cases} x_i^{(1,t+1)} = 0.5\left[(1+\alpha_i)\, x_i^{(1,t)} + (1-\alpha_i)\, x_i^{(2,t)}\right] \\ x_i^{(2,t+1)} = 0.5\left[(1-\alpha_i)\, x_i^{(1,t)} + (1+\alpha_i)\, x_i^{(2,t)}\right] \end{cases} \tag{12}$$

where $x_i^{(1,t+1)}$, $x_i^{(2,t+1)}$ are the i-th variables of the two parents; $x_i^{(1,t)}$, $x_i^{(2,t)}$ are the $i(i = 1, 2, \dots, N)$ variables of the two offspring; α_i is a dynamic parameter; and the expressions are as follows:

$$\alpha_i = \begin{cases} (2y_i)^{\frac{1}{\eta+1}}, & \text{if } \mathrm{y}_i \le 0.5; \\ \left(\frac{1}{2(1-y_i)}\right)^{\frac{1}{\eta+1}}, & \text{Otherwise.} \end{cases} \tag{13}$$

where η represents the distribution index, and y_i is two random numbers uniformly generated in the range [0, 1].

Any crossover variable is restricted to a range relative to the corresponding parent variable, which weakens the global search capability to some extent, and to overcome this, a crossover operator based on two random numbers will be used and updated with the following crossover offspring.

$$\begin{aligned} x_i^{(1,t+1)} &= \begin{cases} x_i^{(1,t)} + r_1\left(x_i^{(2,t)} - x_i^{(1,t)}\right), & \text{if } \mathrm{y}_1 < p_c; \\ x_i^{(1,t)}, & \text{Otherwise.} \end{cases} \\ x_i^{(2,t+1)} &= \begin{cases} x_i^{(2,t)} + r_2\left(x_i^{(1,t)} - x_i^{(2,t)}\right), & \text{if } \mathrm{y}_2 < p_c; \\ x_i^{(2,t)}, & \text{Otherwise.} \end{cases} \end{aligned} \tag{14}$$

where $x_i^{(1,t+1)}$, $x_i^{(2,t+1)}$ are the $i(i = 1, 2, \dots, N)$ variables of the two parents; $x_i^{(1,t)}$, $x_i^{(2,t)}$ are the $i(i = 1, 2, \dots, N)$ variables of the two offspring; y_1, y_2 are two random numbers uniformly generated in the range [0, 1]; p_c is the crossover rate; r_1, r_2 expressions are given as follows:

$$\begin{aligned} r_1 &= \begin{cases} u\left(\frac{1}{2}, \frac{\sqrt{3}}{6}\right), & \text{if } \mathrm{y}_3 < 0.5; \\ g\left(0, 1\right), & \text{Otherwise.} \end{cases} \\ r_2 &= \begin{cases} u\left(\frac{1}{2}, \frac{\sqrt{3}}{6}\right), & \text{if } \mathrm{y}_4 < 0.5; \\ g\left(0, 1\right), & \text{Otherwise.} \end{cases} \end{aligned} \tag{15}$$

where y_3, y_4 are two random numbers uniformly generated in the range [0, 1]; $u(\frac{1}{2}, \frac{\sqrt{3}}{6})$ is a uniform random number with mean $\frac{1}{2}$ and standard deviation $\frac{\sqrt{3}}{6}$; $g(0, 1)$ denotes a Gaussian random number with mean 0 and standard deviation 1.

3.4. Introduction of Gaussian Mutation and Cauchy Mutation Strategies

Traditional genetic algorithm is an optimization algorithm based on genetics and evolutionary theory, in which mutation is an important operation to increase the diversity of the population by changing the genes of certain individuals, thus facilitating global search. However, too high or too low mutation probabilities can affect the performance of the algorithm. When the mutation probability is too high, the algorithm will rely excessively on the mutation operation, resulting in a too random population and lack of effective local search ability, which may lead to slow convergence of the algorithm or fall into local optimal solutions. When the mutation probability is too low, the algorithm may fall into a local optimal solution, and it is difficult to jump out of that optimal solution to find a more optimal solution.

The traditional genetic algorithm mutation operation selects the j-th gene of the i-th individual for mutation, and the mutation operation is as follows:

$$\mathrm{x}_{ij} = \begin{cases} x_{ij} - (x_{ij} - x_{\min}) * (1 - f(g)), & r \leq 0.5 \\ x_{ij} + (x_{\max} - x_{ij}) * (1 - f(g)), & r > 0.5 \end{cases} \quad (16)$$

where $x_{\max}$, $x_{\min}$ are the upper and lower bounds of the gene x_{ij}; $f(g) = r_2(1 - g/\mathrm{g}_{\max})^2$; r_2 is a random number; g is the current number of iterations; $\mathrm{g}_{\max}$ is the maximum number of evolutions; r is a random number between [0, 1].

To solve this problem, the variance probability can be dynamically adjusted according to the fitness value of the population. Specifically, the population can be divided into two subpopulations according to the fitness value, and both Gaussian mutation operator and Cauchy mutation operator can be introduced.

Gaussian mutation operator is a common probability distribution function, which describes the distribution of a set of data by introducing a Gaussian distribution function, also known as a normal distribution function, into the mutation operator. The Gaussian mutation operator can be viewed as a perturbation of the Gaussian distribution of gene values by the mutation operation. Specifically, for a given gene value x, its mutated value can be calculated by the following equation:

$$x' = x + \Delta x \quad (17)$$

where Δx is a Gaussian distributed random number obeying a mean of 0 and a standard deviation of σ, with a density function:

$$f_G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \quad (18)$$

where x is a random variable, and the range of independent variables is $[-\infty, \infty]$.

