Progress In Electromagnetics Research M, Vol. 113, 59–71, 2022

Parameter Identification Based on Chaotic Map Simulated Annealing Genetic Algorithm for PMSWG

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

Abstract—Traditional genetic algorithm identification of permanent magnet synchronous wind generator (PMSWG) parameters is easy to fall into local optimum, resulting in low accuracy of parameter identification results and slow convergence which reduces the accuracy of parameter tuning of proportional-integral (PI) controller. Aiming at this problem, a chaotic mapping simulated annealing genetic algorithm (CMSAGA) for identifying PMSWG parameters is proposed. The traditional genetic algorithm (GA) has the ability of global random search, combined with the probability breakthrough characteristic of the simulated annealing (SA) algorithm, which avoids the parameter identification result falling into the local optimum and finally tends to the global optimum. With the increase of iteration times, the initial population is mapped with tent chaos mapping theory, and the optimal value of the population is disturbed in each iteration to increase the diversity of the population, making the proposed algorithm converge faster and improve the accuracy. Experiments show that the proposed algorithm has good accuracy and convergence speed, and PMSWG stator resistance, stator winding *d*-*q* axis inductance and permanent magnet flux can be identified.

1. INTRODUCTION

Permanent magnet synchronous wind generator (PMSWG) has the advantages of small size, simple structure, high operating efficiency, and high-power density [1–4], which is widely used in the field of wind power generation. To achieve high-performance permanent magnet synchronous wind turbine drive control, it is necessary to accurately obtain parameters such as motor stator winding resistance and *dq* axis inductance which can realize the setting calculation of proportional-integral controller parameters, and accurate acquisition of *dq*-axis inductance and flux linkage is beneficial for improving the decoupling effect of motor control. Due to factors such as demagnetization, high temperature, and magnetic saturation, the parameters are mismatched, which reduces the control performance and reliability of the motor. For such issues, scholars have done a lot of research and proposed different parameter identification methods [5].

The traditional parameter identification method, in [6] and [7], proposed an identification method based on a model reference adaptive system (MRAS). Although it has a fast convergence speed and is relatively easy to implement, it faces the problem of adaptive law selection, which has a great impact on the results of parameter identification. The correct convergence speed of parameters depends on the initial values of the parameters. Ref. [8] proposes an identification method based on the recursive least square method. Although it has high identification accuracy, it has poor robustness to noise and needs to process a large amount of data, so the design of the signal processing system is complicated, and the parameters cannot be identified online in real time. In [9–11], an identification method based on Extended Kalman Filter (EKF) is proposed, which is a recursive filtering method. Although it improves

Received 1 July 2022, Accepted 19 August 2022, Scheduled 9 September 2022

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

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

the accuracy of the system and solves noise-sensitive problems, it requires a lot of matrix calculations during the calculation process, resulting in long convergence time.

Compared with traditional identification methods, artificial intelligence algorithm greatly improves computational performance and identification accuracy, which is often used in the process of feature data. In [12] and [13], an identification method based on a neural network (NN) is proposed. Although it can reduce the steady-state error of the identification result and improve the convergence speed, the speed of the algorithm depends on the selection of the appropriate convergence factor, and inappropriate selection of convergence factor will lead to slow convergence. In [14], a Cauchy Mutation Particle Swarm Optimization (CMPSO) for identifying parameters is proposed, which mutates the particles so that the particle swarm deviates from the current position and searches for more information. This method is too cumbersome, slow in convergence, and complicated in the process. In [15], a coevolutionary particles warm optimization (PSO) algorithm associated with the artificial immune principle is proposed, introducing the immune system mechanism into the particle swarm, although it has better convergence speed and local fine search capability, and the computation process becomes lengthy and computationally intensive. In [16], the application of GA to the parameter identification of permanent magnet synchronous motor has improved the robustness, but it still has the defects of cumbersome calculation and poor accuracy. The SA starts from a certain initial temperature, accompanied by decreasing temperature parameters, and combines the probabilistic breakthrough property to find the global optimal solution of the objective function randomly in the solution space. In [17], SAGA is used for the identification of friction parameters, but it takes a longer time to anneal all populations.

Considering the problems of GA and SAGA, a chaotic mapping simulated annealing genetic algorithm (CMSAGA) is proposed in this paper to obtain better performance for PMSWG parameter identification. The work of our paper is summarized as follows:

(1) In order to avoid rank-deficient in the mathematical model, vector control method of $i_d = 0$ will be used, and a d-axis current of $i_d = -2$ is ejected to construct rank-full in the mathematical model.

(2) In order to improve the convergence speed, reduce the initial annealing temperature of SAGA, and then perform tempering and chaos disturbance operations on the inferior population to ensure the accuracy of the algorithm and improve the search accuracy.

(3) This paper adopts the method of resetting the number of iterations to make the termination condition of the algorithm change with the change of system running time so that each sampling point can be identified and meet the requirements of online identification.

(4) The simulation results and RT-LAB hardware-in-the-loop simulation results show that the four parameters of stator resistance, stator winding d-q axis inductance, and rotor flux can be quickly and accurately identified by the chaotic mapping simulated annealing genetic algorithm (CMSAGA).

The rest of this paper is as follows. Section 2 introduces the mathematical model of PMSWG. Section 3 introduces the traditional genetic algorithm and improved genetic algorithm. Section 4 introduces the identification principle of the chaotic mapping simulated annealing genetic algorithm (CMSAGA). Sections 5 and 6 show the simulated and experimental results. Finally, Section 7 briefly summarizes this paper.

