Research Article

PI Controller of Speed Regulation of Brushless DC Motor Based on Particle Swarm Optimization Algorithm with Improved Inertia Weights

Wei Xie,¹ Jie-Sheng Wang^{1,2} , and Hai-Bo Wang³

¹School of Electronic and Information Engineering, University of Science & Technology Liaoning, Anshan 114044, China

²School of International Finance and Banking, University of Science & Technology Liaoning, Anshan 114044, China

³HBIS Laoting Steel Co., Ltd., Tangshan 063600, China

Correspondence should be addressed to Jie-Sheng Wang; wang_jiesheng@126.com

Received 5 December 2018; Revised 22 January 2019; Accepted 12 February 2019; Published 24 February 2019

Academic Editor: Erik Cuevas

Copyright © 2019 Wei Xie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The brushless director current (DC) motor is a new type of mechatronic motor that has been developed rapidly with the development of power electronics technology and the emergence of new permanent magnet materials. Based on the speed regulation characteristics, speed regulation strategy, and mathematical model of brushless DC motor, a parameter optimization method of proportional-integral (PI) controller on speed regulation for the brushless DC motor based on particle swarm optimization (PSO) algorithm with variable inertia weights is proposed. The parameters of PI controller are optimized by PSO algorithm with five inertia weight adjustment strategies (linear descending inertia weight, linear differential descending inertia weight, incremental-decremented inertia weight, nonlinear descending inertia weight with threshold, and nonlinear descending inertia weight with control factor). The effectiveness of the proposed method is verified by the simulation experiments and the related simulation results.

1. Introduction

The design of brushless DC motor comes from the idea that electronic switch circuit is used to replace the mechanical commutator of brush DC motor. In order to achieve the effective control performance on the motor speed and the direction of motor rotation, the brushless DC motor must have a rotor position sensor and a commutation device consisting of a control circuit and a power inverter bridge [1]. Therefore, the brushless DC motor is a typical mechanical product, whose basic structure consists of three parts (the ontology of motor, power drive circuit, and position sensor). Compared with other types of motors, the brushless DC motor adopts the form of square wave excitation, which improves the utilization rate of the permanent magnet material, reduces the volume of the motor, increases the output power of the motor, and has the characteristics of high efficiency and high reliability.

A brushless DC motor is a self-synchronous rotating motor, which has a rotor with the permanent magnet,

a rotor position signal and a controller of the electronic commutation. Its commutation circuit can be independent or integrated on the ontology of motor. With the rapid development of computer technology and control theory, many microprocessors, such as single-chip microcomputers, digital signal processors (DSPs), field programmable gate arrays (FPGAs), and complex programmable logic devices (CPLDs), have achieved unprecedented development, which leads to a qualitative leap on the instruction speed and storage space and further promotes the development of the brushless DC motor. The brushless DC motor has been widely used in various fields, such as industrial equipment, instrumentation, household electric appliances, robots, and medical equipment. In addition, some advanced control strategies and methods are constantly applied to the control of brushless DC motor. These methods improve the control performance on the brushless DC motor, such as torque ripple suppression, rotational speed dynamics and steady-state response and system anti-interference to a certain extent, and expand the application range of the brushless DC motor [2–4].

The proportional-integral-derivative (PID) controller has been widely used in many industrial processes because it has simple structure, good robustness, and high reliability. The parameter self-tuning of PID controller is important in the industrial applications so as to obtain the high control quality. But the appropriate selection of PID controller parameters needs the expert knowledge about the given complex industrial processes and control theory [5]. Now many intelligent PID controllers have been proposed to solve the shortcomings of the typical PID controllers, such as the expert PID controller, the PID controller based rules, the PID controller based on the neural network, and the fuzzy PID controller [6]. But the nondifferentiated or nonlinear parameter searching space of the adopted PID controller will make the global optimum unacquirable. With the rapid development of swarm intelligent theory, many PID controller parameters self-tuning methods based on the swarm intelligent theory appeared, such as genetic algorithm (GA), extreme optimization (EO) algorithm, particle swarm optimization (PSO) algorithm, big bang–big crunch optimization (BB–BC) algorithm, artificial bee colony (ABC) algorithm, imperialist competitive algorithm (ICA), differential evolution (DE) algorithm, and harmony search (HS) algorithm, which have been applied in the parameters optimization of PID controller [7–14].

