IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 May 2023, accepted 7 June 2023, date of publication 9 June 2023, date of current version 21 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3284971

THEORY

PID Controller for PMSM Speed Control Based on Improved Quantum Genetic Algorithm Optimization

HONGZHI WANG, SHUO XU, AND HUANGSHUI HU, (Member, IEEE)
College of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China

Corresponding author: Shuo Xu (1085200638@qq.com)

This work was supported in part by the Horizontal Project in Jilin Province under Grant 202-142254 and Grant 202-142255, in part by the Natural Science Foundation of Jilin Provincial Department under Grant 20210201051GX, and in part by the Jilin Provincial Department of Education under Grant JJKH20220686KJ.

ABSTRACT When traditional proportional integral and differential controllers are applied to speed control in permanent magnet synchronous motors (PMSM), their coefficients are basically determined based on experience, which inevitably leads to unsatisfactory results when using this parameter to control the speed stability of permanent magnet synchronous motors. Therefore, this paper proposes an improved quantum genetic algorithm using quantum states as the basic unit. Utilizing quantum properties for global optimization to optimize the coefficients of proportional integral and differential control, improving the rotation angle of quantum state particles through the idea of velocity changes in particle swarm optimization (PSO), introducing adaptive weight changes, using Hadamard gates to replace traditional algorithm mutation strategies, and incorporating disaster mechanisms. In addition, this paper uses four test functions to find the minimum value, thereby verifying that our algorithm has better performance in optimization iteration compared to other algorithms, providing the initial basis for the next step of application in PID parameter optimization. Prove that this method can solve the problem of traditional genetic algorithms falling into local optima due to improper selection, crossover, and mutation methods, which cannot effectively control the stability of motor speed. Finally, this paper uses Matlab2018a simulation to compare with the other four algorithms, and the results show that this algorithm can find better PID parameter values to achieve better results in motor oscillation, overshoot, and faster target speed.

INDEX TERMS Particle swarm optimization (PSO), permanent magnet synchronous motor (PMSM), proportional integral and differential (PID), quantum genetic algorithm (QGA).

I. INTRODUCTION

PMSM is widely used in electric vehicles [1], aerospace [2], industry [3], and other fields. Therefore, scholars at home and abroad have a strong interest in the stability control [4], [5] method of permanent magnet synchronous motor, such as using Sliding mode control to make the speed more stable. In [6], an improved Sliding mode control strategy for time-varying disturbance observer is proposed. The results show that compared with the traditional Sliding mode control, the overshoot is smaller, the transient response is faster, the control accuracy is higher, and the robustness is stronger. The model predictive current controller is a popular and effective technology that enables motors to respond faster to sudden changes, making motor current more stable [7]. PMSM is a typical nonlinear control system, traditional PI control is difficult to obtain satisfactory control performance. Therefore, some new control methods such as model predictive control (MPC) are proposed and applied to PMSM. MPC has faster dynamic performance than FOC but its steady-state performance is poor due to the lack of modulation units. However, many scholars have improved it. In [8], the amplitude of voltage vector is adjusted by adding zero vector; in [9], the phase and amplitude are adjusted by adding virtual

The associate editor coordinating the review of this manuscript and approving it for publication was Jinquan Xu.

vector on the basis of [8]. In addition, these options are outlined [10]. Use Kalman filtering algorithm to improve the anti-interference ability of PMSM or adaptively change motor parameters [11], [12].

In the speed control of PMSM, the use of a PID controller is a very common control method. PID is popular in other industrial control fields due to its simple structure, simple principle, and many other advantages. However, the determination of the three parameters of PID: proportional (Kp), integral (Ki), and differential (Kd) can generally only be determined by empirical trial and error. The selected parameters may not be the optimal solution, which will inevitably lead to problems such as poor stability and slow response speed in the control method. With the continuous development of control technology and computer technology, researchers are also deepening their research on PID parameter-setting methods. Although optimizing algorithms may increase the burden on computers, it is still worthwhile compared to spending more time accumulating experience and trial and error. In [13], they proposed a variable coefficient that is appropriately defined and optimized through parameter relationships, and a simplified fractional order PID (FOPID) controller to optimize the variable coefficient. The use of classical fuzzy control PID in [14] and [15] achieved good results.

Among various PID optimization algorithms, the use of population optimization algorithms is also receiving more and more attention. Most population optimization algorithms do not rely on whether the optimization system has accurate system equations, and various population algorithms become more accurate over time, constantly updating various parameters, and achieving satisfactory results in various fields. An ant lion optimizer (ALO) was used in [16] and its effectiveness was verified through simulation experiments However, ALO is also prone to falling into local optima and has a slower convergence speed. The genetic algorithm and particle swarm optimization algorithm are equally prominent. After years of use and improvement by scholars, these two algorithms are still applied in various time-varying systems and linear continuous systems [17], [18].

A genetic algorithm was used in [19] and [20] to obtain a set of PID parameters, ensuring the stability and robustness of the closed-loop system. However, traditional GA algorithms cannot perform global optimization well. In [21], they combined genetic algorithm and particle swarm optimization algorithm to optimize PID parameters and achieved satisfactory results. In [22], the particle swarm optimization algorithm was improved to adaptively change the weight of the algorithm to maintain system stability as much as possible. However, the improvement of the algorithm was too simple, and although the stability speed became faster, it was still difficult to jump out if trapped in local optima. In [23], adaptive weight changes were introduced while allowing the global optimal solution to mutate in a small range, reducing the dependence of all particles on the optimal particle, and achieving satisfactory indicators in all aspects.

The quantum genetic algorithm was first proposed in [24], which combines quantum computing with genetic algorithm and can express population diversity with a smaller number of populations. However, although the initial quantum genetic algorithm achieved satisfactory results in global optimization, and its rotation angle was fixed, it often missed the optimal value in the local optimization process, thereby missing the global optimal. Through comparative experiments in [25], it was demonstrated that the QGA algorithm has better development prospects compared to the GA algorithm. The combination of quantum algorithm and pigeon swarm algorithm in [26] proves the feasibility of combining quantum algorithm with other algorithms. The application of the QGA algorithm in various fields in [27], [28], and [29] has proven that the adaptability of the QGA algorithm is strong enough. In [30], the update method of the rotation angle of the quantum genetic algorithm was improved to change according to the maximum and minimum fitness values, and its superiority was verified by finding the minimum value of the binary function.

In summary, in order to ensure that genetic algorithm does not fall into local optima and can approach the global optimal solution as closely as possible, and improve the iteration efficiency of the algorithm, this paper introduces quantum optimization algorithm and combines the idea of PSO speed update to propose an Improved Quantum Genetic Algorithm (IQGA) to optimize PID control, and conducts comparative simulation experiments with other optimization algorithms in other papers.

II. MODEL OF PMSM SYSTEM

At present, the common methods of traditional vector control are $i_d = 0$ control and maximum torque current ratio control. The former is mainly suitable for surface-mounted three-phase PMSM, while the latter is mainly used for interior three-phase PMSM [31], [32], [33]. This paper selects surface-mounted three-phase PMSM, and the control block diagram is shown in Figure 1.

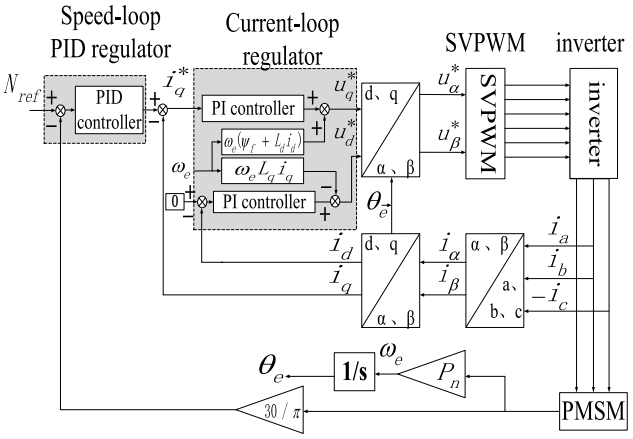

FIGURE 1. Block diagram of surface-mounted three-phase PMSM vector control.