2. PMSWG MODEL

Due to the complex control of the motor, the system model of the PMSWG is simplified; therefore, the equation of IPMSM in the d-q coordinate system is expressed as

$$\begin{cases} u_d = R_s i_d + \dfrac{d\varphi_d}{dt} - \omega_e \varphi_q \\ u_q = R_s i_q + \dfrac{d\varphi_q}{dt} + \omega_e \varphi_d \end{cases} \tag{1}$$

The flux linkage equation is

$$\begin{cases} \varphi_d = \varphi_f + L_d i_d \\ \varphi_q = L_q i_q \end{cases} \tag{2}$$

where u_d, u_q, i_d, i_q are the voltages and currents of the d-axis and q-axis; R_s, L_d, L_q, φ_f are stator resistance, d-q axis inductance, and permanent magnet flux linkage; ω_e is the electrical angular velocity; φ_d, φ_q are flux components on d-axis and q-axis, respectively.

When the motor is in steady state operation, the perturbations in i_d and i_q are very small and are regarded as (3)

$$\begin{cases} u_d = R_s i_d - \omega_e L_q i_q \\ u_q = R_s i_q + \omega_e (\varphi_f + L_d i_d) \end{cases} \tag{3}$$

In order to avoid rank-deficiency in the mathematical model, vector control method of $i_d = 0$ will be used, and a d-axis current of $i_d = -2$ is injected to construct rank-full in the mathematical model. The data sampling diagram is shown in Fig. 1.

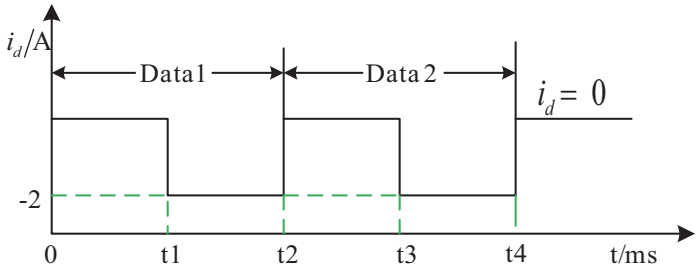


Figure 1. Data sampling diagram.

The same amount of data is collected at $i_d = 0$ and $i_d = -2$, and the discrete equation for the 4th-order full-rank reference model is expressed as (4)

$$\begin{cases} u_{d0}(k) = -\omega_e(k) L_q i_{q0}(k) \\ u_{q0}(k) = R_s i_{q0}(k) + \omega_e(k) \varphi_f \\ u_{d1}(k) = R_s i_{d1}(k) - \omega_e(k) L_q i_{q1}(k) \\ u_{q1}(k) = R_s i_{q1}(k) + \omega_e(k) (\varphi_f + L_d i_{d1}(k)) \end{cases} \tag{4}$$

where $u_{d_0}(k)$, $u_{q0}(k)$, $i_{q0}(k)$ are the data collected for the k-th time in time 0 to $t1$ in Fig. 1, and $u_{d1}(k)$, $u_{q1}(k)$, $i_{d1}(k)$, $i_{q1}(k)$ are the data collected for the k-th time in time $t1$ to $t2$.

3. CHAOTIC MAP SIMULATED ANNEALING GENETIC ALGORITHM

3.1. Basic Genetic Algorithm

GA is a method of simulating Darwin's genetic selection and the biological evolution process of natural elimination to simulate the search for optimal solutions. The basic process of GA: the initial population is generated randomly, then through the fitness function to evaluate each individual, with higher fitness value of individual participation in the genetic operation, low fitness of individuals is eliminated. The individual sets to form a new generation of population genetic operation until meet the stop criterion will be the best individual offspring as the results of GA [18].

The core parts of GA are selection, crossover and mutation.

(1) Selection

The core idea of the roulette selection method is that the probability of each individual being selected is proportional to its fitness value; therefore, the selection probability p_i of individual i is expressed as

$$F_i = 1/f_i \tag{5}$$

$$p_i = \frac{F_i}{\sum_{j=1}^{n} F_j} \tag{6}$$

where f_i is the fitness value of individual i; because smaller fitness values are better, the reciprocal of the fitness value is taken before individual selection; n is the number of individuals in the population.

(2) Crossover

In this paper, the method of crossover operation is real number coding, and the crossover operation method of the k-th chromosome α_k and the l-th chromosome at the j position is expressed as

$$\begin{cases} \alpha_{kj} = \alpha_{kj}(1-\beta) + \alpha_{lj}\beta \\ \alpha_{lj} = \alpha_{lj}(1-\beta) + \alpha_{kj}\beta \end{cases} \tag{7}$$

where β is a random number between $[0, 1]$.

(3) Mutation

The j-th gene of the i-th individual is selected for mutation, and the mutation operation is expressed as

$$\alpha_{ij} = \begin{cases} \alpha_{ij} - (\alpha_{ij} - \alpha_{\min}) * (1 - f(g)), & r \le 0.5 \\ \alpha_{ij} + (\alpha_{\max} - \alpha_{ij}) * (1 - f(g)), & r > 0.5 \end{cases} \tag{8}$$

where $\alpha_{\max}$, $\alpha_{\min}$ are upper and lower bounds for genes α_{ij}; $f(g) = r_(1 - g/g_{\max})^2$; g is the current iteration number; $g_{\max}$ is the maximum number of evolutions; r is a random number.

3.2. Simulated Annealing Genetic Algorithm

SA algorithm comes from the principle of solid annealing, which is a probabilistic algorithm. The SA algorithm starts from a certain high initial temperature, along with the decrease of the temperature parameter, combined with the probability breakthrough characteristic, which finds the global optimal solution of the objective function in the solution space randomly, i.e., the local optimal solution can jump out probabilistically and eventually tend to the global optimum.