Particle swarm optimization (PSO) algorithm is a new evolutionary algorithm developed by the J. Kennedy and R. C. Eberhart [15]. Based on the observation of the activity behavior of animal swarms, the individual information sharing in the population is used to make the whole population's motion produce an evolution process from the disorder to the order in the problem solving space so as to obtain the optimal solution. The PSO algorithm has been widely used in many fields, such as the parameter estimation for time-delay chaotic system [16], the prediction of seismic slope stability [17], the carton heterogeneous vehicle routing problem [18], the optimal power flow problem [19], the optimal power management of plug-in hybrid electric vehicles [20], the structural failure prediction of multistoried RC buildings [21], the energy-efficient dynamic scheduling for a flexible flow shop [22], the multiattribute decision making [23], the electrochemical model parameter identification of a lithium-ion battery [24], the medical image segmentation [25], and the parameters optimization of PID controller [26, 27]. In this paper, based on the speed regulation characteristics, speed regulation strategy, and mathematical model of brushless DC motor, a parameter optimization method of the adopted PI controller on speed regulation for the brushless DC motor based on PSO algorithm with five inertia weights is proposed. The effectiveness of the proposed method is verified by simulation experiments. The paper is organized as follows. In Section 2, the mathematical model of brushless DC motor is introduced. In Section 3, the particle swarm optimization (PSO) algorithm with different inertia weights is discussed. The parameter tuning method of PID controller based on PSO algorithm is illustrated in Section 4. The simulation experiment and result analysis are carried out in Section 5. Finally, the conclusion is illustrated in the last part.

2. Mathematical Model of Brushless DC Motor

The brushless DC motor is similar in structure to the permanent magnet synchronous motor, which is mainly composed of the stator with armature winding and rotor with permanent magnet pole. The position sensor in the brushless DC motor plays the role of detecting the rotor magnetic pole position and providing correct commutation information for the logic switching circuit; that is to say it converts the position signal of the rotor magnetic pole into an electrical signal and then controls the stator winding commutation so as to make the current of the motor armature winding commutated in a certain order with the change of the position of the rotor and form the stepping rotating magnetic field through the air gap to drive the permanent magnet rotor to continuously rotate. Taking a two-pole and three-phase brushless DC motor as an example, the stator winding of the motor is concentrated full pitch winding connected with Y, the rotor is constructed with the hidden inner rotor structure, and the three hall elements are placed symmetrically in space 120° apart. Based on this structure, the following assumptions are made to simplify the analysis process.

- (1) Ignore the saturation of motor core and exclude eddy current loss and hysteresis loss.
- (2) Exclude the armature reaction. The air gap magnetic field distribution is approximately considered as the trapezoidal wave such that the flat top width is 120° electrical angle.
- (3) Ignore the cogging effect. The armature conductor is continuously and evenly distributed on the armature surface.
- (4) The power transistor and diode of the drive system inverter circuits have ideal switching characteristics.

The phase voltage of each phase winding in the brushless DC motor is composed of the resistance voltage drop and the winding induced potential. Any phase voltage of the winding can be expressed as

$$u_x = R_x i_x + e_{\Psi_x} \quad (1)$$

where u_x is the phase voltage, i_x is the phase current, and e_{Ψ_x} is the phase induced electromotive force. The subscript x represents the three-phase winding A, B, and C. R_x is the phase winding and $R_A = R_B = R_C = R$ in the three-phase symmetrical winding.

The induced electromotive force of the winding is numerically equal to the change rate of the flux with time through the winding closed loop, so the induced electromotive force here is equal to the positive value of change rate of flux, that is to say:

$$e_{\Psi_x} = \frac{d\Psi_x}{dt} \quad (2)$$

In addition to being hinged with the magnetic flux generated by its own current, each phase winding is also respectively hinged with the permanent magnet rotor and the magnetic flux generated by other winding currents. Taking the A-phase winding as an example, the flux linkage can be expressed as

$$\Psi_A = L_A i_A + M_{AC} i_C + \Psi_{pm}(\theta) \quad (3)$$

where $\Psi_{pm}(\theta)$ is the permanent magnet flux of the A-phase winding chain, θ is the rotor position angle, L_A is the self-inductance of A-phase winding, and M_{AB} and M_{AC} are the self-inductance of B-phase winding and the C-phase winding to the A-phase winding, respectively.