The mathematical model of the three-phase PMSM selected in this paper is established on the synchronous rotating coordinate axis d-q [34], [35] and the stator voltage equation in the d-q coordinate system is (1): α

$$\begin{cases} u_d = Ri_d + L_s \dfrac{d}{dt} i_d - \omega_e L_s i_q \\ u_q = Ri_q + \dfrac{d}{dt} i_q + \omega_e (L_s i_d + \psi_f) \end{cases} \tag{1}$$

where u_d, u_q, i_d, i_q are the stator d-axis voltage, q-axis voltage, d-axis current, and q-axis current in the d-q coordinate of the synchronous rotation axis. R is the stator resistance, ω_e is the electrical angular velocity, ψ_f is the permanent magnet flux, and L_s is the inductance. Since this paper uses a surface-mounted PMSM, the d-axis and q-axis inductance components of the motor are equal, i.e.: $L_d = L_q = L_s$ [36].

To facilitate the parameter tuning of the speed-loop PID regulator, the motor motion equation of the three-phase PMSM is rewritten as:

$$J \frac{d\omega_m}{dt} = T_e - T_L - B\omega_m \tag{2}$$

$$T_e = \frac{3}{2} P_n i_q [i_d (L_d - L_q) + \psi_f] \tag{3}$$

In (2) and (3), ω_m is the mechanical angular velocity; J is the moment of inertia; B is the damping coefficient; T_L is the load torque; T_e is the electromagnetic torque; P_n is the pole pair. reference [37] proposed the concept of ''active damping'' to design the parameters of the speed-loop PID controller. He defined active damping as (4):

$$i_q = i_q^{\iota} - B_a \omega_m \tag{4}$$

This paper adopts a control strategy of $i_q^* = 0$ and assumes that the motor starts in an unloaded state, i.e. $T_L = 0$. From (2) to (4), (5) can be obtained:

$$\frac{d\omega_m}{dt} = \frac{1.5 p_n \psi_f}{J} (i_q^{\iota} - B_a \omega_m) - \frac{B}{J} \omega_m \tag{5}$$

Assign the poles of (5) to the desired closed-loop bandwidth β to obtain the transfer function of the rotational speed relative to the q-axis current as follows [38], [39]:

$$\omega_m(s) = \frac{1.5 p_n \psi_f / J}{s + \beta} i_q^{\iota}(s) \tag{6}$$

The coefficient B_a of active damping can be obtained from (5) and (6) as follows:

$$B_a = \frac{\beta J - B}{1.5 p_n \psi_f} \tag{7}$$

In (7), β is the expected bandwidth of the speed-loop.

The traditional PID controller is as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \tag{8}$$

The transfer function of the PID controller is as follows:

$$G(s) = K_p + K_i \frac{1}{s} + K_d s \tag{9}$$

Therefore, the expression for the speed-loop controller of traditional PID controllers is [40]:

$$i_q^* = (K_p + K_i \frac{1}{s} + K_d s) \times (\omega_m^* - \omega_m) - B_a \omega_m \tag{10}$$

The simulation model of the speed-loop PID regulator constructed by (10) is as follows

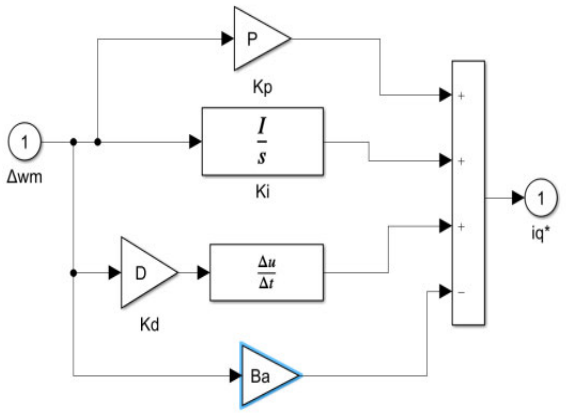

FIGURE 2. Speed-loop PID control model.

III. IMPROVED QUANTUM GENETIC ALGORITHM

A. TRADITIONAL GENETIC ALGORITHM

GA mainly includes the following steps:

1) Population initialization represents the problem that requires a solution as a chromosome or individual in the genetic space through encoding. Common encoding methods include real encoding, Grey encoding, multi-level parameter encoding, and so on.

2) Set the fitness function, which is used to distinguish the quality of evaluation chromosomes. This paper solves the minimum value of the fitness function.

3) When selecting chromosomes, the better the fitness value, the greater the probability that chromosomes will be selected.

4) Cross operation, randomly selecting two chromosomes from a population, and generating new chromosomes through chromosome exchange and combination.

5) Mutation operation: In order to maintain population diversity, a chromosome is randomly selected from the population and mutated at a certain point to generate a new chromosome. The mutated chromosome may not be better than the parent generation.

6) Iterative optimization: if the fitness value meets the expected value or the number of iterations reaches the maximum value, it will end; otherwise, go to step (2).

The flowchart of the GA is shown in Figure 3.

B. IMPROVED QUANTUM GENETIC ALGORITHM

QGA is a probability evolution algorithm that combines quantum computing with genetic algorithm. Traditional genetic algorithms adhere to the principle of survival of the

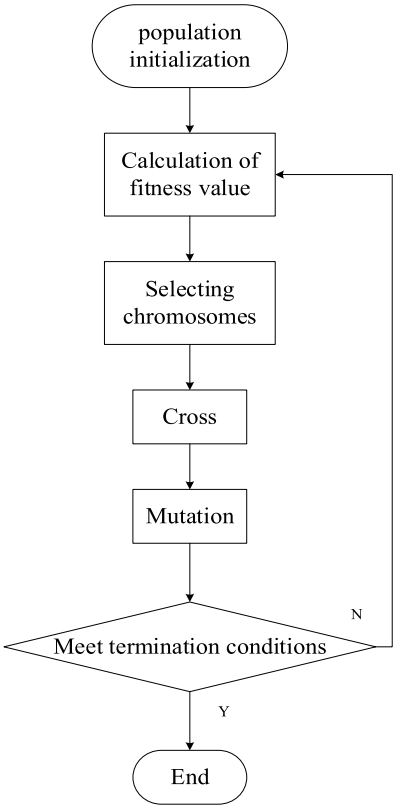

FIGURE 3. GA flowchart.

fittest in the biological world when dealing with optimization problems to find the optimal individual. They are not affected by the natural factors of the optimization problem and have strong applicability. However, if the selection, crossover, and mutation methods are not appropriate, GA needs to increase the number of iterations and has a slow convergence speed, which can easily fall into the local extremum.

QGA is a kind of GA based on the principle of quantum computing. By replacing the original chromosome code with the quantum state vector, one chromosome can represent the superposition of multiple states and quantum logic gates are used to update chromosomes [41].

A quantum bit is an information storage unit in a quantum computer. It differs from a classical chromosome in that it can be in the superposition state of two quantum states at the same time, as shown in (11):

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{11}$$

In (12), (α, β) is referred to as the probability amplitude and is two amplitude constants that satisfy:

$$|\alpha^2| + |\beta^2| = 1 \tag{12}$$

In (11), $|0\rangle$ and $|1\rangle$ represent the spin-down and spin-up states, respectively, so a quantum bit can simultaneously exhibit information in either the "0" or "1" state. That is to say, genes encoded using quantum bits no longer contain certain information but contain all possible information [42].

By manipulating the gene, QGA has better diversity features compared to GA. Using quantum bit encoding can also achieve good convergence. As $|\alpha^2|$ approaches 0 and $|\beta^2|$ approaches 1, the chromosomes encoded by quantum bits will converge to a single state.

1) QUANTUM BIT ENCODING INITIALIZATION

Set "*Sizepop*" to represent the population size, "*Lenchrom*" to represent the binary length vector of each variable $[L_1\ L_2\ \ldots\ L_d]$, where d is the dimension of the optimized problem. The initialized population "chrome" is a matrix of the $[2 * Sizepop, sum(Lenchrom)]$-dimension.

The probability amplitude encoding of the population is shown in (13):

$$P_i = \begin{bmatrix} cos(\theta_{i1}) & cos(\theta_{i2}) & \ldots & cos(\theta_{in}) \\ sin(\theta_{i1}) & sin(\theta_{i2}) & \ldots & sin(\theta_{in}) \end{bmatrix} \tag{13}$$

where $\theta_{ij} = 2 * \pi * rand$, $i = 1, 2, 3, \ldots m$, m is the total number of chromosomes, θ is the rotation angle, *rand* is a random constant between 0 and 1, and n is the dimension of the solution space, $n = sum(Lenchrom)$.