The steps of the SA algorithm:

(1) Given an initial temperature T_0, an initial population x_0 is randomly generated. The corresponding objective function value $f(x_0)$ of length of Markov chain l (the number of iterations at any temperature) is calculated.

(2) Make the current temperature T equal to the next value T_i in the cooling schedule.

(3) Generate a new population x_j near the current population x_i randomly, and calculate the objective function value $f(x_j)$ of the new population.

(4) According to the Metropolis criterion, $\Delta f = f(x_j) - f(x_i)$ if $\Delta f < 0$, accept the new solution x_j if $\Delta f > 0$, calculate $p = e^{-\Delta f/T_i}$, then generate a random number r on $[0, 1]$, if $r < p$, accept the new solution x_j.

(5) At temperature T_i, repeat steps (3) and (4) l_i times.

(6) Judge whether the exit conditions are met. If so, exit the iteration, otherwise, go back to Step (2) to continue the iteration.

Metropolis criterion — accept a new state with probability p which is expressed as

$$p_l(i \to j) = \begin{cases} 1, & f(i) \ge f(j) \\ \exp\left(\dfrac{f(i) - f(j)}{T}\right), & f(i) < f(j) \end{cases} \tag{9}$$

In order to solve the problem that the GA is easy to fall into the local optimum, and the SA algorithm has poor convergence speed, the annealing process is integrated into the GA, so that the simulated annealing genetic algorithm (SAGA) can converge to the global optimum quickly.

During the search, the genetic operator provides a set of initial solutions for the annealing process at each temperature, which accepts each solution by Metropolis criterion until the equilibrium condition is reached. The genetic operation continues to optimize in parallel using the solution found through the annealing process. As the temperature decreases, the new solution accepted by the annealing process between neighbors tends to decrease as the temperature decreases, and the iteration continues until the end condition is satisfied [19]. The steps of the SAGA:

(1) Initial population P is randomly generated. Set the initial temperature T_0, population number n, crossover rate p_c, mutation rate p_m, temperature attenuation coefficient α, length l of Markov chain, and build fitness function fitness [20].

(2) Select the initial population.

(3) Cross operation and simulated annealing operation, P_1, P_2 crossover operation to generate offspring c_1, c_2, calculate the fitness value $f(P_i)$, $f(c_i)$ of P_i, c_i, $i = 1, 2$. If $f(c_i) < f(P_i)$, replace c_i with P_i. If $f(c_i) > f(P_i)$, calculate the probability $p = \exp(-(f(c_i) - f(P_i))/T)$, generate a random number r on $0, 1]$, and if $r < p$, accept c_i.

(4) Mutation operation and SA operation are carried out in the same way as step (3).

(5) Iterate through the cooling formula $T = T_0 \cdot \alpha^l$, where l is the number of iterations.

(6) By judging the termination conditions of the algorithm, if the termination conditions are met, the optimal solution is output, and the algorithm ends, otherwise, go to step (2) to continue the iteration.

3.3. Chaotic Map Simulated Annealing Genetic Algorithm

It is proved that Tent map can be used as a chaotic sequence to generate optimization algorithm; therefore, tent chaos mapping adopts r to increase the initial population diversity of SAGA in this paper, and at the same time, chaotic perturbation is performed on the optimal population value during each iteration Chaotic sequence which is expressed as

$$t_{n+1} = \begin{cases} v * t_n, & t_n \leq 0.5 \\ v * (1 - t_n), & t_n > 0.5 \end{cases} \tag{10}$$

where $t_n, t_{n+1} \in [0, 1]$, v is a random number of $(1, 2)$.

In order to increase the initial population diversity of the SAGA, use tent chaotic mapping to map its initial population. Put the chaotic sequence carrier generated by Equation (10) into the space to be solved, as shown in Equation (11).

$$x_{Dim,New} = (x_{\max} - x_{\min}) * (t - 0.5) + (x_{\max} + x_{\min}) * 0.5 \tag{11}$$

where dim is the dimension of the chaotic sequence; $x_{Dim,New}$ is the new value of chaotic sequence carrier in the space to be solved; $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of variables in the sequence respectively.

The new value obtained by tent chaos mapping is used as the initial population distribution of SAGA, which increases the diversity of population and improves the global searching ability of SAGA. In order to further enrich the diversity of SAGA population, tent chaos perturbation is performed on the optimal value of SAGA population after each SAGA iteration. The excellent individuals in the population are disturbed by formula (12).

$$x_{New} = (x + x_{Dim,New})/2 \tag{12}$$

where x is the individual that needs to be disturbed, and x_{New} is the new position of the excellent individual after disturbance.

Judging the new position obtained after the disturbance, if it is better than the previous position, update the original position with the new position, otherwise keep the original position unchanged.

4. IDENTIFICATION PRINCIPLE OF CMSAGA

The identification process of PMSWG is based on the difference between the output of its reference model and the output of the adjustable model, and the parameters to be identified of the adjustable model are modified according to the fitness function by an intelligent optimization algorithm to obtain the parameters of PMSWG. Motor reference model is expressed as

$$\begin{cases} \dot{x} = f(\theta, x, u) \\ y = Cx \end{cases} \tag{13}$$

where x is the state vector, u the input vector, θ the parameter vector to be identified, y a measurable vector, C a constant matrix of suitable order. Motor adjustable model can be expressed as Equation (14), and the variables in Equation (14) are similar to those in Equation (13).

$$\begin{cases} \hat{\dot{x}} = f\left(\hat{\theta}, \hat{x}, u\right) \\ \hat{y} = C\hat{x} \end{cases} \tag{14}$$

where $\hat{x}$ is the state vector of a motor adjustable model, $\hat{\theta}$ is an estimate of θ, $\hat{y}$ is an estimate of y.