When the rotor position angle is α , the permanent magnet flux of the A-phase winding chain is described as

$$\Psi_{pm}(\alpha) = N\phi_{pm}(\alpha) \quad (4)$$

$$\phi_{pm}(\alpha) = \int_{-\pi/2+\phi}^{\pi/2+\alpha} B(\theta) S d\theta \quad (5)$$

where $\phi_{pm}(\alpha)$ is the permanent magnet flux of the A-phase winding chain when the rotor position angle is α , $B(\theta)$ is the magnetic density distribution of radial air gap of the rotor permanent magnet, which is trapezoidal along the θ angle, N is the number of winding, and S is the area enclosed by the winding on the inner diameter surface of the stator.

Substitute (2)-(5) into (1) to obtain

$$\begin{aligned} u_A &= Ri_A + \frac{d}{dt} (L_A i_A + M_{AB} i_B + M_{AC} i_C + \psi_{pm}) \\ u_A &= u_A \\ &= Ri_A + \frac{d}{dt} (L_A i_A + M_{AB} i_B + M_{AC} i_C) \\ &\quad + \frac{d}{dt} \left[NS \int_{-\pi/2+\theta}^{\pi/2+\theta} B(x) dx \right] \\ u_A &= Ri_A + \frac{d}{dt} (L_A i_A + M_{AB} i_B + M_{AC} i_C) + e_A \end{aligned} \quad (6)$$

where e_A is the opposite electromotive force of A-phase winding.

The rotor of a brushless DC motor is generally constructed with the surface-adhesive hidden pole structure, whose winding inductance is a constant that does not change with time. Since the structure of the stator three-phase winding is symmetrical, the self-inductance of each phase winding is equal, and the mutual inductance among all phase windings is also equal; that is to say $L_A = L_B = L_C = L$ and $M_{AB} = M_{BA} = M_{BC} = M_{CB} = M_{AC} = M_{CA} = M$. So substitute them into (6) to obtain

$$u_A = Ri_A + L \frac{di_A}{dt} + M \frac{di_A}{dt} + M \frac{di_B}{dt} + M \frac{di_C}{dt} + e_A \quad (7)$$

$$\begin{aligned} e_A &= \frac{d}{dt} \left[NS \int_{-\pi/2+\theta}^{\pi/2+\theta} B(x) dx \right] \\ &= NS \left[B\left(\frac{\pi}{2} + \theta\right) - B\left(-\frac{\pi}{2} + \theta\right) \right] \frac{d\theta}{dt} \\ e_A &= NS\omega \left[B\left(\frac{\pi}{2} + \theta\right) - B\left(-\frac{\pi}{2} + \theta\right) \right] \end{aligned} \quad (8)$$

where ω is the angular velocity of the electric motor.

According to the distribution of the air gap magnetic density, $B(\theta)$ has a period of 2π , and $B(\theta + \pi) = -B(\theta)$, so obtain

$$\begin{aligned} e_A &= NS\omega \left[B\left(\frac{\pi}{2} + \theta\right) - B\left(-\frac{\pi}{2} + \theta\right) \right] \\ &= 2NS\omega B\left(\frac{\pi}{2} + \theta\right) \end{aligned} \quad (9)$$

It can be seen that the changing waveform of the opposite electromotive force of A-phase along the angle θ is ahead of electrical angle $\pi/2$ than the air gap magnetic density distribution $B(\theta)$. So e_A can be written as

$$e_A = 2NS\omega B_m \int_A(\theta) = \omega\psi_m \int_A(\theta) \quad (10)$$

where B_m is the maximum of the magnetic density distribution of air gap of the rotor permanent magnet, ψ_m is the maximum of the permanent magnet flux of the chain in each phase winding, and $\int_A(\theta)$ is the waveform function of the opposite electromotive force of A-phase.