From this, we can express the probability amplitude in (14) as follows:

$$\begin{cases} \alpha = cos(\theta) \\ \beta = sin(\theta) \end{cases} \tag{14}$$

2) QUANTUM BIT ENCODING CONVERSION

Set the random number "*rand*" in QGA, and set the position where *rand* is greater than $cos^2(\theta_{2*i-1,j})$ to $1\rangle$, and the position where rand is less than $cos^2(\theta_{2*i-1,j})$ to $0\rangle$, so that we can get a group of binary codes containing quantum information. (15) is the formula for converting the binary code into decimal variables:

$$X = X_{min} + \frac{De}{2^{L_l - 1}} \times (X_{max} - X_{min}) \tag{15}$$

where X_{min} is the lower limit of the variable, X_{max} is the upper limit of the variable, De is the decimal number converted from binary encoding to decimal, and X is the d-dimensional variable. $l = 1, 2, \ldots d$.

3) STATUS UPDATE

The rotation angle in QGA is generally updated through Table 1 [43]:

In Table 1, χ_i is the i-th position of the current chromosome; $best_i$ is the i-th position of the current optimal chromosome; $f(\chi)$ is the fitness function; $\Delta\theta_i$ is the updated rotation angle size. The update strategy of the rotation angle is to compare the fitness of the current measurement value with the fitness of the optimal chromosome. If $f(\chi) > f(best)$, adjust the quantum bits at the corresponding positions in the current measurement value, so that the probability amplitude (α_i, β_i) evolves in the direction of favorable 'χ_i'.On the contrary, it evolves in the direction of '$best_i$'.

TABLE 1. QGA rotation angle update strategy.

χ_i	$best_i$	$f(\chi) > f(best)$	$\Delta\theta_i$	$s(\alpha_i, \beta_i)$			
				$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	×	0	0	0	0	0
0	0	√	0	0	0	0	0
0	1	×	$0.01 * \pi$	+1	-1	0	±1
0	1	√	$0.01 * \pi$	-1	+1	±1	0
1	0	×	$0.01 * \pi$	-1	+1	±1	0
1	0	√	$0.01 * \pi$	+1	-1	0	±1
1	1	×	0	0	0	0	0
1	1	√	0	0	0	0	0

4) ROTATION ANGLE UPDATE IMPROVEMENT

However, in the update strategy of the rotation angle mentioned above, the updated value of the rotation angle is fixed, and excessive fixed values may occur during the operation process, leading to the algorithm missing the optimal solution and constantly taking values around the optimal solution. A fixed value that is too small can lead to slow algorithm convergence and the possibility of falling into local optima.

This paper is based on the idea of velocity update in PSO [44], and adaptively updates the rotation angle. The speed update in PSO is shown in (16):

$$v_{id}^{t+1} = \varpi v_{id}^{t} + c_1 r_1 (p_{id}^{t} - x_{id}^{t}) + c_2 r_2 (p_{gd}^{t} - x_{id}^{t}) \quad (16)$$

In (16), d is the dimension of the optimization problem; t is the current number of iterations; ϖ is the weight, c_1, c_2 is a constant coefficient, r_1, r_2 is a random number within 0∼1; p_i is the individual optimal value; p_g is the global optimal value.

Based on the idea of (16), this paper modifies the formula for updating the rotation angle increment to (17):

$$\begin{cases} \Delta\theta_{ij}(t+1) = |\varpi \Delta\theta_{ij}(t) + c_1 r_1 (\Delta\theta_b) + c_2 r_2 (\Delta\theta_g)| \\ \Delta\theta_b = \theta_b - \theta_{ij} \\ \Delta\theta_g = \theta_g - \theta_{ij} \end{cases} \quad (17)$$

In (17), θ_b is the optimal rotation angle of the individual at the quantum state position that the chromosome has previously searched for. θ_g is the global optimal rotation angle of the globally optimal chromosome at the quantum state position. The direction of the rotation angle is (18):

$$\Delta s_{ij} = sign(\begin{vmatrix} \alpha_b & \alpha_{ij} \\ \beta_b & \beta_{ij} \end{vmatrix}) \quad (18)$$

where Δs_{ij} is the direction of the j-th quantum on the i-th chromosome, and $(\alpha_b, \beta_b)^{\mathrm{T}}$ is the global optimal probability amplitude at that position.

The inertia weight ϖ directly affects the convergence speed of the algorithm. A larger ϖ is more advantageous for the algorithm to perform a global search in the early stage and discover more suitable regions. In the later stage, a relatively smaller ϖ is needed for the algorithm to perform a local search. This paper proposes a nonlinear adaptive weight ϖ, as shown in (19):

$$\varpi = \begin{cases} \varpi_{max} - (\dfrac{(\varpi_{max} - \varpi_{min}) \times (f_i - f_{av})}{f_{min} - f_{av}} & f_i > f_{av} \\ \varpi_{max} - (\varpi_{max} - \varpi_{min}) \times (\dfrac{t}{MaxIert})^3 & f_i \le f_{av} \end{cases} \quad (19)$$

In (19), f_i is the fitness of the current chromosome, f_{min} is the minimum fitness value of the current iteration, f_{av} is the average fitness value of the current iteration, ϖ_{max} is the maximum weight, ϖ_{min} is the minimum weight, and $MaxIert$ is the maximum number of iterations. When $f_i > f_{av}$, we expect the weight w to be adaptively adjusted to accelerate convergence speed. When $f_i \le f_{av}$, we expect the weight w to gradually reduce the search space with the number of iterations, making it easier to find the optimal value nearby. The variation pattern of ϖ when $f_i \le f_{av}$ is shown in Figure 4:

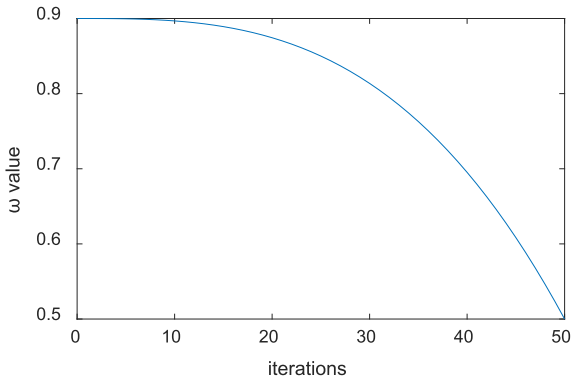

FIGURE 4. The variation pattern ϖ of when $f_i \le f_{av}$.

The quantum bit probability amplitude update method uses the quantum rotation gate strategy to update, as shown in (20):

$$\begin{aligned} &\begin{bmatrix} cos(\theta_{ij}(t+1)) \\ sin(\theta_{ij}(t+1)) \end{bmatrix} \\ &= \begin{bmatrix} cos(\Delta\theta_{ij}(t+1)) & -sin(\Delta\theta_{ij}(t+1)) \\ sin(\Delta\theta_{ij}(t+1)) & cos(\Delta\theta_{ij}(t+1)) \end{bmatrix} \times \begin{bmatrix} cos(\theta_{ij}(t)) \\ sin(\theta_{ij}(t)) \end{bmatrix} \\ &= \begin{bmatrix} cos(\theta_{ij}(t + \Delta\theta_{ij}(t+1)) \\ sin(\theta_{ij}(t + \Delta\theta_{ij}(t+1)) \end{bmatrix} \end{aligned} \quad (20)$$

The updated probability amplitude position is (21), as shown at the bottom of the next page.

It can be seen from (20) and (21) that the quantum rotation gate realizes the simultaneous movement of the two positions of the quantum state by changing the quantum bit phase of the chromosome. Therefore, if the population size of the problem does not change, then the use of quantum bit coding can expand the ergodicity of chromosome (or we can also call it particle) optimization, and improve the global optimization ability of the algorithm.

5) MUTATION AND CATASTROPHE

In order to ensure that QGA does not fall into local optima as much as possible, mutation operators are introduced in

evolution. Using Hadamard gates for mutation, as shown in (22):

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\times\begin{bmatrix}cos(\theta_{ij})\\ sin(\theta_{ij})\end{bmatrix}=\begin{bmatrix}cos(\frac{\pi}{4}-\theta_{ij})\\ sin(\frac{\pi}{4}-\theta_{ij})\end{bmatrix} \tag{22}$$

$$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\times\begin{bmatrix}cos(\theta_{ij})\\ sin(\theta_{ij})\end{bmatrix}=\begin{bmatrix}cos(\frac{\pi}{2}-\theta_{ij})\\ sin(\frac{\pi}{2}-\theta_{ij})\end{bmatrix} \tag{23}$$

Compared to the compilation method using the quantum non-gate of (23), the possibility of quantum crossing the optimal value due to mutation using the Hadamard gate of (22) is smaller. The probability of variation is set to 0.01 in this article.