In order to identify the parameters, the fitness function is used to compare the reference model with the adjustable model, which is expressed as

$$H(\theta) = (y - \hat{y})^T (y - \hat{y}) \tag{15}$$

The first-order forward Euler method is used for discretization which is expressed as

$$\begin{bmatrix} \hat{i}_d(k+1) \\ \hat{i}_q(k+1) \end{bmatrix} = \begin{bmatrix} 1 - \dfrac{\hat{R}_s(k+1)T_s}{\hat{L}_d(k)} & \omega_e(k)\dfrac{\hat{L}_q(k)T_s}{\hat{L}_d(k)} \\ -\omega_e(k)\dfrac{\hat{L}_d(k)T_s}{\hat{L}_q(k)} & 1 - \dfrac{\hat{R}_s(k)T_s}{\hat{L}_q(k)} \end{bmatrix} \begin{bmatrix} \hat{i}_d(k) \\ \hat{i}_q(k) \end{bmatrix} + \begin{bmatrix} \dfrac{T_s}{\hat{L}_d(k)} & 0 \\ 0 & \dfrac{T_s}{\hat{L}_q(k)} \end{bmatrix} \begin{bmatrix} u_d(k) \\ u_q(k) - \omega_e(k)\hat{\varphi}_f(k) \end{bmatrix} \tag{16}$$

According to (15), the fitness function is expressed as

$$f(R_s, L_d, L_q, \varphi_f) = \left(i_d(k) - \hat{i}_d(k)\right)^2 + \left(i_q(k) - \hat{i}_q(k)\right)^2 \tag{17}$$

The inputs of the reference model and the adjustable model are u_d, u_q, i_d, i_q, ω_e, and the outputs of the two are compared by fitness function. CMSAGA modifies the identified parameters through the value of fitness function, and the modified parameters replace the parameters of the reference model and repeat the process until the error between the reference model and the adjustable model output is minimal. The principle of parameter estimation is shown in Fig. 2.

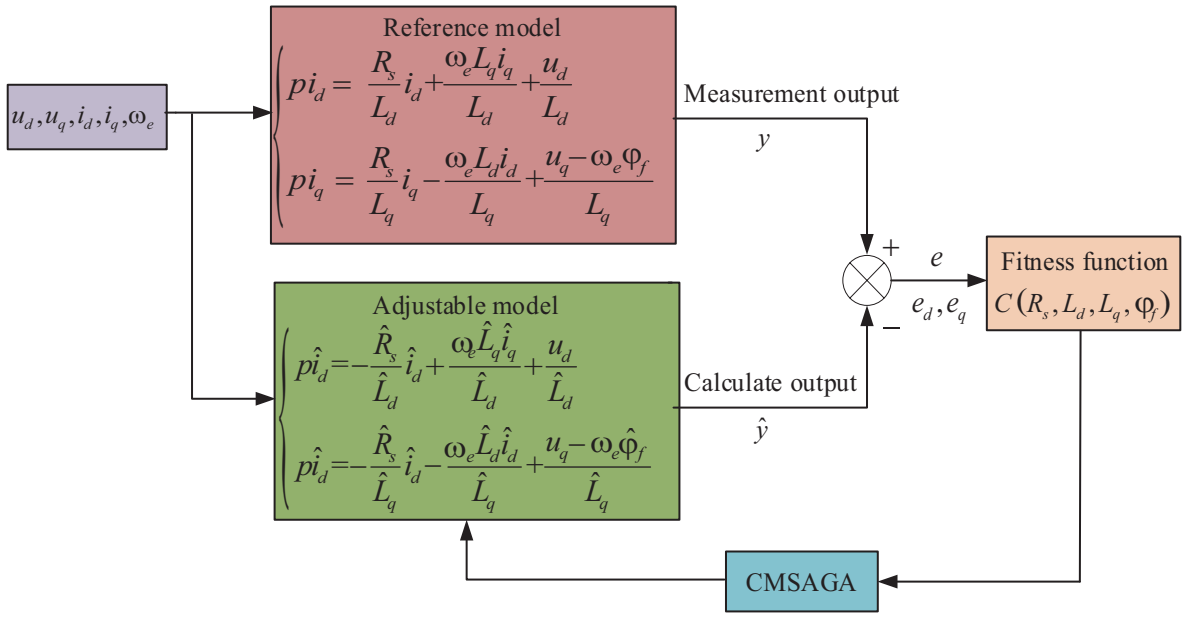


Figure 2. Block diagram of parameter identification.

(1) Collect and save electrical signals under $i_d = 0$ and $i_d = -2A$, including u_d, u_q, i_d, i_q, ω_e.

(2) The range of parameters to be identified was set, and the population and related parameters were initialized. Given the values of crossover rate and mutation rate, the maximum number of iterations was set as 1000.

(3) Calculate the fitness value of individual through the mathematical model of the motor, refresh the optimal individual of the population and the optimal individual constantly, and each individual generates the next generation of individuals through CMSAGA, so that the fitness value of the next generation of individuals is smaller than the previous generation. Judge whether the maximum number of iterations is reached. If yes, identify the parameter output and end. If no, go ahead.

Figure 3 is the schematic diagram of CMSAGA.