The Cauchy mutation operator introduces a Cauchy distribution function into the mutation operator, and the Cauchy mutation operator treats the mutation operation as a perturbation of the Cauchy distribution of gene values. Specifically, for a gene value x, the mutated value can be calculated by the following equation.

$$x' = x + \Delta x \quad (19)$$

where Δx is a random number obeying the Cauchy distribution with a density function.

$$f_C(x) = \frac{1}{\pi} \frac{\gamma}{(x - x_0)^2 + \gamma^2} \quad (20)$$

where x is a random variable, and the independent variable ranges from $[-\infty, \infty]$. γ and x_0 are constant coefficients, $\gamma > 0$, and when $\gamma = 1$, $x_0 = 0$, x obeys the standard Cauchy distribution.

The Cauchy and Gaussian distributions are shown in Fig. 1.

The main idea of Gaussian mutation is to generate a new solution randomly in the vicinity of the current individual, so that the current individual can search within a closer distance from itself. This prevents the algorithm from falling into a local optimum and improves the local search ability of the algorithm. The Gaussian distribution has a bell-shaped curve with the center point being the current individual, and the probability of a randomly generated new solution decreases as it gets farther from the center point, which allows the algorithm to search the surrounding solution space in a more targeted manner. The main idea of the Cauchy mutation is to give less well-adapted individuals a greater probability of moving away from itself, allowing less well-adapted individuals to jump out of the worse solution region, enhancing the global search capability of the algorithm. The shape of the Cauchy distribution is a spike curve with the center point being the current individual, and the probability of a new randomly generated solution does not decrease as it gets farther away from the center point, which makes the algorithm more exploratory and easier to jump out of the local optimal solution.

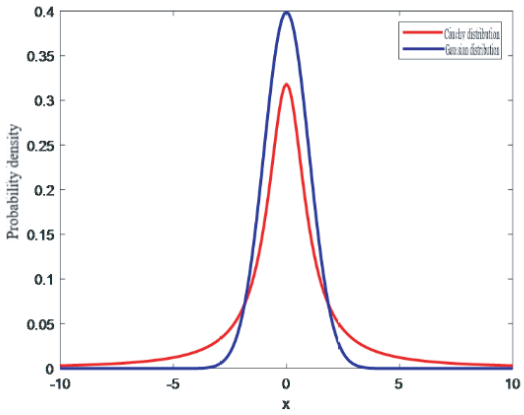

FIGURE 1. Cauchy distribution and Gaussian distribution diagram.

$$x_i' = \begin{cases} x_i + \text{range} \cdot p(x_i) \cdot G_i(0,1) & p(x_i) \leq 0.5 \\ x_i + \text{range} \cdot p(x_i) \cdot C_i(1,0) & p(x_i) > 0.5 \end{cases} \quad (21)$$

where $p(x_i) = \frac{f(x_i) - f_{\min}}{f_{\max} - f_{\min}}$ is the proportional transformation function; $f_{\max}$, $f_{\min}$ are the maximum and minimum values of the individual function values in the population, respectively; $f(x_i)$ is the fitness function value of the individual. x_i, x_i' are the i-th chromosome before and after the mutation, respectively, and range is a uniform random number between [0, 1].

The populations after the Gaussian mutation operation and the Cauchy mutation operation are merged into one population, and the individuals are moved closer to the population optimum by Equation (22). The flowchart of population mutation after the merger is shown in Fig. 2.

$$x_i = (1 - c)\, x_i + c x_t \quad (22)$$

where c is the individual movement speed and generally chosen as 0.1–0.6.

3.5. Basic Steps of the Improved Operator Genetic Algorithm

The coding, selection, crossover, and mutation of the traditional genetic algorithm are improved by the description in the previous sections, and the detailed steps of the improved operator genetic algorithm are shown in Fig. 3.

(1) Initialization of populations by quantum bit encoding and setting of parameters;

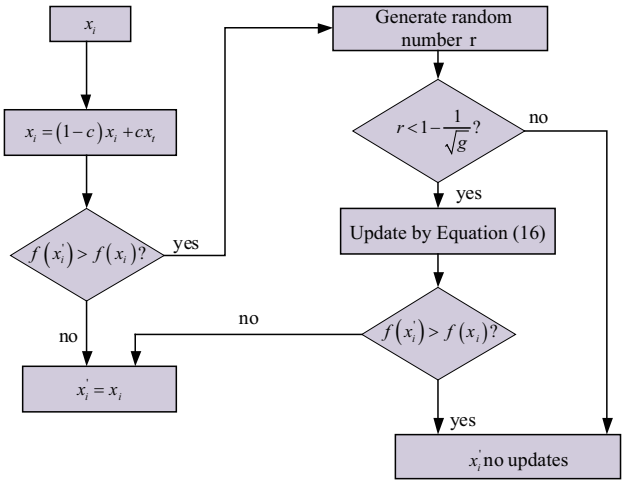

FIGURE 2. Mixed mutation flow chart.

(2) Set the fitness function $f(x_i)$, and calculate the value of the fitness function and record the optimal individuals x_{best};

(3) Linear ranking selection of the initialized population, where individuals with high fitness are selected with high probability;

(4) Two random crossovers of individuals with high adaptation after selection were performed by Equation (16);

(5) The mutation operation is performed on the population after the crossover operation in step (4), and the proportional transformation function is used to determine whether the population will undergo Gaussian mutation or Cauchy mutation, when the population will undergo Gaussian mutation and vice versa;

(6) Combine the populations after Gaussian and Cauchy mutations into a new population, and in order to enhance the merit-seeking ability of the population, perform mutation operations on the combined populations according to the steps in Fig. 3 and record the most individuals x_i after mutation;

(7) Since the problem to be solved is about minimizing the value of fitness, when comparing the relationship between the previous generation and the new generation, if $f(x_{best}) < f(x_t)$, then $n = 0$, $x_{best} = x_t$, otherwise $n = n + 1$; determine whether the number of iterations n of the optimal solution update is greater than 20, and if yes, then update using Equation (21) and obtain the new population with $n = 0$;

(8) Judge the relationship between the current iteration number g and the maximum iteration number g_{max}, if $g < g_{max}$, then $g = g + 1$; otherwise, record the optimal individual x_{best} and end.