Catastrophe is a step set to prevent the algorithm from being locally optimal and unable to jump out. The catastrophe strategy is relatively simple. When the optimal fitness value of the algorithm is the same for three consecutive times, start the catastrophe strategy according to the principle of survival of the fittest, and reinitialize the worst chromosomes in this iteration. Generally, the number is one-tenth of the total population. This can prevent the optimal fitness from being close to the global optimal but being erased, The population diversity has been updated again.

The flowchart of the IQGA is shown in Figure 5:

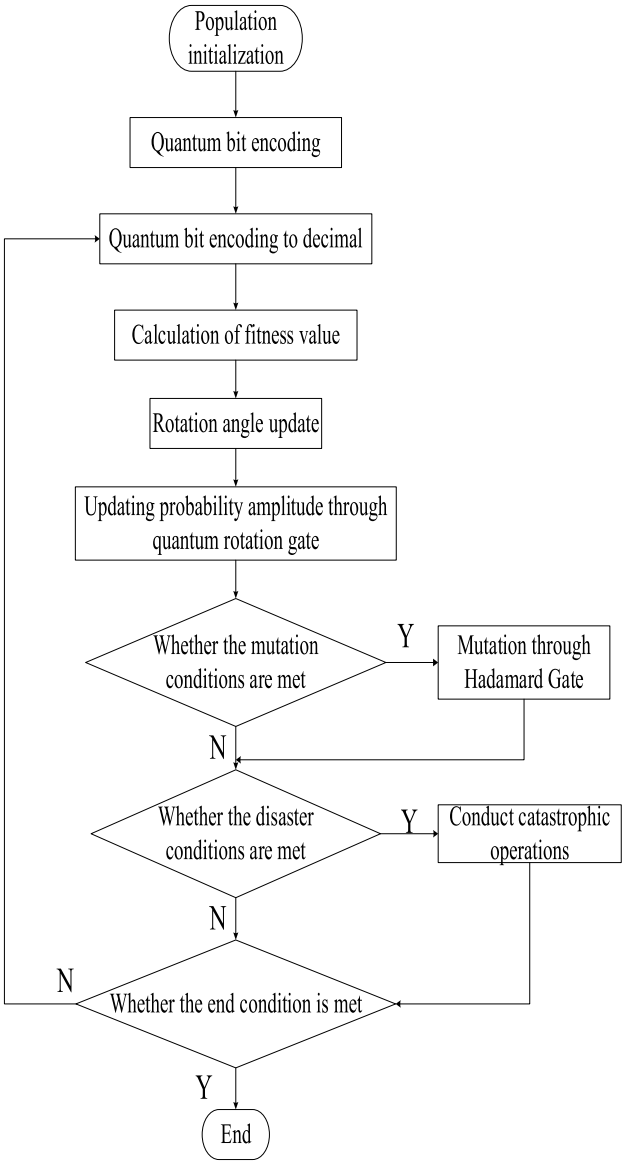

FIGURE 5. Flowchart of IQGA.

IV. ALGORITHM TESTING EVALUATION

In order to test the optimization ability of the IQGA algorithm in this paper to prove the feasibility of the improved algorithm, The Ackley function, Rosenbrock function, Rastigin function, and Schaffer function are selected for testing, and compared with the basic particle swarm optimization algorithm (BPSO), the algorithm in [23] (QGA), the algorithm in [19] (hereinafter referred to as IPSO algorithm in this paper) and the algorithm in [25] (hereinafter referred to as AQGA algorithm in this paper), The number of experimental iterations is 50, and the population size is 50. Each group of experiments is conducted 50 times.

A. ACKLEY FUNCTION TESTING

The Ackley function formula is shown in (24):

$$f(x,y)=-A\times exp(-B\times\sqrt{\frac{1}{d}\sum_{i=1}^{d}x^2})-exp(\sqrt{\frac{1}{d}\sum_{i=1}^{d}cos(C\pi\times y^2)})+A+exp(1) \tag{24}$$

This paper sets A=20, B=0.2, C=2. $x=y\in(-10:10)$.

This function has a minimum value of $f(0,0)=0$; The function image is shown in Figure 6:

Take a presentation with better fitness value in each iteration, as shown in Figure 7:

B. RASTIGIN FUNCTION TESTING

The Rastigin function formula is shown in (25):

$$f(x,y)=\sum_{i=1}^{d}[x_i^2-A\times cos(2\pi\times y_i)+B] \tag{25}$$

$$P_i^{t+1}=\begin{bmatrix}cos(\theta_{i1}(t)+\Delta\theta_{i1}(t+1)) & \dots & cos(\theta_{in}(t)+\Delta\theta_{in}(t+1))\\ sin(\theta_{i1}(t)+\Delta\theta_{i1}(t+1)) & \dots & sin(\theta_{in}(t)+\Delta\theta_{in}(t+1))\end{bmatrix} \tag{21}$$

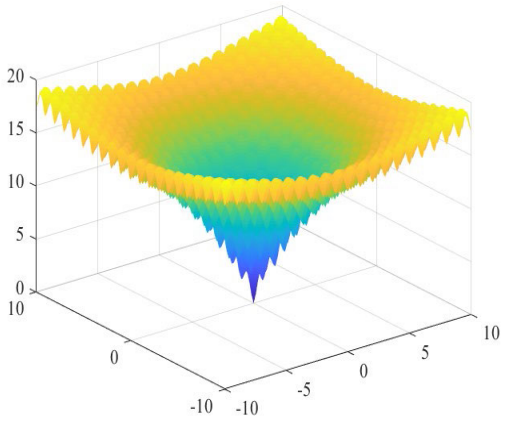

FIGURE 6. Ackley function image.

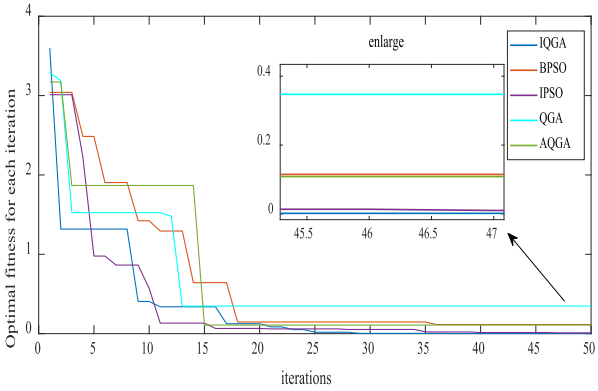

FIGURE 7. A group of 5 algorithms that optimize Ackley functions better.

This paper sets A=10, B=10; $x = y \in (-10 : 10)$, This function has a minimum value of $f(0, 0) = 0$; The function image is shown in Figure 8:

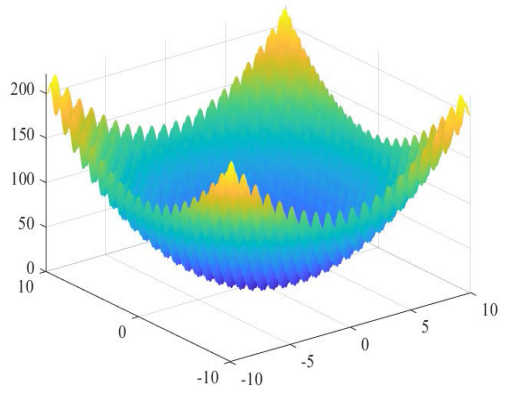

FIGURE 8. Rastigin function image.

Take a presentation with better fitness value in each iteration, as shown in Figure 9:

C. ROSENBROCK FUNCTION TESTING

The Rosenbrock function formula is shown in (26):

$$f(x, y) = (1 - x)^2 + A \times (y - x^2)^2 \tag{26}$$

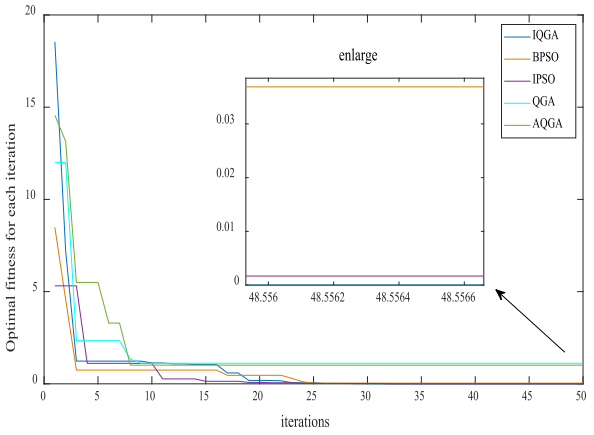

FIGURE 9. A group of 5 algorithms that optimize Rastigin functions better.