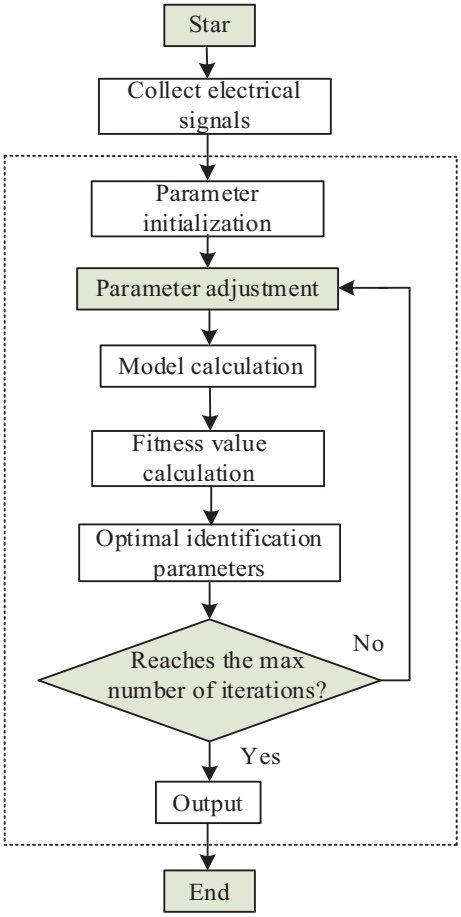


Figure 3. Flow chart of PMSWG parameter identification by CMSAGA.

5. SIMULINK SYSTEM SIMULATION

In order to verify the identification effect of CMSAGA, in the Matlab/Simulink environment, the vector control system simulation model in the synchronous rotation coordinate system was built, as shown in Fig. 4.

The PMSWG parameters used for the simulation are shown in Table 1.

Table 1. PMSWG parameter table.

Parameter	Value	Unit
Pole pairs	4	pairs
Resistance	0.933	Ω
Stator d-axis inductance	5.2	mH
Stator q-axis inductance	11.5	mH
Permanent magnet flux	0.175	Wb
Moment of inertia	0.003	kg·m^2
Rated power	1.0	kW
Rated speed	100	rpm
Rated torque	1	N·m

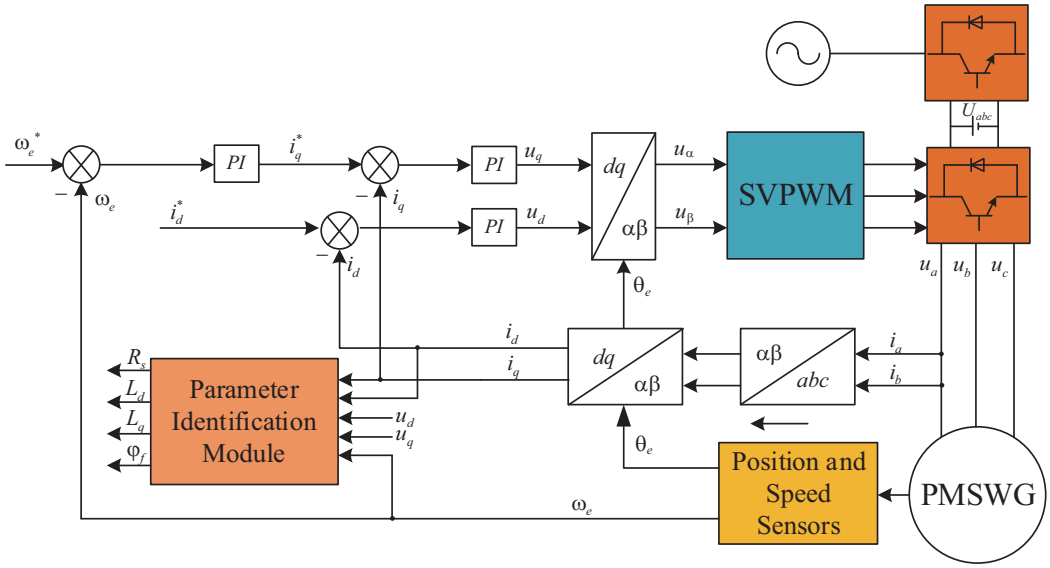


Figure 4. Simulation model of vector control system.

During simulation, rated speed is 1000 r/min; rated torque is 10 N · m; in the vector control strategy, the data is collected in the two states of $id = 0$ and $id = -2A$; a total of 1000 sets of 40 ms data were collected and saved; sampling time is 1e-6 s. In order to ensure the comparison accuracy of the three algorithms, all initial parameters are set to the same data; population is 20; the maximum number of iterations is 1000; the crossover rate is 0.7; and the mutation rate is 0.1. In order to detect the global optimization ability of the algorithm, the value range of the four parameters of motor identification is between $[0, 2]$, away from the reference value of the motor.

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

Table 2. Comparison of three algorithms simulation.

Parameter	GA	SAGA	CMSAGA
Resistance/Ω	0.972	0.962	0.941
Error/%	4.2	3.1	0.9
Ld/mH	5.63	4.96	5.26
Error/%	8.2	4.6	1.2
Lq/mH	12.3	11.86	11.41
Error/%	6.7	3.1	.8
Flux/Wb	0.182	0.171	0.173
Error/%	4.1	2.2	1.2
Fitness value	2.1	1.2	0.8
Recognition time/ms	32	3	11

6. EXPERIMENTAL VERIFICATION

In order to verify the feasibility of parameter identification of CMSAGA, the Simulink simulation model was downloaded to RT-Lab to achieve the hardware-in-the-loop simulation experiment of PMSWG drive system. The RT-LAB hardware-in-the-loop system configuration diagram of PMSWG is shown in Fig. 5. The RT-LAB experiment platform is shown in Fig. 6.