3.6. Improved of GA (IOGA) Parameter Identification Principle

With the above operation steps of the improved operator, the principle of IOGA identification is shown in Fig. 4.

(1) Acquisition of electrical signals, initial population and setting of initial parameters, setting the range of parameters to be identified;

(2) Bringing the initial population into Equations (4) and (5) and obtaining the corresponding fitness values;

(3) Finding the optimal individual for the initial individual based on the calculated fitness value;

(4) The optimal individual found is substituted into Equation (14) for linear ranking selection, e.g,:

$$\begin{cases} \hat{J}_i^{(1,t+1)} = 0.5\left[(1+\alpha_i)\,\hat{J}_i^{(1,t)} + (1-\alpha_i)\,\hat{J}_i^{(2,t)}\right] \\ \hat{J}_i^{(2,t+1)} = 0.5\left[(1-\alpha_i)\,\hat{J}_i^{(1,t)} + (1+\alpha_i)\,\hat{J}_i^{(2,t)}\right] \end{cases} \tag{23}$$

(5) The individuals selected by linear sorting were subjected to Gaussian mutation and Cauchy mutation. If the proportional transformation function $p(x_i) \leq 0.5$, the population will undergo Gaussian mutation, and vice versa, the population will undergo Cauchy mutation.

(6) The populations after Gaussian mutation and Cauchy mutation are combined into one population and updated by Equation (22), like the optimal solution approached, and if the individual with the lowest fitness value is obtained, it replaces the previous generation of individuals; otherwise, it is updated by Equation (21), and if the individual with the lowest fitness value is obtained, it replaces the previous generation of individuals; otherwise, the original individuals are not updated, e.g.,

$$\hat{J}_i' = \begin{cases} \hat{J}_i + \text{range} \cdot p\left(\hat{J}_i\right) \cdot G_i(0,1) & p\left(\hat{J}_i\right) \leq 0.5 \\ \hat{J}_i + \text{range} \cdot p\left(\hat{J}_i\right) \cdot C_i(1,0) & p\left(\hat{J}_i\right) > 0.5 \end{cases} \tag{24}$$

$$or\ \hat{J}_i = (1-c)\,\hat{J}_i + c\hat{J}_t \tag{25}$$

(7) Solve for the minimum value of the fitness function according to Equation (7), if $f(x_{best}) < f(x_t)$, then $n = 0$, $x_{best} = x_t$; otherwise, $n = n + 1$; determine whether n is greater than 20, and if yes, then update using Equation (21) and obtain a new population with $n = 0$;

(8) If the termination condition is satisfied, the optimal parameters are output; otherwise, go to step (4).

4. EXPERIMENTAL RESULTS AND ANALYSIS

To verify the feasibility of IOGA-based parameter identification, the compiled Simulink simulation model was downloaded to RT-LAB to implement hardware-in-the-loop simulation experiments of the PMSWG drive system. Fig. 5 is a block diagram of the vector control strategy with id = 0, and Fig. 6 is RT-LAB experimental platform. PMSWG parameters as shown in Table 1.

PMSWG is carried out by improving the genetic operator method. Method I introduces quantum bit encoding with linear ranking selection IOGA1; method II introduces the crossover of two random numbers IOGA2; method III introduces Gaussian

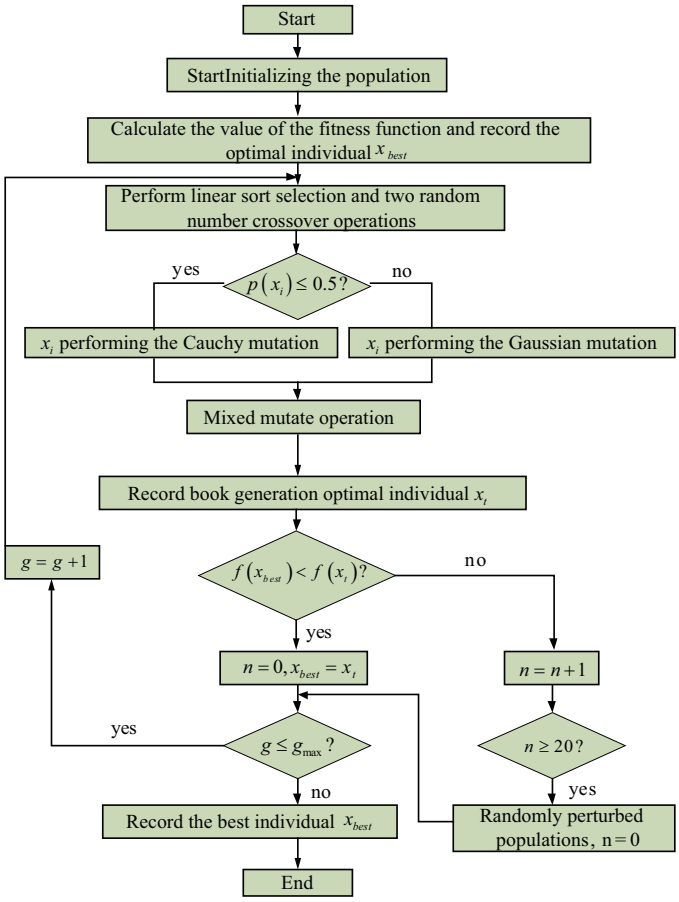

FIGURE 3. Flow chart of improved operator genetic algorithm.