This article sets A=1; $x = y \in (-10 : 10)$, This function has a minimum value of $f(0, 0) = 0$; The function image is shown in Figure 10:

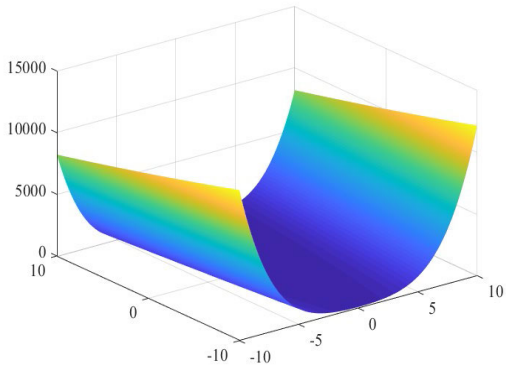

FIGURE 10. Rosenbrock function image.

Take a presentation with a better fitness value in each iteration, as shown in Figure 11:

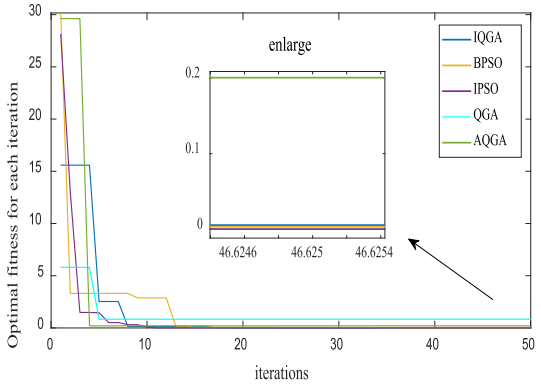

FIGURE 11. A group of 5 algorithms that optimize Rosenbrock functions better.

D. SCHAFFER FUNCTION TESTING

The Schaffer function formula is shown in (27):

$$f(x, y) = 0.5 - \frac{(sin\sqrt{x^2 + y^2})^2 - 0.5}{[1 + 0.001 \times (x^2 + y^2)]^2} \tag{27}$$

This article assumes that $x = y \in (-10 : 10)$; This function has a minimum value of $f(0, 0) = 0$; The function image is shown in Figure 12:

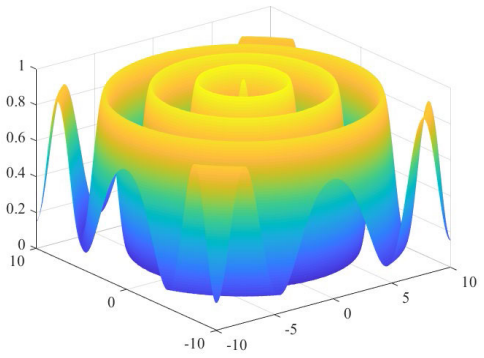

FIGURE 12. Schaffer function image.

Take a presentation with a better fitness value in each iteration, as shown in Figure 13:

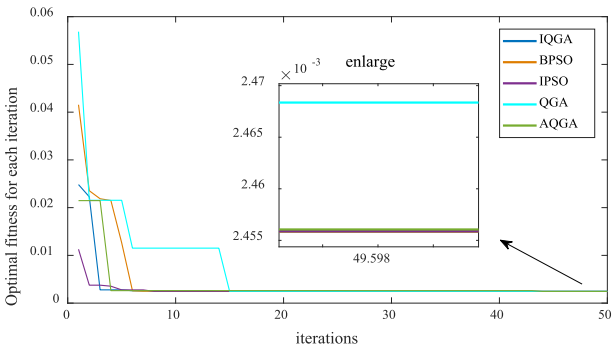

FIGURE 13. A group of 5 algorithms that optimize Schaffer functions better.

E. TEST FUNCTION ANALYSIS

Five algorithms test four different functions, and the average minimum fitness value of each group of 50 tests is shown in Table 2:

TABLE 2. Average minimum fitness value of five algorithms.

	IQGA	BPSO	IPSO	QGA	AQGA
Ackley function	0.02851	0.0590	0.03549	0.34333	0.2440
Rastrigin function	0.00172	0.12525	0.01841	1.00261	0.8777
Rosenbrock function	0.01904	0.0213	0.00972	0.8331	0.4664
Schaffer function	0.0024	0.0025	0.0025	0.0025	0.0025

From Table 2, we can be seen that the IQGA algorithm used in this article is the best in most cases. However, due to the smooth surface of the Rosenbrock function and the structure of the minimum being the center value, the PSO does not fall into local optima and can better find the minimum value. Therefore, when optimizing this function, the algorithm in this article is slightly inferior to the IPSO algorithm but still stronger than other algorithms. The Overall comparison shows that the algorithm proposed in this paper has certain advantages in terms of optimization and iteration.

V. SIMULATION ANALYSIS OF PMSM

In the simulation experiment, the population of the 5 algorithms is set to 50, and the values of iterations are set to [0-100]. The various parameters of PMSM during simulation are shown in Table 3:

TABLE 3. Various parameters of PMSM.

Symbol	Parameter name	Parameter Value/Unit
P_n	Pole pairs	4
L_s	Stator phase inductance	5.25/mH
R_s	Stator phase resistance	0.958/ Ω
ψ_f	Flux linkage established by magnets	0.1827/Wb
J	Moment of Inertia	0.006/kg · m^2
B	Friction factor	0.008/N · m · s
T	Simulation time	0.5/s
T_s	Sampling period	10e-6/s
V_{dc}	DC side voltage	311/V
β	Speed-loop bandwidth	50/rad/s

Figure 14 shows the IQGA optimized PID control block diagram.

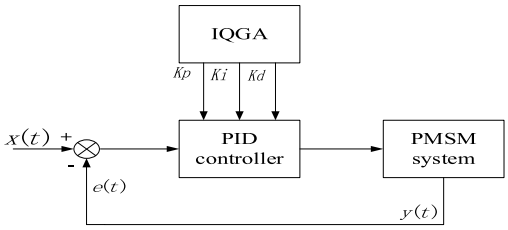

FIGURE 14. IQGA optimized PID control block diagram.

This paper selects ITAE as the error performance index to evaluate the rapidity and accuracy of the system, as shown in (28).

$$\mathrm{ITAE} = \int_0^\infty t|e(t)| \tag{28}$$

Figure 14 shows "simout" as $t|e(t)|$, and the integration operation is completed in the m file. From (7) and Table 3, we can be determined that $B_a = 0.02664$.

This article considers PID optimization for the following three scenarios

1) No load, with a fixed speed of 1200 (r/min).

2) At 0 seconds, there is no load, and at 0.2 seconds, a load of 10N · m is added, with a fixed speed of 1200 (r/min).

3) At 0 seconds, there is no load and the speed is 1200 (r/min). At 0.2 seconds, a load of 10N m is added, and the speed increases to 1500 (r/min) at 0.25 seconds.

Compare with QGA [23], AQGA [25], IGA [15], and IPSO [19]. The PID parameters obtained by each algorithm are shown in Table 4:

TABLE 4. The final optimized parameters of Kp/Ki/Kd.

Algorithm/PID parameters	Kp	Ki	Kd
IQGA	96.4874	0.53278	0.10374
QGA	37.1034	5.07422	0.149441
AQGA	97.6852	3.48082	0.248814
IGA	34.5	3.01	0.10
IPSO	100	1.137186	0.0211256

A. ANALYSIS OF THE 1(ST) SCENARIO

From Figure 15, we can be seen that although IPSO reached the required speed at the fastest, there was an overshoot of approximately 67r followed by a rebound of over 60r at 0.137s, and the oscillation remained stable at 0.0185s;

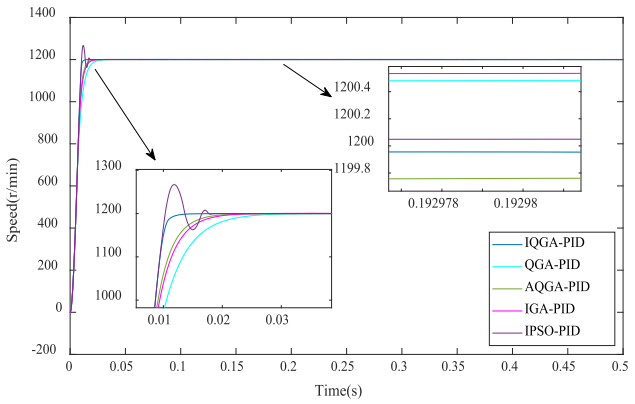

FIGURE 15. Speed comparison under no-load condition.