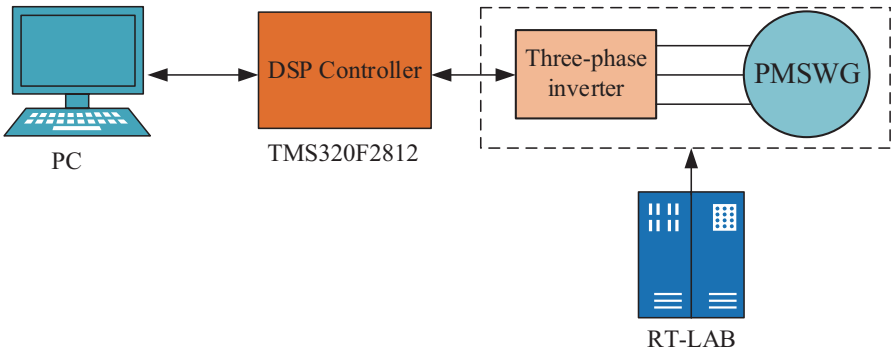


Figure 5. RT-LAB hardware-in-the-loop system.

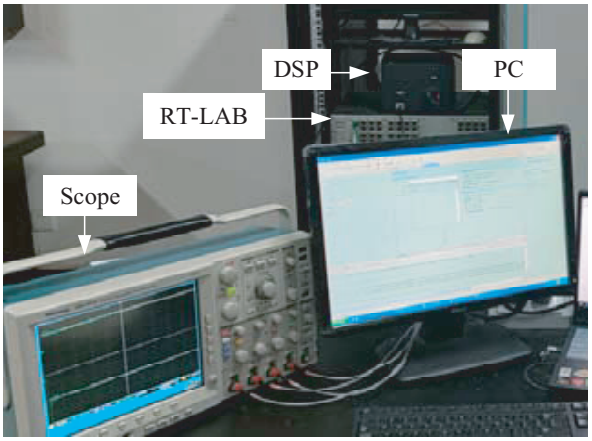
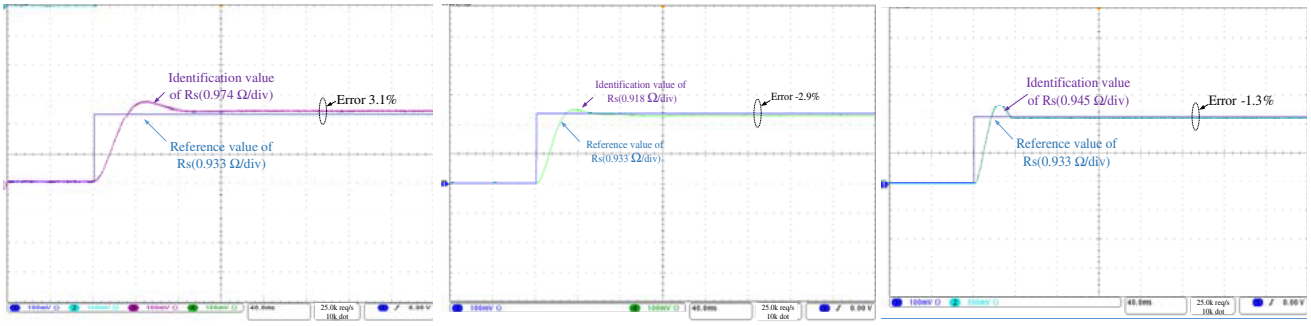


Figure 6. RT-LAB experimental platform.

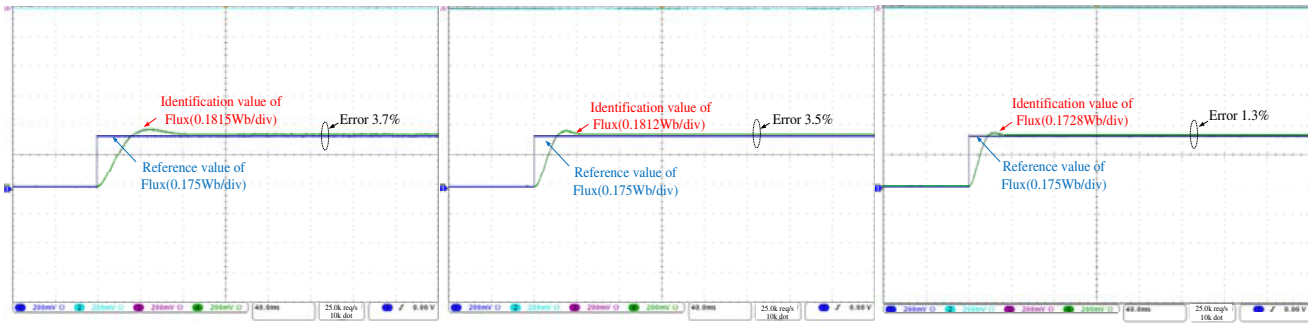


(a) (b) (c)

Figure 7. Identification curve of stator resistance. (a) GA. (b) SAGA. (c) CMSAGA.

When the speed is 1000 r/min and the load torque 10 N · m, the identification results of the three methods are shown in Figs. 7–10. Because the values of d-axis inductance, q-axis inductance, and permanent magnet flux are small, in order to better observe the identification results, the d-axis inductance, q-axis inductance, and permanent magnet flux are amplified by 1000, 500, and 20 times respectively, and the simulation time is 0.5 s.