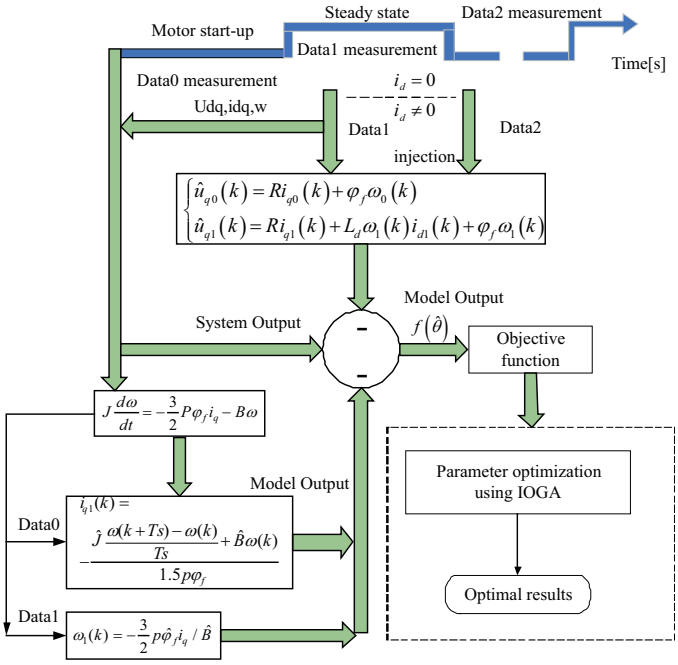

FIGURE 4. Parameter identification schematic diagram.

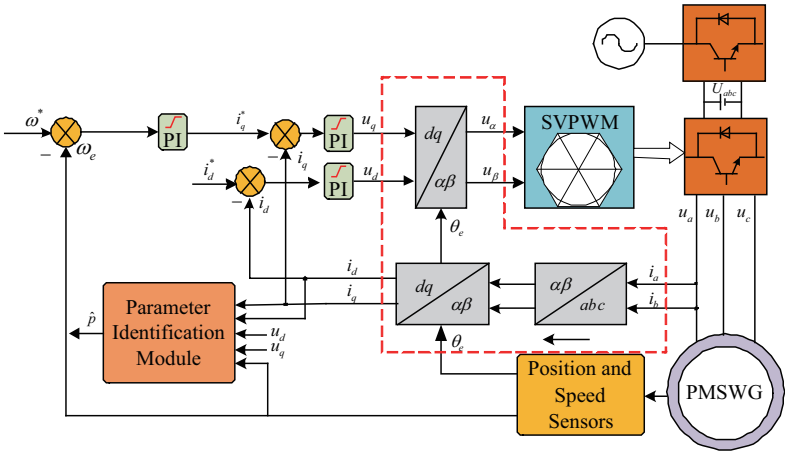

FIGURE 5. Block diagram of the vector control strategy with id = 0.

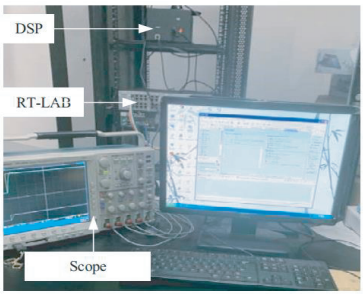

FIGURE 6. RT-LAB experimental platform.

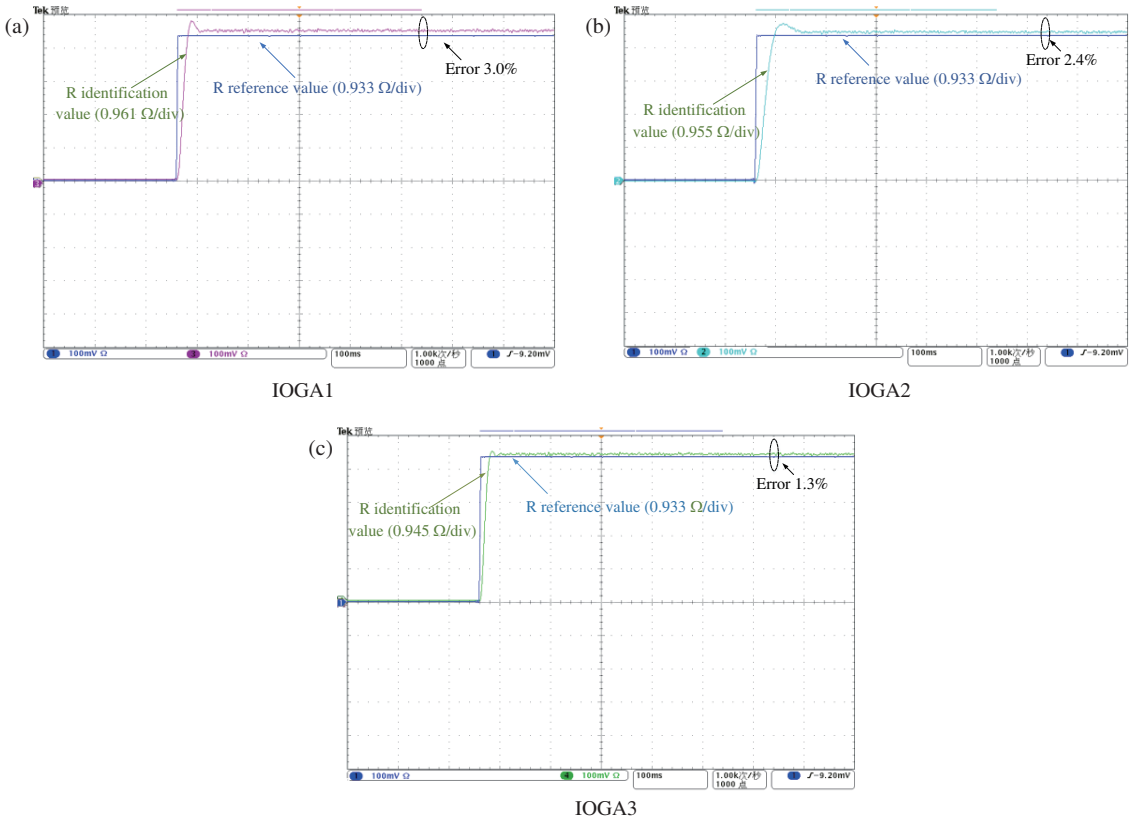

FIGURE 7. Stator resistance identification curve.