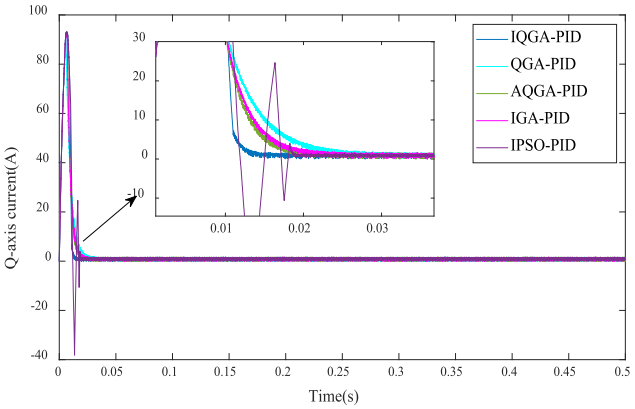

FIGURE 16. Q-axis current under no-load conditions.

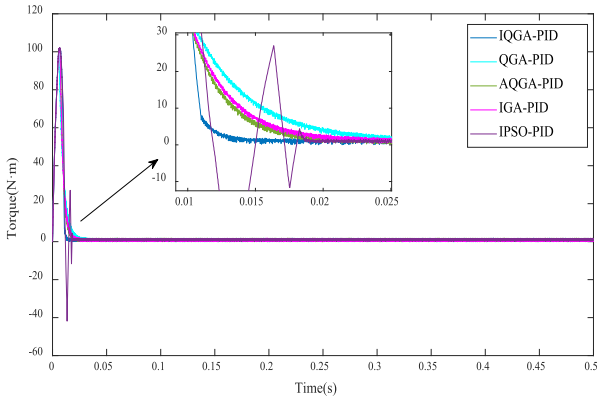

FIGURE 17. Torque under no-load conditions.

IGA has no overshoot or oscillation, stable at 0.0255 seconds; AQGA has no overshoot or oscillation, and is stable within 0.024 seconds; QGA has no overshoot or oscillation, stable at 0.034s; IQGA has no overshoot or oscillation, and is stable at 0.014s. After stabilization, the errors of IQGA and IPSO are both within 0.1r, AQGA is within 0.25r, and IGA and QGA are around 0.5r.

From Figure 16, we can see that the overshoot current of the five algorithms during motor startup is roughly the same, while IPSO has the second fastest stabilization time, but severe oscillations occur. IQGA has the fastest stabilization time and no oscillations.

From Figure 17, we can see that the electromagnetic torque of IQGA reaches stability the fastest, and IPSO has severe oscillation. The oscillation amplitude after stabilization of the five algorithms is roughly the same.

In summary, the algorithm proposed in this paper can effectively control the motor to quickly maintain stability under no load conditions.

B. ANALYSIS OF THE 2(ND) SCENARIO

From Figure 18, we can be seen that the five algorithms did not show significant changes in the initial startup phase

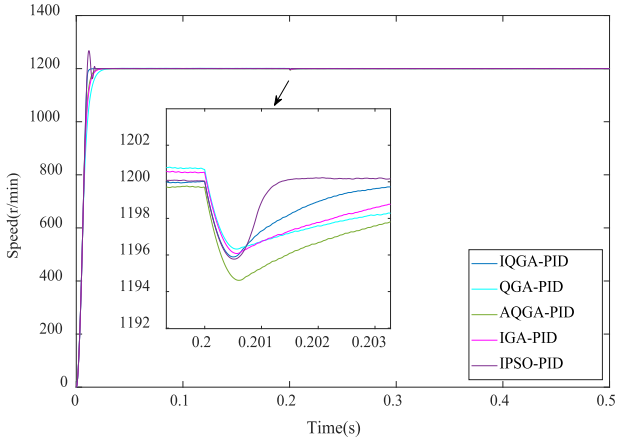

FIGURE 18. The comparison of speed response under the fixed load condition.

compared to the unloaded state. After adding a load of 10N m at 0.2s, the IPSO reached its lowest point of 1995.8r and stabilized at 0.2016s. The lowest point of IGA downward was 1196.08r, which stabilized at 0.2072s; The lowest point of AQGA downward was 1194.62r and stabilized at 0.207s; The lowest point of QGA downward was 1196.33r and stabilized at 0.2086s; The lowest point of IQGA downward was 1195.9r and stabilized at 0.2042s.

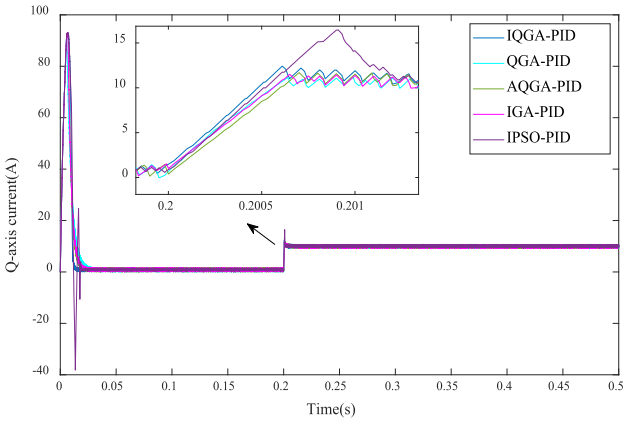

FIGURE 19. Q-axis current under the fixed load condition.

From Figure 19, we can see that when the load is applied in 0.2 seconds, IPSO has an overshoot current of about 7A, and the current requires a longer stabilization time compared to the other four algorithms. The difference between the other four algorithms is not significant.

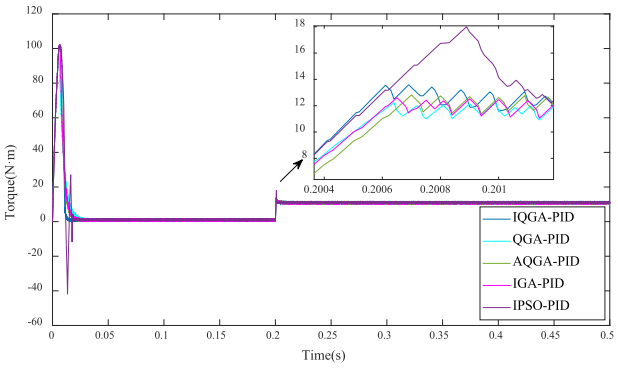

FIGURE 20. Torque under the fixed load condition.

From Figure 20, we can see that after adding the load, IPSO has a certain degree of overshoot, and IQGA is the first to stabilize, but the advantage is not obvious.

In summary, the impact of the five algorithms on speed is not significant after adding load, with IPSO being relatively prominent in terms of recovery time, followed by IQGA, indicating that PSO has stronger resistance to load. However, the q-axis current overshoot and torque overshoot of IPSO is more pronounced and the recovery time is longer compared to other algorithms.

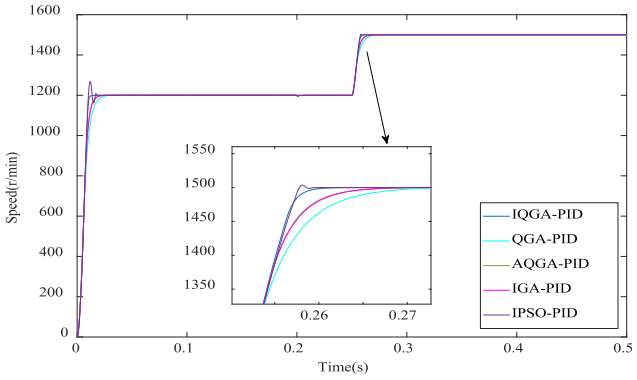

FIGURE 21. The comparison of speed response under the speed changes condition.