Figures 7–8 show the identification results of the three algorithms of stator resistance and permanent magnet flux, respectively. In stator resistance identification, the identification errors of GA, SAGA, and CMSAGA are 3.1%, 2.9%, and 1.3%, respectively, and the identification accuracies of GA and

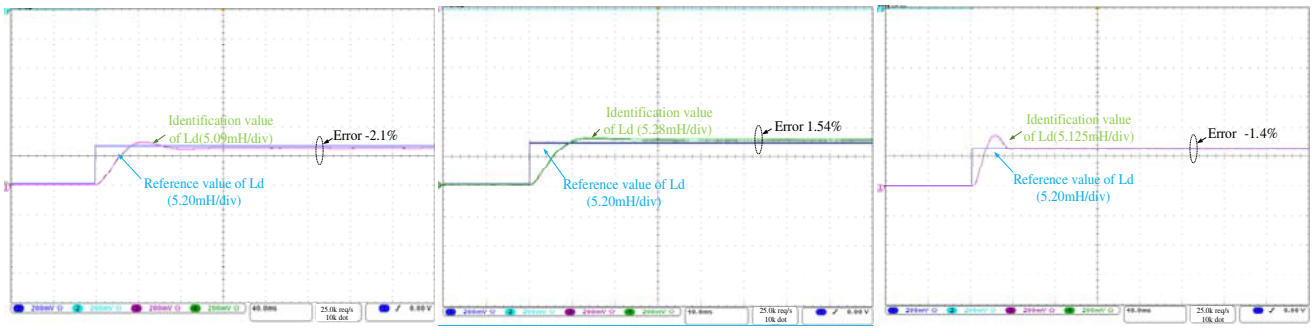


(a) (b) (c)

Figure 8. Identification curve of permanent magnet flux. (a) GA. (b) SAGA. (c) CMSAGA.

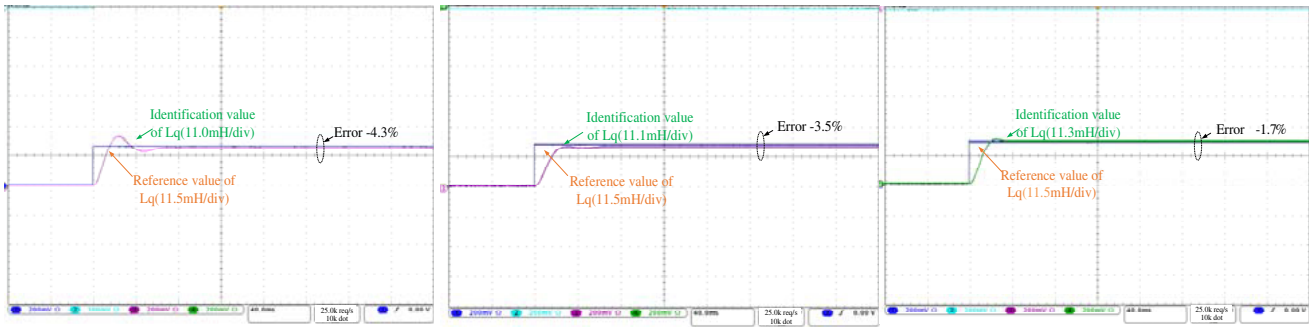
SAGA are close, but the convergence speed of SAGA is faster than GA. In permanent magnet flux, the identification errors of GA, SAGA, and CMSAGA are 3.7%, 3.5%, and 1.3%, respectively, and the identification accuracies of GA and SAGA are close.

Figures 9–10 show the identification results of three algorithms for d-axis inductance and q-axis inductance, respectively. In the d-axis inductance identification, the identification errors of GA, SAGA, and CMSAGA are 2.1%, 1.54%, and 1.4%, respectively, and the identification accuracies of SAGA and CMSAGA are almost the same. In the q-axis inductance, the identification errors of GA, SAGA, and CMSAGA are 4.3%, 3.5%, and 1.7%, respectively, and the identification accuracy of CMSAGA is 51.4% and 60.5% higher than that of SAGA and GA, respectively.



(a) (b) (c)

Figure 9. Identification curve of Ld. (a) GA. (b) SAGA. (c) CMSAGA.



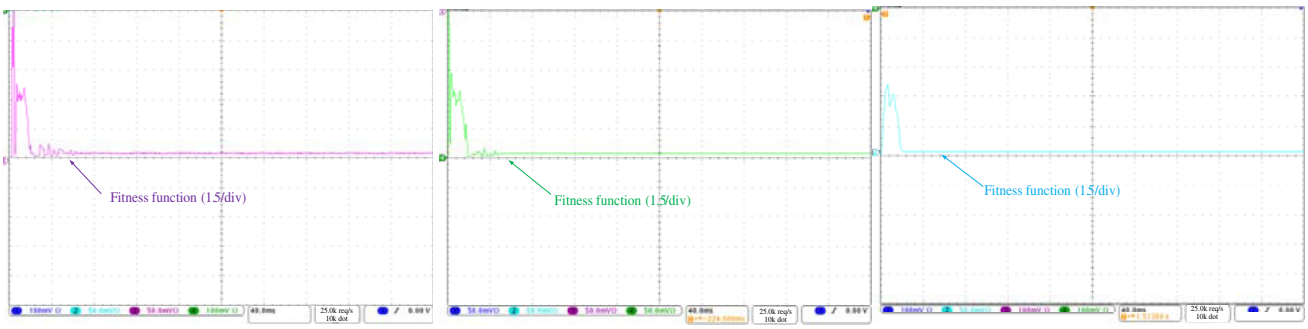
(a) (b) (c)

Figure 10. Identification curve of Lq. (a) GA. (b) SAGA. (c) CMSAGA.

GA completed the parameter identification of the motor at 67 ms. SAGA completed motor parameter identification at 52 ms. CMSAGA completed the motor parameter identification at 34 ms. The convergence rate of CMSAGA is nearly twice that of GA and 53% faster than that of SAGA.

The identification accuracy and convergence speed of GA are poor, and the identification accuracies of SAGA and GA in identifying stator resistance and permanent magnet flux are close. However, the convergence speed of SAGA is faster than that of GA; the accuracy of parameter identification of CMSAGA is higher; and the convergence speed is the fastest.