The three-phase currents meet the following condition.

$$i_A = i_B = i_C = 0 \quad (11)$$

So, (9) can be further simplified as

$$u_A = Ri_A + (L - M) \frac{di_A}{dt} + e_A \quad (12)$$

After derivation, the similar results can be obtained for the B-phase and C-phase, so the matrix equation of the phase voltage equation of the brushless DC motor can be expressed as

$$\begin{aligned} \begin{bmatrix} u_A \\ u_B \\ u_C \end{bmatrix} &= \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} \\ &\quad + \begin{bmatrix} L - M & 0 & 0 \\ 0 & L - M & 0 \\ 0 & 0 & L - M \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} \\ &\quad + \begin{bmatrix} e_A \\ e_B \\ e_C \end{bmatrix} \end{aligned} \quad (13)$$

Similar to other motors, the power and torque of a brushless DC motor can be analyzed from energy transfer angle. When the motor is running, it can absorb the electric power from the power supply. A small part of the electric power is converted into the copper consumption and iron consumption in these electric power and most of the power is transmitted into the torque action of the rotor permanent magnet passed to the rotor through the air gap magnetic field, which is equal to the sum of the product of the opposite electromotive force and the current of the three-phase winding, that is to say

$$P_e = e_A i_A + e_B i_B + e_C i_C \quad (14)$$

Excluding the mechanical losses and stray losses of the rotor, this power can be absolutely converted into the rotor kinetic energy.

$$P_e = T_e \Omega \quad (15)$$

where T_e is the electromagnetic torque and Ω is the mechanical angular velocity of the motor.

Equation (16) can be deduced by (14) and (15).

$$T_e = \frac{e_A i_A + e_B i_B + e_C i_C}{\Omega} \quad (16)$$

The above voltage equation and torque equation form a complete mathematical model of an electromechanical system by introducing the motion equation of the motor, which is described as follows:

$$T_e - T_L = J \frac{d\Omega}{dt} + B_v \Omega \quad (17)$$

where T_L is the load torque, J is moment of inertia of rotor, and $B_v \Omega$ is the viscous friction coefficient.

Equations (13), (16), and (17) together constitute the mathematical model of the brushless DC motor.

3. Particle Swarm Optimization (PSO) Algorithm

3.1. Principle of PSO Algorithm. The PSO algorithm proposed by Eberhart and Kennedy in 1995 originates and imitates the predation behavior of cooperation and competition of the bird flocks. The PSO algorithm searches for the optimal solution by random and iteration, whose positions are tracked by two extreme values in each iteration. The first is the optimal solution pBest found by the particle itself and the other is the optimal solution gBest found in the entire population, also known as the global extreme. Suppose a population consisting of M particles fly at a certain speed in a D -dimensional space. The properties of particle i at time t are described as follows.

(1) Position: $x_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{id}^t)^T$, $x_{id}^t \in [L_d, U_d]$, L_d is the lower limit of searching space and U_d is the upper limit of searching space.

(2) Velocity: $v_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{id}^t)^T$, $v_{id}^t \in [v_{\min,d}, v_{\max,d}]$, $v_{\min}$ and $v_{\max}$ are the minimum and maximum velocities respectively.

(3) The individual optimal position: $p_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{id}^t)^T$. The global optimal position: $p_g^t = (p_{g1}^t, p_{g2}^t, \dots, p_{gd}^t)^T$, $1 \leq d \leq D$, $1 \leq i \leq M$.

Then the position of the particle at time $t + 1$ can be obtained by the following equations:

$$v_{id}^{t+1} = w v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) \quad (18)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (19)$$

where r_1 and r_2 are random numbers uniformly distributed in the interval $(0, 1)$ and c_1 and c_2 are learning factors, generally $c_1 = c_2 = 2$.

FIGURE 1: Flowchart of PSO algorithm.

Equation (18) is mainly composed of three parts. The first part is v_{id}^t to indicate the previous movement velocity and direction of the particle. The second part is $c_1 r_1 (p_{id}^t - x_{id}^t)$. The current position and the optimal individual position of the particles are compared and the related difference is used to determine the next direction of movement. The third part is $c_2 r_2 (p_{gd}^t - x_{id}^t)$, which means that the movement of other particles is realized through information sharing among particles, so the consistency with other particles is sought.

The basic flowchart of the standard PSO algorithm is shown in Figure 1. The basic algorithm steps are described as follows.

Step 1. Initialize particles in the related population.

Step 2. Calculate the individual extreme pBest and the global extreme gBest.

Step 3. Evaluate each particle that is to calculate the fitness value of each particle. If the current position of the particle is better than the individual extreme, the position is updated to the individual extreme. If the current position of the particle is better than the global extreme, the position is updated to the global extreme.

Step 4. Calculate the updated function fitness values.