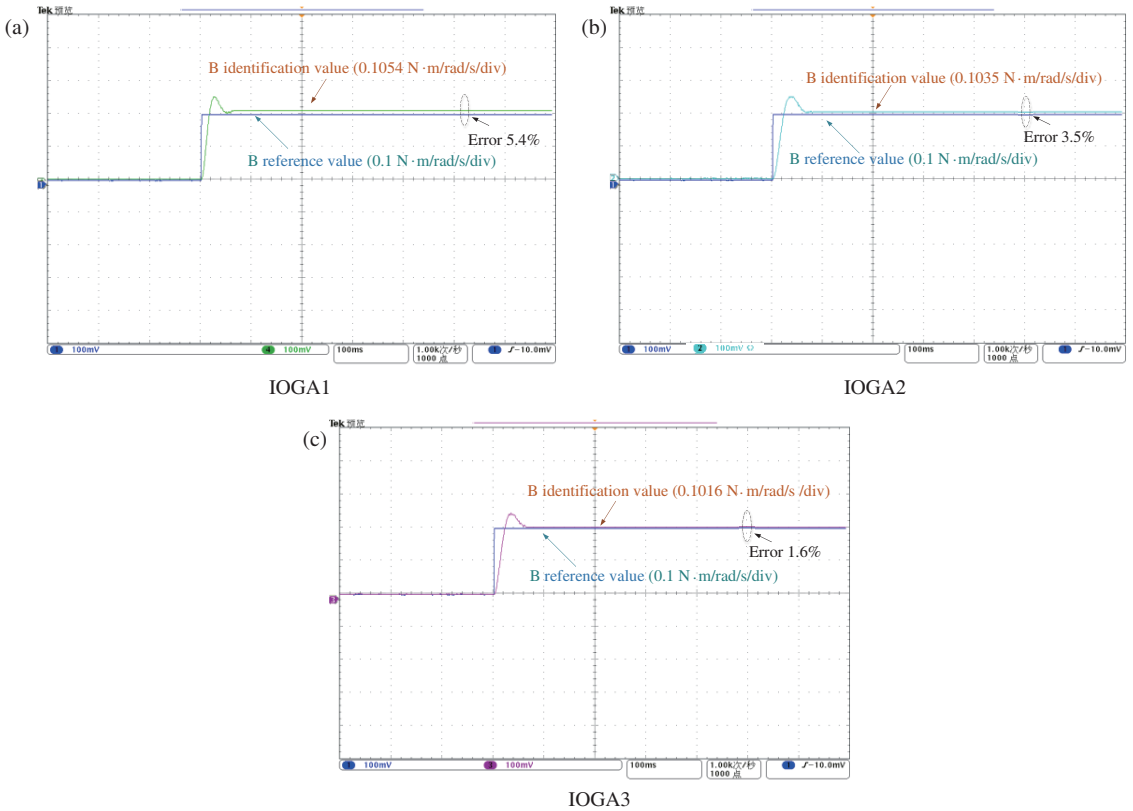

FIGURE 8. Coefficient of viscous friction identification curve.

TABLE 1. IPMSWG parameter table.

Parameters	Value
Polar number	4
Stator resistance/Ω	0.933
Stator d axis inductance/mH	5.20
Stator q axis inductance/mH	11.5
Permanent magnet flux/Wb	0.175
Torque inertia/kg · m^2	0.003
Rated voltage/V	380
Rated torque/(N · m)	10
Rated speed/(r/min)	1000
Rated power/kW	4
Coefficient of viscous friction B/N · m/rad/s	0.1

TABLE 2. Comparison of three algorithms simulation.

Parameters	IOGA1	IOGA2	IOGA3
Stator resistance/Ω	0.961	0.955	0.945
error/%	3.0	2.4	1.3
Torque inertia/kg · m^2	3.12e-3	2.895e-3	2.933e-3
error/%	4	3.5	2.2
Permanent magnet flux/Wb	0.1822	0.18	0.177
error/%	4.1	2.9	1.1
COVF/N · m/rad/s	0.1054	0.1035	0.1016
error/%	5.4	3.5	1.6
Identification time/(ms)	62	55	51
Fitness value	0.84	0.71	0.67