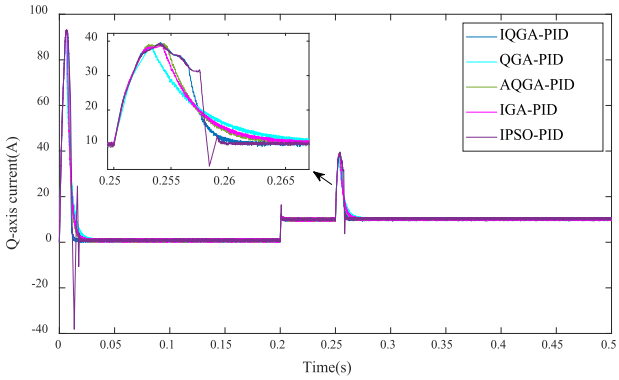

FIGURE 22. Q-axis current under the speed changes condition.

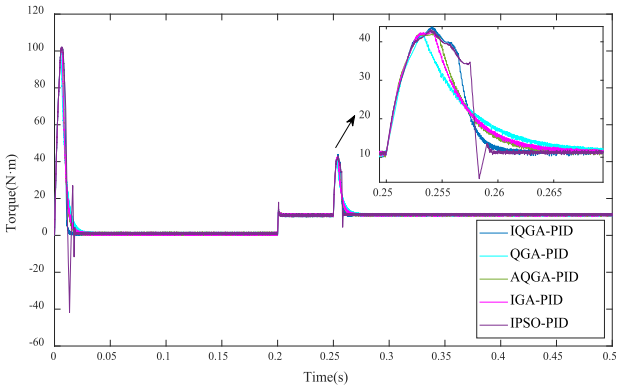

FIGURE 23. Torque under the speed changes condition.

C. ANALYSIS OF THE 3(RD) SCENARIO

Figure 21 shows that there is no significant change in the five algorithms compared to the previous two scenarios during the initial startup stage, and there is no significant change compared to the second scenario after adding a load. When the speed rises to 1500r at 0.25s, the IPSO also exhibits significant oscillations and tends to stabilize in 0.264s; IGA has no oscillation and tends to stabilize at 0.275 seconds; AQGA has no oscillation and tends to stabilize at 0.272 seconds; QGA has no oscillation and tends to stabilize at 0.284 seconds; IQGA has no oscillation and tends to stabilize at 0.264s.

From Figure 22, we can see that when the speed increases from 0.25s to 1500r, IPSO has a slight oscillation, and the time for IQGA to recover stability is almost the same as IPSO, which is faster than the other three algorithms. The overshoot of the five algorithms is almost equal.

From Figure 23, we can see that when the speed increases to 1500r, the IPSO exhibits slight oscillation, which is roughly the same as the stabilization time of IQGA. However, IQGA does not experience oscillation, followed by IGA and AQGA, and finally stabilizes.

In summary, IPSO and IQGA are relatively prominent among the five algorithms, and IQGA does not experience significant oscillations, which has a better performance in extending the lifespan of the motor.

VI. CONCLUSION

This paper proves that IQGA proposed in this paper has better performance in iterative and extremum-seeking processes by searching for the minimum values of four different functions: Ackley, Rastigin, Rosenbrock, and Schaffer. This provides a basis for finding the optimal parameters in PID control and applying them to PMSM to achieve faster speed and a stable state

In the simulation, IQGA can achieve the required speed faster than QGA, IGA, and AQGA, and there is no overshoot or oscillation compared to IPSO, and the speed is more stable. From the perspective of q-axis current and electromagnetic torque, this algorithm still has certain advantages compared to the other four algorithms.

REFERENCES

[1] P. Zhao and G. Yang, "Torque density improvement of five-phase PMSM drive for electric vehicles applications," *J. Power Electron.*, vol. 11, no. 4, pp. 401–407, Jul. 2011, doi: 10.6113/jpe.2011.11.4.401.

[2] S. Fang, Y. Wang, and H. Liu, "Design study of an aerospace motor for more electric aircraft," *IET Electr. Power Appl.*, vol. 14, no. 14, pp. 2881–2890, Dec. 2020, doi: 10.1049/iet-epa.2020.0507.

[3] T. Yuan, D. Wang, X. Wang, X. Wang, and Z. Sun, "High-precision servo control of industrial robot driven by PMSM-DTC utilizing composite active vectors," *IEEE Access*, vol. 7, pp. 7577–7587, 2019.

[4] Y. Deng, J. Wang, H. Li, J. Liu, and D. Tian, "Adaptive sliding mode current control with sliding mode disturbance observer for PMSM drives," *ISA Trans.*, vol. 88, pp. 113–126, May 2019, doi: 10.1016/j.isatra.2018.11.039.

[5] N. Ren, L. Fan, and Z. Zhang, "Sensorless PMSM control with sliding mode observer based on sigmoid function," *J. Electr. Eng. Technol.*, vol. 16, no. 2, pp. 933–939, Mar. 2021, doi: 10.1007/s42835-021-00661-4.

[6] Z. Zhang, X. Liu, J. Yu, and H. Yu, "Time-varying disturbance observer based improved sliding mode single-loop control of PMSM drives with a hybrid reaching law," *IEEE Trans. Energy Convers.*, early access, May 18, 2023, doi: 10.1109/TEC.2023.3277628.

[7] H. Liu, W. Lin, Z. Liu, C. Buccella, and C. Cecati, "Model predictive current control with model-aid extended state observer compensation for PMSM drive," *IEEE Trans. Power Electron.*, vol. 38, no. 3, pp. 3152–3162, Mar. 2023, doi: 10.1109/TPEL.2022.3225626.

[8] X. Sun, T. Li, Z. Zhu, G. Lei, Y. Guo, and J. Zhu, "Speed sensorless model predictive current control based on finite position set for PMSHM drives," *IEEE Trans. Transport. Electrific.*, vol. 7, no. 4, pp. 2743–2752, Dec. 2021, doi: 10.1109/TTE.2021.3081436.

[9] X. Sun, T. Li, M. Yao, G. Lei, Y. Guo, and J. Zhu, "Improved finite-control-set model predictive control with virtual vectors for PMSHM drives," *IEEE Trans. Energy Convers.*, vol. 37, no. 3, pp. 1885–1894, Sep. 2022, doi: 10.1109/TEC.2021.3138905.

[10] T. Li, X. Sun, G. Lei, Y. Guo, Z. Yang, and J. Zhu, "Finite-control-set model predictive control of permanent magnet synchronous motor drive systems—An overview," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 12, pp. 2087–2105, Dec. 2022, doi: 10.1109/JAS.2022.105851.

[11] H. Ding, X. Qin, and L. Wei, "Sensorless control of surface-mounted permanent magnet synchronous motor using adaptive robust UKF," *J. Electr. Eng. Technol.*, vol. 17, no. 5, pp. 2995–3013, Sep. 2022, doi: 10.1007/s42835-022-01061-y.

[12] Z. Wang, X. Liu, W. Wang, Y. Lv, B. Yuan, W. Li, Q. Li, S. Wang, Q. Chen, and Y. Zhang, "UKF-based parameter estimation and identification for permanent magnet synchronous motor," *Frontiers Energy Res.*, vol. 10, p. 114, Feb. 2022, doi: 10.3389/fenrg.2022.855649.

[13] W. J. Zheng, Y. Luo, Y. Q. Chen, and X. H. Wang, "A simplified fractional order PID controller's optimal tuning: A case study on a PMSM speed servo," *Entropy*, vol. 23, no. 2, Feb. 2021, Art. no. 130, doi: 10.3390/e23020130.

[14] H. H. Choi, H. M. Yun, and Y. Kim, "Implementation of evolutionary fuzzy PID speed controller for PM synchronous motor," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 540–547, Apr. 2015, doi: 10.1109/tii.2013.2284561.

[15] N. Soundirarrajan and K. Srinivasan, "Performance evaluation of ant lion optimizer-based PID controller for speed control of PMSM," *J. Test. Eval.*, vol. 49, no. 2, pp. 1104–1118, Mar. 2021, doi: 10.1520/jte20180892.

[16] T. Wang, H. Wang, C. Wang, and H. Hu, "A novel PID controller for BLDCM speed control using dual fuzzy logic systems with HSA optimization," *Sci. Rep.*, vol. 12, no. 1, p. 19, Jul. 2022, doi: 10.1038/s41598-022-15487-x.

[17] T. Nishida and T. Sakamoto, "Adaptive PSO for online identification of time-varying systems," *Electron. Commun. Jpn.*, vol. 95, no. 7, pp. 10–18, Jul. 2012, doi: 10.1002/ecj.11391.