Step 5. Update the positions and velocities of the particles according to (18) and (19).

Step 6. Judge whether the termination condition is met. If the adopted termination condition is met, output the found optimal solution, otherwise return to Step 3.

3.2. Inertia Weight Adjustment Strategy. It can be seen from (18) of PSO algorithm that inertia weight is the key factor affecting the local searching ability and the global searching ability of PSO algorithm. They can be balanced to some extent by adjusting the magnitude of the inertia weight. Most of the linear adjustment strategies of inertia weights may be divided into two kinds: linear descending strategy and linear differential descending strategies.

(1) Linear Descending Inertia Weight. The linear descending adjusting strategy of inertia weight was firstly proposed by Y. Shi and R. C. Eberhart [28], which can be described as follows:

$$w(t) = w_{start} - \frac{w_{start} - w_{end}}{t_{max}} \times t \quad (20)$$

where w_{start} is the initial value of the inertia weight, w_{end} is the value of the inertia weight at the end of the iteration, t is the current number of iterations, and t_{max} is the maximum number of iterations.

This method improves the ability of local search and global search and accelerates the convergence speed, but it easily makes the algorithm fall into the local optimum.

(2) Linear Differential Descending Inertia Weight. This method is proposed to overcome the limitations of typical linear descending strategy [29]. The linear differential descending strategy of inertia weight is described as

$$\frac{dw(t)}{d(t)} = \frac{2(w_{start} - w_{end})}{t_{max}^2} \times t \quad (21)$$

$$w(t) = w_{start} - \frac{(w_{start} - w_{end})}{t_{max}^2} \times t^2 \quad (22)$$

This method not only improves the global search ability and accelerates the convergence speed but also weakens the limitations of the typical descending strategy to some extent.

The nonlinear inertia weighting adjusting strategy is an improvement of the linear strategy, which avoids the phenomenon such that the algorithm is difficult to jump out once it enters the local extreme points. The nonlinear inertia weighting adjusting strategy mainly includes many methods,

such as the incremental-decremented inertia weight, the nonlinear descending inertia weight with threshold, and the nonlinear descending inertia weight with control factor.

(1) Incremental-Decremental Inertia Weight. In order to overcome the defects of the descending strategies, an incremental-decremented inertia weight adjusting strategy described in (23) was proposed [30].

$$w(t) = \begin{cases} 1 \times \frac{t}{t_{max}} + 0.4 \\ -1 \times \frac{t}{t_{max}} + 1.4 \end{cases} \quad (23)$$

After analysis, this algorithm has fast convergence speed in the early stage and strong local search ability in the late stage, and it also overcomes some shortcomings and improves the performance of the algorithm at the same time.

(2) Nonlinear Descending Inertia Weight with Threshold. A nonlinear descending inertia weight with threshold was proposed based on the typical descending strategy [31], which can be described as

$$w(t) = w_i - \left(\frac{t-1}{T_0-1} \right)^\lambda (w_i - w_j) \quad (24)$$

where λ , w_j , w_i , and T_0 are related parameters. This algorithm has a good effect on the accuracy, convergence speed, and stability of solutions for the low-dimensional problems.

(3) Nonlinear Descending Inertia Weight with Control Factor. The nonlinear descending inertia weight adjusting strategy with control factor [32] is described as follows:

$$w(t) = (w_{start} - w_{end} - d_1) \frac{1}{1 + d_2 t / t_{max}} \quad (25)$$

4. Parameter Tuning Method of PID Controller Based on PSO Algorithm

In the modern industrial field, the PID controller is widely used due to its advantages, such as simple structure, good robustness, and high reliability. The traditional PID parameter optimization methods include the experience trial and error method, the critical proportional method, the attenuation curve method, and the dynamic characteristic method. These traditional optimization methods are not only complicated in process but also prone to producing large overshoot. The PSO algorithm is widely used in the parameter design of PID controllers because of its simple idea, easy implementation, and fewer parameters to be adjusted [33].

The basic principle of the standard PID controller is that the control quantity $u(t)$ is formed according to the linear combination of proportional-integral-derivative of the deviation $e(t)$ between the set value and the actual value. Then the control quantity $u(t)$ is used to control the control object. The law of the continuous PID controller is described as

$$u(t) = K_P \left(e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right) \quad (26)$$

where K_p is the proportional coefficient, T_I is the integral coefficient, and T_D is the differential coefficient.