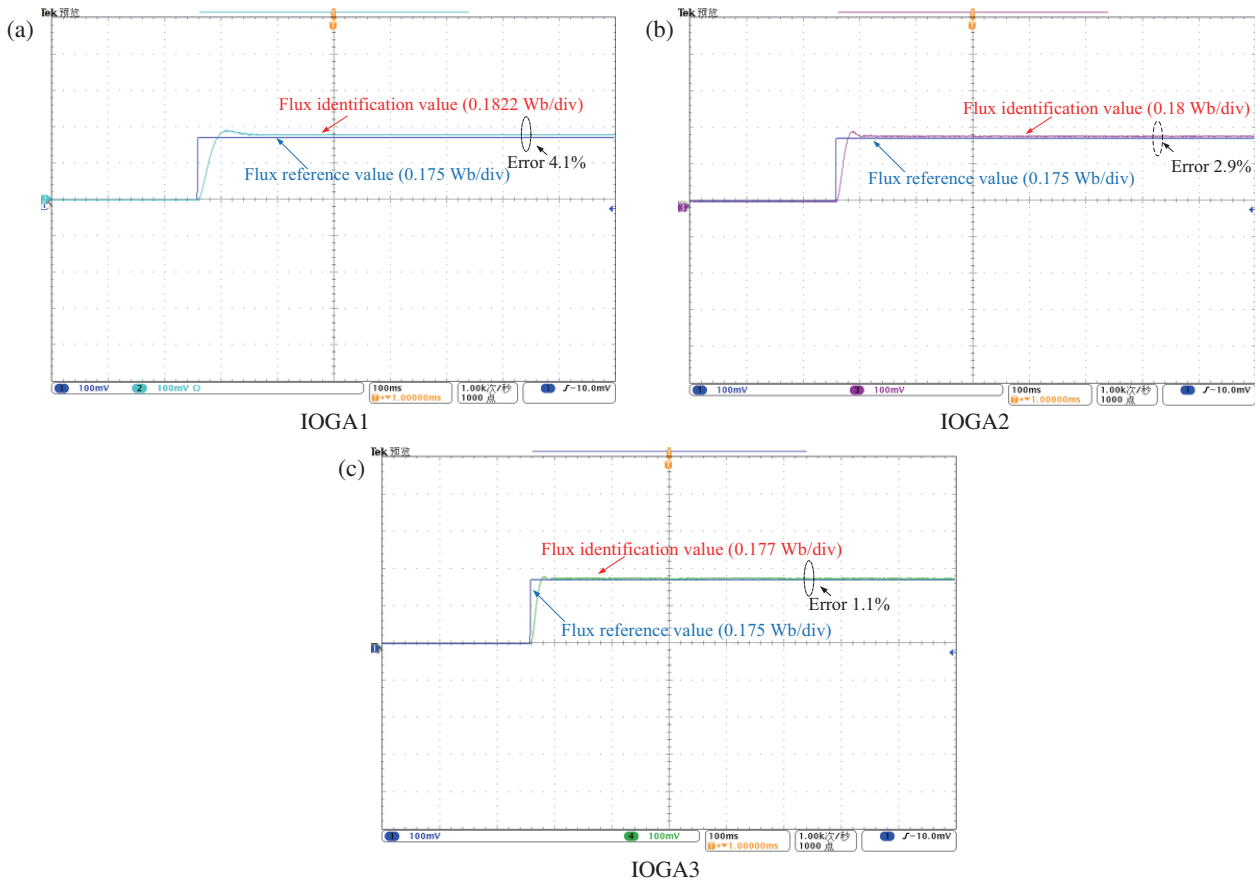

FIGURE 9. Permanent magnet flux identification curve.

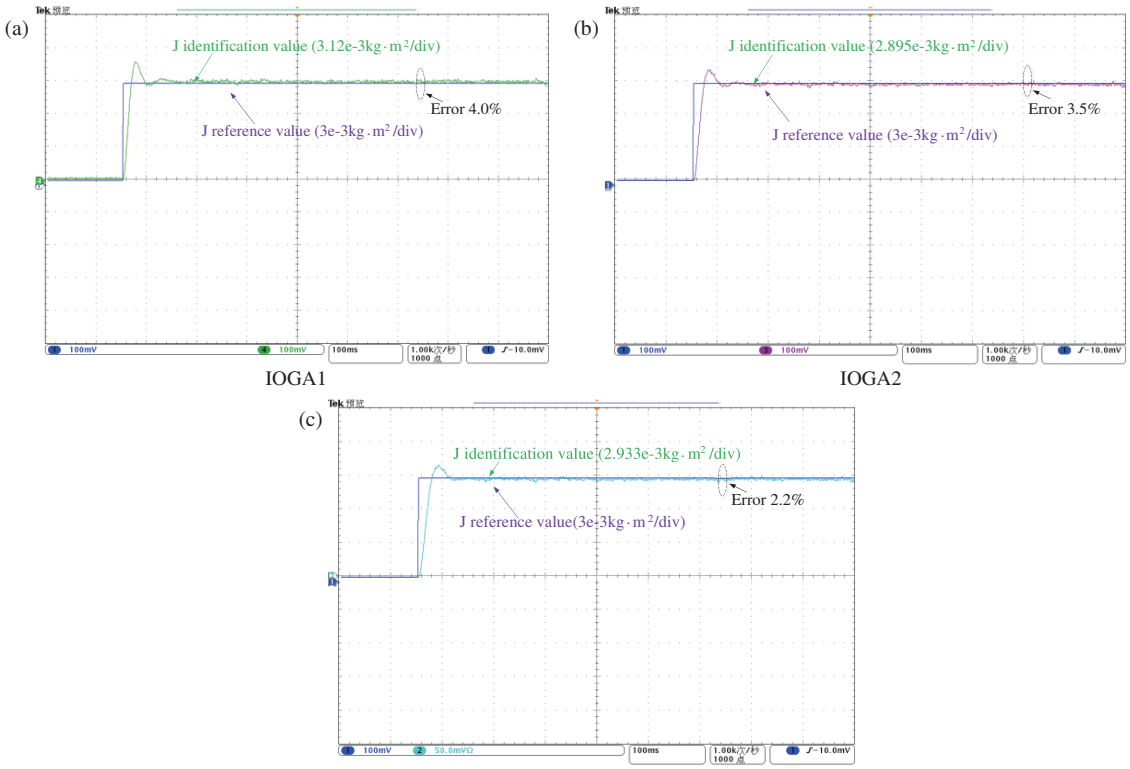

FIGURE 10. Torque inertia identification curve.