Figure 11 is the fitness function curves of the three algorithms. The fitness value of CMSAGA converges to 1.8 at 34 ms, while SAGA needs 52 ms. GA converges to 2.9 at 64 ms due to local optimization and fluctuates greatly in the early experiment.



(a) (b) (c)

Figure 11. Fitness function value curve. (a) GA. (b) SAGA. (c) CMSAGA.

The main reason that the identification speed and accuracy of the experiment have greater errors than the simulation results is that the experimental noise, electromagnetic interference, hardware circuit connection, and signal transmission errors will have a certain impact on the accuracy and convergence speed of the parameter identification.

7. CONCLUSION

GA has poor accuracy, and SAGA has slow convergence speed. A motor parameter identification method based on CMSAGA is proposed, which adds the randomness of SA and overcomes the defect that GA is easy to fall into local optimum. In order to improve the convergence speed of the algorithm, the initial SAGA annealing temperature was reduced, and then the inferior populations were tempered and chaotic to ensure the accuracy of the algorithm. Conclusions from the experiments:

(1) The termination condition of the algorithm changes with the running time of the system, and the motor stator resistance, d axis inductance, q axis inductance, and permanent magnet flux can be identified online.

(2) The error of the proposed motor parameter identification method is less than 2%, and the convergence speed is 34 ms. The parameter identification accuracy and convergence speed are better than traditional GA and SAGA.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant Number 51907061, Educational Commission of Hunan Province of China under Grant Number 21B0552, Natural Science Foundation of Hunan Province of China under Grant Number 2022JJ50094.

REFERENCES

1. Liu, X., Y. Pan, L. Wang, et al., "Model predictive control of permanent magnet synchronous motor based on parameter identification and dead time compensation," *Progress In Electromagnetics Research C*, Vol. 120, 253–263, 2022.
2. Wen, D., C. Shi, K. Liao, et al., "Fast backfire double annealing particle swarm optimization algorithm for parameter identification of permanent magnet synchronous motor," *Progress In Electromagnetics Research M*, Vol. 104, 23–38, 2021.
3. Liu, X., Y. Pan, Y. Zhu, H. Han, and L. Ji, "Decoupling control of permanent magnet synchronous motor based on parameter identification of fuzzy least square method," *Progress In Electromagnetics Research M*, Vol. 103, 49–60, 2021.
4. Zhu, L., B. Xu, and H. Zhu, "Interior permanent magnet synchronous motor dead-time compensation combined with extended Kalman and neural network bandpass filter," *Progress In Electromagnetics Research M*, Vol. 98, 193–203, 2020.
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," *2015 9th International Conference on Power Electronics and ECCE Asia (ICPE-ECCE Asia)*, 345–350, 2015.
6. Li, M., K. Lv, C. Wen, et al., "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.
7. Ouyang, Y. and Y. Dou, "Speed sensorless control of PMSM based on MRAS parameter identification," *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*, 1618–1622, IEEE, 2018.
8. Sun, P., Q. Ge, B. Zhang, et al., "Sensorless control technique of PMSM based on RLS on-line parameter identification," *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*, 1670–1673, IEEE, 2018.
9. 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.
10. Xiao, Q., K. Liao, C. Shi, et al., "Parameter identification of direct-drive permanent magnet synchronous generator based on EDMPSO-EKF," *IET Renewable Power Generation*, Vol. 16, No. 5, 1073–1086, 2022.
11. Sel, A., B. Sel, U. Coskun, et al., "Comparative study of an EKF-based parameter estimation and a nonlinear optimization-based estimation on PMSM system identification," *Energies*, Vol. 14, No. 19, 610, 2021.
12. Hussain, S. and M. A. Bazaz, "Sensorless control of PMSM drive using Neural Network Observer," *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, 1–5, IEEE, 2016.
13. Wang, S., G. Yang, Z.-J. Qu, et al., "Identification of PMSM based on EKF and elman neural network," *2009 IEEE International Conference on Automation and Logistics*, 1459–1463, IEEE, 2009.
14. Zou, Y., P. X. Liu, C. Yang, et al., "Collision detection for virtual environment using particle swarm optimization with adaptive cauchy mutation," *Cluster Computing*, Vol. 20, No. 2, 1765–1774, 2017.
15. Liu, Z., J. Zhang, S. Zhou, X. Li, and K. Liu, "Coevolutionary particle swarm optimization using AIS and its application in multiparameter estimation of PMSM," *IEEE Transactions on Cybernetics*, Vol. 43, No. 6, 1921–1935, Dec. 2013.
16. Avdeev, A. and O. Osipov, "PMSM identification using genetic algorithm," *2019 26th International Workshop on Electric Drives: Improvement in Efficiency of Electric Drives (IWED)*, 1–4, 2019.
17. Guo, H., B. Zhou, P. Yang, and X. Gu, "Application of modified Stribeck model and simulated annealing genetic algorithm in friction parameter identification," *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, 1–5, 2017.

18. Kumar, M., D. Husain, N. Upreti, et al., "Genetic algorithm: Review and application," Available at SSRN 3529843, 2010.
19. Zhang, D., W. Li, X. Wu, et al., "Application of simulated annealing genetic algorithm-optimized Back Propagation (BP) neural network in fault diagnosis," *International Journal of Modeling, Simulation, and Scientific Computing*, Vol. 10, No. 04, 1950024, 2019.
20. Guo, H., B. Zhou, P. Yang, and X. Gu, "Application of modified Stribeck model and simulated annealing genetic algorithm in friction parameter identification," *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, 1–5, 2017.