The digital PID controller is generally used to improve the reliability of modern motor control system. Therefore, the discrete PID controller, also named as the positional PID controller, can be described as

$$u(k) = K_p \left[e(k) + \frac{T}{T_I} \sum_{j=0}^k e(j) + \frac{T_D}{T} (e(k) - e(k-1)) \right] \quad (27)$$

where K_I is the integral coefficient, K_D is the differential coefficient, T is the sampling period, and $e(k)$ and $e(k-1)$ are the deviation values at the k th and $(k-1)$ th sampling time.

The most common type of the brushless DC motor control system is the proportional-integral form, because the differential element can effectively reduce overshoot and reduce the maximum dynamic deviation, but it also easily makes the system affected by high frequency interference at the same time. The main study of this paper is the combination form of proportional-integral too. In some motor digital controller, the PID controller shown in (27) is easy to generate large errors, and the poor dynamic performance, the incremental PID controller can be used to overcome these shortcomings. According to the recursive principle, the incremental PID controller based on (27) can be expressed as

$$\Delta u(k) = K_p (e(k) - e(k-1)) + K_I e(k) + K_D (e(k) - 2e(k-1) + e(k-2)) \quad (28)$$

The PID controller achieves the specified performance of the system by adjusting three parameters K_p , K_i , and K_d . From the perspective of optimization, it is to find the optimal solution in the variable space of three parameters K_p , K_i , and K_d to make the system reach the optimal state. The structure of the proposed PSO-PID controller is shown in Figure 2.

The key problem of the PID parameter optimization tuning method based on PSO algorithm is how to solve the parameter coding and the selection of fitness function. Firstly, it is assumed that there is a population P and there are a total of S particles in the population P ; the position vector of each particle consists of three control parameters K_p , K_i , and K_d , and the dimension of the particle position vector is $D=3$. Then the population can be represented by a matrix of $S \times D$ in

$$P(S, D) = \begin{bmatrix} K_p^1 & K_i^1 & K_d^1 \\ K_p^2 & K_i^2 & K_d^2 \\ \vdots & \vdots & \vdots \\ K_p^S & K_i^S & K_d^S \end{bmatrix} \quad (29)$$

Due to the diversity of the system, the range of each parameter is formulated according to the needs and actual conditions; the initial population is also randomly generated within the range of values. In order to obtain the dynamic characteristic of satisfactory transient process, the time integral performance index of the error absolute value is used as

FIGURE 2: Structure of the proposed PSO-PID controller.

TABLE 1: Parameters of PSO algorithm.

Parameter	Value
Learning factor c_1 and c_2	2.05
The number of particles	20
The size of the particles dim	2
The number of iterations $steps$	120

the minimum objective function of the parameter selection. To prevent the excessive control energy, the squared term of the control input is added to the objective function. The adopted objective function is described as follows:

$$F = \int_0^\infty (w_1 |e(t)| + w_2 u^2(t)) dt \quad (30)$$

where $e(t)$ is the system error, $u(t)$ is the controller output, and w_1 and w_2 are the weight values.

5. Simulation Experiment and Result Analysis

5.1. Parameter Settings

(1) *Motor Parameter Initialization.* The calibration power $P_n = 4e3$, standard voltage $U_n = 280$, standard current $I_n = 17.5$, angular velocity $W_n = (1980 \times P_i)/30$, armature resistance $R_n = 1.69$, armature reactance $L_n = 6.6e - 3$, rated torque $T_n = 19.3$, and the frequency of converter switch is 4000. K_i , K_w and K_u are defined as

$$K_i = \frac{1}{I_n} \quad (31)$$

$$K_w = \frac{1}{W_n} \quad (32)$$

$$K_u = \frac{1}{U_n} \quad (33)$$

(2) *Initialization of PSO Algorithm.* The parameters of PSO algorithm are listed in Table 1. In the simulation experiments, five different inertia weight adjusting strategies are adopted to verify the effect of the PSO algorithm for the PI controller of the brushless DC motor.

FIGURE 3: Waveform of electric motor input.

FIGURE 4: Output current waveform of electric motor.

FIGURE 5: Output speed waveform of electric motor.

FIGURE 6: Output voltage waveform of electric motor.