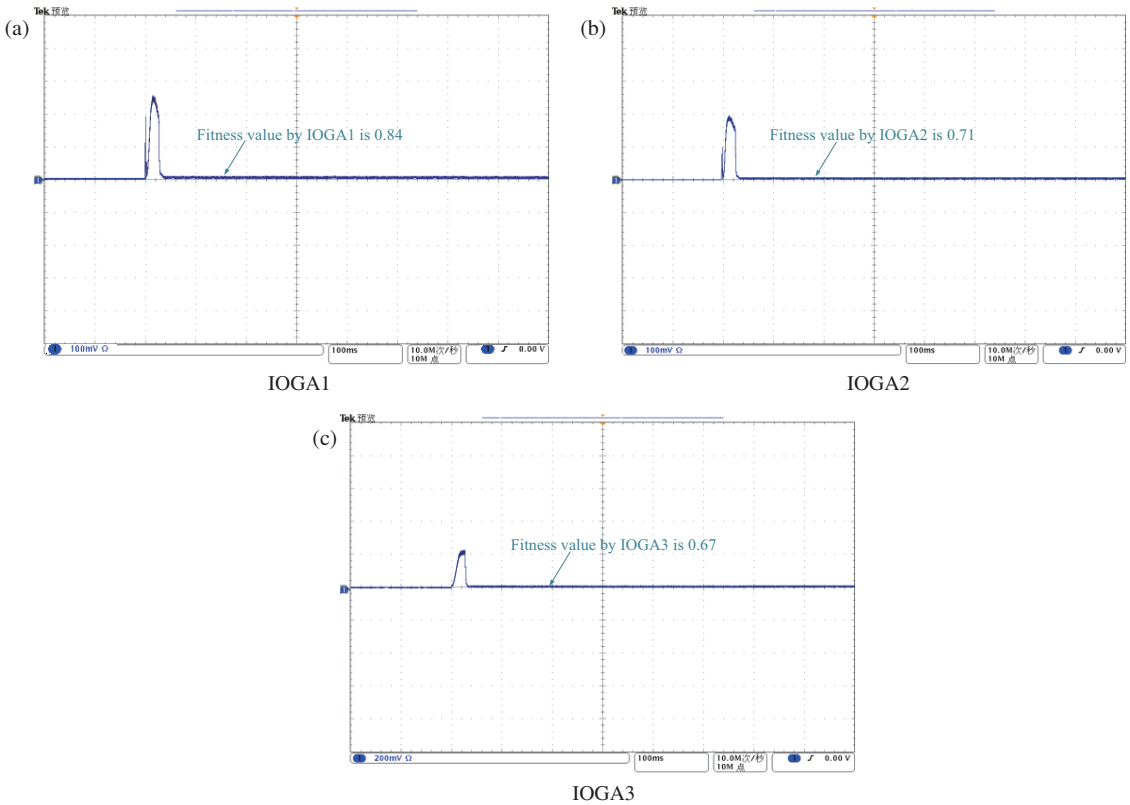

FIGURE 11. Fitness function value curve.

mutation and Cauchy mutation strategy IOGA3. The comparison of three algorithms' simulations is shown in Table 2.

The identification results of the three methods are shown in Fig. 7 to Fig. 10. Since the values of d-axis inductance, q-axis inductance, and permanent magnet flux are relatively small, the

stator resistance, torque inertia, coefficient of viscous friction, and permanent magnet chain are enlarged by 10, 1000, 500, and 200 times respectively for a simulation time of 0.5 s in order to better observe the identification results.

In Fig. 7 to Fig. 8, the identification results of three algorithms for stator resistance and viscous friction coefficient are shown respectively. In the stator resistance discrimination, the discrimination errors of IOGA1, IOGA2, and IOGA3 are 3.0%, 2.4%, and 1.4%, respectively. In IOGA1 and IOGA2, although the accuracy of IOGA2 is higher than that of IOGA1, it is slower than IOGA1 in terms of convergence speed, which may be because IOGA1 selects to the optimal individual at the initial population. IOGA3's convergence and discrimination accuracy are better than IOGA1 and IOGA2.

In the coefficient of viscous friction identification, the discrimination errors of IOGA1, IOGA2, and IOGA3 are 5.4%, 3.5%, and 1.6%, respectively. The larger discrimination accuracy error of IOGA1 is likely due to the small population size, which causes the algorithm to converge to the local optimal solution too early. The discrimination accuracy error of IOGA3 is the smallest, thanks to the mixed variance of Gaussian and Cauchy mutation, which makes the search range of the population larger.

In Fig. 9 to Fig. 10, the identification results of the three algorithms for permanent magnet flux and torque inertia are shown respectively. In the identification of permanent magnet flux, the identification errors of IOGA1, IOGA2, and IOGA3 are 4.1%, 2.9%, and 1.1%, respectively. Among the three methods, IOGA1 is worse than IOGA2 and IOGA3 in terms of convergence speed and identification accuracy, because the population diversity of IOGA1 is too small and easy to fall into local optimum.

In the torque inertia identification, the identification errors of IOGA1, IOGA2, and IOGA3 are 4.0%, 3.5%, and 2.2%, respectively. Among the three methods of identification, the convergence speed and identification accuracy of IOGA1 and IOGA2 are similar, and the identification accuracy and convergence speed of IOGA3 are the best among the three algorithms, and the strategy of introducing mixed variants makes the initial population diverse enough to ensure the breadth and depth of the search space of the algorithm.

The fitness function curves of the three algorithms in this paper are shown in Fig. 11, which converge to 0.67 at 51 ms by IOGA3 discrimination method. The fitness function values of IOGA1 and IOGA2 discrimination methods are 0.71 and 0.84, respectively, and the convergence times are 55 ms and 62 ms, respectively. Among the three algorithms, due to the introduction of mixed variance in IOGA3 discrimination method strategy is introduced in IOGA3, which results in the smallest fitness value and fastest convergence speed.

5. CONCLUSION

PMSWG parameter identification method with improved operator genetic algorithm is proposed to address the problem of the influence of perturbations caused by mechanical parameter changes on the dynamic performance of motor speed control systems. The method is able to identify electrical and mechanical parameters with higher accuracy of the identification model. Through experimental analysis, the following conclusions are drawn:

(1) The IOGA method proposed in this paper can identify the stator resistance R, permanent magnet flux, torque inertia J, and coefficient of viscous friction B of PMSWG online and in real-time.

(2) The method in this paper has better parameter identification capability; the identification error does not exceed 2.2%; the convergence speed is 51 ms; and the convergence speed and parameter accuracy are better than IOGA1 and IOGA2 identification methods.

ACKNOWLEDGEMENT

This work was supported by the Natural Science Foundation of Hunan Province of China under Grant Number 2022JJ50094.