[18] J. L. Fernandez-Martinez and E. Garcia-Gonzalo, "Stochastic stability analysis of the linear continuous and discrete PSO models," *IEEE Trans. Evol. Comput.*, vol. 15, no. 3, pp. 405–423, Jun. 2011, doi: 10.1109/tevc.2010.2053935.

[19] R.-M. Jan, C.-S. Tseng, and R.-J. Liu, "Robust PID control design for permanent magnet synchronous motor: A genetic approach," *Electr. Power Syst. Res.*, vol. 78, no. 7, pp. 1161–1168, Jul. 2008, doi: 10.1016/j.epsr.2007.09.011.

[20] H. S. Hu, T. T. Wang, S. Y. Zhao, and C. H. Wang, "Speed control of brushless direct current motor using a genetic algorithm–optimized fuzzy proportional integral differential controller," *Adv. Mech. Eng.*, vol. 11, no. 11, p. 13, Nov. 2019, doi: 10.1177/1687814019890199.

[21] H. Chen, X. Wang, M. Benbouzid, J. F. Charpentier, N. Ait-Ahmed, and J. G. Han, "Improved fractional-order PID controller of a PMSM-based wave compensation system for offshore ship cranes," *J. Mar. Sci. Eng.*, vol. 10, no. 9, Sep. 2022, Art. no. 1238, doi: 10.3390/jmse10091238.

[22] T. Wu, C. Zhou, Z. Yan, H. Peng, and L. Wu, "Application of PID optimization control strategy based on particle swarm optimization (PSO) for battery charging system," *Int. J. Low-Carbon Technol.*, vol. 15, no. 4, pp. 528–535, Nov. 2020, doi: 10.1093/ijlct/ctaa020.

[23] H. Feng, W. Ma, C. B. Yin, and D. H. Cao, "Trajectory control of electro-hydraulic position servo system using improved PSO-PID controller," *Autom. Constr.*, vol. 127, Jul. 2021, Art. no. 103722, doi: 10.1016/j.autcon.2021.103722.

[24] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002.

[25] Z. Laboudi and S. Chikhi, "Comparison of genetic algorithm and quantum genetic algorithm," *Int. Arab J. Inf. Technol.*, vol. 9, no. 3, pp. 243–249, May 2012.

[26] B. Chen, H. Lei, H. Shen, Y. Liu, and Y. Lu, "A hybrid quantum-based PIO algorithm for global numerical optimization," *Sci. China Inf. Sci.*, vol. 62, no. 7, p. 12, Jul. 2019, doi: 10.1007/s11432-018-9546-4.

[27] Z. B. Su, Y. W. Lu, J. T. Gu, R. Gao, M. Zheng, and Q. M. Kong, "Research on rice leaf area index inversion model based on improved QGA-ELM algorithm," (in Chinese), *Spectrosc. Spectral Anal.*, vol. 41, no. 4, pp. 1227–1233, Apr. 2021. [Online]. Available: http://www.gpxygpfx.com/EN/10.3964/j.issn.1000-0593(2021)04-1227-07

[28] L. Xu, S. Zhao, N. Li, Q. Gao, T. Wang, and W. Xue, "Application of QGA-BP for fault detection of liquid rocket engines," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 5, pp. 2464–2472, Oct. 2019.

[29] Y. Ren, D. Zhang, Y. He, R. Chang, S. Guo, S. Ma, M. Yao, and F, Guan, "Injectable and antioxidative HT/QGA hydrogel for potential application in wound healing," *Gels*, vol. 7, no. 4, p. 13, Dec. 2021, Art. no. 204, doi: 10.3390/gels7040204.
[30] T. Y. Liu, Q. Sun, H. C. Zhou, and Q. Wei, "Optimization of network coding resources based on improved quantum genetic algorithm," *Photonics*, vol. 8, no. 11, p. 16, Nov. 2021, Art. no. 502, doi: 10.3390/photonics8110502.
[31] S. Huang, X. Guo, R. Wang, and Y. Mei, "A SVPWM algorithm based on four-switch three-phase inverter for PMSM under the imbalance of bus capacitor voltage," *J. Power Electron.*, vol. 21, no. 12, pp. 1812–1822, Dec. 2021, doi: 10.1007/s43236-021-00324-7.
[32] W. Liang, J. Wang, P. C. Luk, W. Fang, and W. Fei, "Analytical modeling of current harmonic components in PMSM drive with voltage-source inverter by SVPWM technique," *IEEE Trans. Energy Convers.*, vol. 29, no. 3, pp. 673–680, Sep. 2014, doi: 10.1109/tec.2014.2317072.
[33] X. Lin, W. Huang, and L. Wang, "SVPWM strategy based on the hysteresis controller of zero-sequence current for three-phase open-end winding PMSM," *IEEE Trans. Power Electron.*, vol. 34, no. 4, pp. 3474–3486, Apr. 2019, doi: 10.1109/tpel.2018.2856372.
[34] B. K. Bose, "An adaptive hysteresis-band current control technique of a voltage-fed PWM inverter for machine drive system," *IEEE Trans. Ind. Electron.*, vol. 37, no. 5, pp. 402–408, 1990.
[35] M. N. Uddin, M. A. Abido, and M. A. Rahman, "Development and implementation of a hybrid intelligent controller for interior permanent magnet synchronous motor drive," in *Proc. IEEE Ind. Appl. Conf., 37th IAS Annu. Meeting*, Oct. 2002, pp. 1439–1446.
[36] T. Guo, Y. J. Chen, Q. H. Chen, T. L. Lin, and H. L. Ren, "An IPMSM control structure based on a model reference adaptive algorithm," *Machines*, vol. 10, no. 7, p. 15, Jul. 2022, Art. no. 575, doi: 10.3390/machines10070575.
[37] L. Harnefors, K. Pietilainen, and L. Gertmar, "Torque-maximizing field-weakening control: Design, analysis, and parameter selection," *IEEE Trans. Ind. Electron.*, vol. 48, no. 1, pp. 161–168, Feb. 2001.
[38] P. C. Krause, O. Wasynczuk, and S. D. Sudhoff, *Analysis of Electric Machinery and Drive Systems*. Piscataway, NJ, USA: IEEE Press, 2002.
[39] B. Le Pioufle, "Comparison of speed nonlinear control strategies for the synchronous servomotor," *Electr. Mach. Power Syst.*, vol. 21, no. 2, pp. 151–169, Mar. 1993.
[40] R. Ortega, A. Astolfi, and N. E. Barabanov, "Nonlinear PI control of uncertain systems: An alternative to parameter adaptation," *Syst. Control Lett.*, vol. 47, no. 3, pp. 259–278, Oct. 2002.
[41] X. Zhang, X. Wang, Z. Li, J. Huang, and Y. Zhang, "Optimization of impedance-accelerated inverse-time over-current protection based on improved quantum genetic algorithm," *Energies*, vol. 16, no. 3, p. 1119, Jan. 2023, doi: 10.3390/en16031119.
[42] B. Chen, T. Zhang, L. Cong, J. Ma, and W. Hu, "Forward kinematics of body posture perception using an improved BP neural network based on a quantum genetic algorithm," *Laser Phys. Lett.*, vol. 19, no. 9, Sep. 2022, Art. no. 095201.
[43] Y. J. Libin, "Research of quantum genetic algorith and its application in blind source separation," *J. Electron. Sci., English Ed.*, vol. 20, no. 1, pp. 62–68, 2003.
[44] K. Kanwar, J. Vajpai, and S. K. Meena, "Design of PSO tuned PID controller for different types of plants," *Int. J. Inf. Technol.*, vol. 14, no. 6, pp. 2877–2884, Oct. 2022, doi: 10.1007/s41870-022-01051-3.

HONGZHI WANG received the M.Tech. degree in communication and electronic system from the Jilin University of Technology, Changchun, Jilin, China, in 1993, and the Ph.D. degree in information and communication engineering from Jilin University, Changchun, in 2000. Currently, he is a Professor with the Department of Computer Science and Engineering, Changchun University of Technology, Changchun. His research interests include signal processing in communication, image processing, high order spectral estimation, and motor control.

SHUO XU received the B.S. degree from the Changchun University of Technology, China, in 2021, where he is currently pursuing the master's degree.

His main research interest includes motor control.

HUANGSHUI HU (Member, IEEE) received the Ph.D. degree in computer application technology from Jilin University, China, in 2012. He is currently a Professor with the College of Computer Science and Engineering, Changchun University of Technology, China. His research interests include topology control in wireless sensor networks and multifunction vehicle bus networks.