REFERENCES

[1] Singh, N. and V. Agrawal, "A review on power quality enhanced converter of permanent magnet synchronous wind generator," *International Review of Electrical Engineering*, Vol. 8, No. 6, 1681–1693, 2013.

[2] Hsiao, C.-Y., S.-N. Yeh, and J.-C. Hwang, "Design of high performance permanent-magnet synchronous wind generators," *Energies*, Vol. 7, No. 11, 7105–7124, Nov. 2014.

[3] Xia, Y., K. H. Ahmed, and B. W. Williams, "A new maximum power point tracking technique for permanent magnet synchronous generator based wind energy conversion system," *IEEE Transactions on Power Electronics*, Vol. 26, No. 12, 3609–3620, Dec. 2011.

[4] Wu, Z., W. Gao, J. Wang, and S. Gu, "A coordinated primary frequency regulation from permanent magnet synchronous wind turbine generation," in *2012 IEEE Power Electronics and Machines in Wind Applications (PEMWA)*, 1–6, Denver, Co, Jul. 2012.

[5] Zhang, Y., Z. Yin, X. Sun, and Y. Zhong, "On-line identification methods of parameters for permanent magnet synchronous motors based on cascade MRAS," in *2015 9th International Conference on Power Electronics and ECCE Asia (ICPE-ECCE Asia)*, 345–350, Seoul, South Korea, Jun. 2015.

[6] Ma, X. and C. Bi, "A technology for online parameter identification of permanent magnet synchronous motor," *CES Transactions on Electrical Machines and Systems*, Vol. 4, No. 3, 237–242, 2020.

[7] Inoue, Y., Y. Kawaguchi, S. Morimoto, and M. Sanada, "Performance improvement of sensorless IPMSM drives in a low-speed region using online parameter identification," *IEEE Transactions on Industry Applications*, Vol. 47, No. 2, 798–804, Mar.–Apr. 2011.

[8] Wang, G., Y. Wang, J. Qi, R. Ni, W. Chen, and D. Xu, "Offline inductance identification of PMSM with adaptive inverter nonlinearity compensation," in *2015 9th International Conference on Power Electronics and Ecce Asia (ICPE-ECCE Asia)*, 2438–2444, Seoul, South Korea, Jun. 2015.

[9] Zhang, Q. and Y. Fan, "The online parameter identification method of permanent magnet synchronous machine under low-speed region considering the inverter nonlinearity," *Energies*, Vol. 15, No. 12, 4314, Jun. 2022.

[10] Deng, W. and S. Zuo, "Electromagnetic vibration and noise of the permanent-magnet synchronous motors for electric vehicles: An overview," *IEEE Transactions on Transportation Electrification*, Vol. 5, No. 1, 59–70, Mar. 2019.

[11] Tang, J., Y. Yang, F. Blaabjerg, J. Chen, L. Diao, and Z. Liu, "Parameter identification of inverter-fed induction motors: A review," *Energies*, Vol. 11, No. 9, 2194, Sep. 2018. [Online]. Available: https://doi.org/10.3390/en11092194.

[12] Zhang, Y., T. Y. Ji, M. S. Li, and Q. H. Wu, "Application of discrete wavelet transform for identification of induction motor stator inter-turn short circuit," in *2015 IEEE Innovative Smart Grid Technologies — Asia (ISGT ASIA)*, 1–5, Bangkok, Thailand, Nov. 2015.

[13] Sokolov, E. and M. Mihov, "Parameter estimation of an interior permanent magnet synchronous motor," in *2019 16th Conference on Electrical Machines, Drives and Power Systems (ELMA)*, Drives and Power Systems (ELMA), Varna, Jun. 2019.

[14] Boileau, T., 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, 1944–1957, Jul.–Aug. 2011.

[15] Nahid-Mobarakeh, B., F. Meibody-Tabar, and F. Sargos, "Mechanical sensorless control of PMSM with online estimation of stator resistance," *IEEE Transactions on Industry Applications*, Vol. 40, No. 2, 457–471, Mar.–Apr. 2004.

[16] Sun, P., Q. Ge, B. Zhang, and X. Wang, "Sensorless control technique of PMSM based on RLS on-line parameter identification," in *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*, 1670–1673, Jeju, South Korea, Oct. 2018.

[17] Li, M., K. Lv, C. Wen, Q. Zhao, X. Zhao, and X. Wang, "Sensorless control of permanent magnet synchronous linear motor based on sliding mode variable structure MRAS flux observation," *Progress In Electromagnetics Research Letters*, Vol. 101, 89–97, 2021.

[18] Ouyang, Y. and Y. Dou, "Speed sensorless control of PMSM based on MRAS parameter identification," in *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*, 1618–1622, Jeju, South Korea, Oct. 2018.

[19] Jiang, X., P. Sun, and Z. Q. Zhu, "Modeling and simulation of parameter identification for PMSM based on EKF," *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*, 345–348, 2010.

[20] Xiao, Q., K. Liao, C. Shi, and Y. Zhang, "Parameter identification of direct-drive permanent magnet synchronous generator based on EDMPSO-EKF," *IET Renewable Power Generation*, Vol. 16, No. 5, 1073–1086, Apr. 2022.

[21] Fortes, M. Z., V. H. Ferreira, and A. P. F. Coelho, "The induction motor parameter estimation using genetic algorithm," *IEEE Latin America Transactions*, Vol. 11, No. 5, 1273–1278, Sep. 2013.

[22] Jafar-Zanjani, S., S. Inampudi, and H. Mosallaei, "Adaptive genetic algorithm for optical metasurfaces design," *Scientific Reports*, Vol. 8, 11040, Jul. 2018.

[23] Dong, Q.-R., T. Chen, S.-J. Gao, Y.-K. Liu, J.-Q. Zhang, and H. Wu, "Identification of opto-electronic fine tracking systems based on an improved differential evolution algorithm," *Chinese Optics*, Vol. 13, No. 6, 1314–1323, Dec. 2020.