5.2. Simulation Results. The simulation experiments are carried out to obtain the diagrams of current (I), voltage (U), and angular velocity (W). The parameters K_i and K_p of the brushless DC motor and the global optimal solution (Gbest) are obtained based on PSO algorithm. The simulation results under different inertia weights for PSO algorithm are compared. The input waveform is shown in Figure 3.

(1) *Linear Descending Inertia Weight.* When the typical linear descending strategy is used on the inertia weight, the simulated output waveforms of the current, speed, and voltage of the motor are shown in Figures 4–6 through optimization of the PSO algorithm.

When the typical linear descending strategy of inertia weight and the input waveform shown in Figure 3 are adopted, the optimized results of K_i and K_p are 152.865 and 21.521, and the optimal performance is 224.9267. With the increasing number of iterations, the changes of the parameters K_p , K_i and the optimal value are shown in Figures 7–9.

(2) *Linear Differential Descending Inertia Weight.* When adopting the linear differential descending strategy inertia weight and the same input waveform, the simulated output waveforms of the current, speed, and voltage of the motor

FIGURE 7: Variation of K_p with the number of iterations. FIGURE 8: Variation of K_i with the number of iterations.

FIGURE 9: Variation of the optimal value with the number of iterations.

FIGURE 10: Output current waveform of electric motor.

FIGURE 11: Output speed waveform of electric motor.

FIGURE 12: Output voltage waveform of electric motor.

are shown in Figures 10–12. When the differential descending strategy of inertia weight and the input waveform shown in Figure 3 are adopted, the optimized results of K_i and K_p are 228.784 and 27.766, and the optimal performance is 234.9818. With the increasing number of iterations, the changes of the parameters K_p , K_i and the optimal value are shown in Figures 13–15.

FIGURE 13: Variation of K_p with the number of iterations. FIGURE 14: Variation of K_i with the number of iterations.

FIGURE 15: Variation of the optimal value with the number of iterations.

FIGURE 16: Output current waveform of electric motor.

(3) *Incremental-Decremented Inertia Weight.* When the incremental-decremented inertia weight is used, the simulated output waveforms of the current, speed, and voltage of the motor are shown in Figures 16–18 through optimization of the PSO algorithm.

When the incremental-decremented inertia weight and the input waveform shown in Figure 3 are adopted, the

FIGURE 17: Output speed waveform of electric motor.

FIGURE 18: Output voltage waveform of electric motor.

FIGURE 19: Variation of K_p with the number of iterations. FIGURE 20: Variation of K_i with the number of iterations.

FIGURE 21: Variation of the optimal value with the number of iterations.

optimized results of K_i and K_p are 395.768 and 45.034, and the optimal performance is 278.6788. With the increasing number of iterations, the changes of the parameters K_p , K_i and the optimal value are shown in Figures 19–21.

(4) *Nonlinear Descending Inertia Weight with Threshold.* When the nonlinear descending inertia weight with threshold is used, the simulated output waveforms of the current, speed, and voltage of the motor are shown in Figures 22–24 through optimization of the PSO algorithm.

FIGURE 22: Output current waveform of electric motor.

FIGURE 23: Output speed waveform of electric motor.

FIGURE 24: Output voltage waveform of electric motor.

FIGURE 25: Variation of K_p with the number of iterations. FIGURE 26: Variation of K_i with the number of iterations.

When the nonlinear descending inertia weight with threshold and the input waveform shown in Figure 3 are adopted, the optimized results of K_i and K_p are 152.865 and 21.931, and the optimal performance is 224.8972. With the increasing number of iterations, the changes of the parameters K_p , K_i and the optimal value are shown in Figures 25–27.

(5) *Nonlinear Descending Inertia Weight with Control Factor.* When the nonlinear descending inertia weight with control factor is used, the simulated output waveforms of the current, speed, and voltage of the motor are shown in Figures 28–30 through optimization of the PSO algorithm.

When the nonlinear descending inertia weight with control factor and the input waveform shown in Figure 3 are adopted, the optimized results of K_i and K_p are 316.710 and 49.273, and the optimal performance is 199.6130. With

FIGURE 27: Variation of the optimal value with the number of iterations.

FIGURE 28: Output current waveform of electric motor.

FIGURE 29: Output speed waveform of electric motor.

FIGURE 30: Output voltage waveform of electric motor.

FIGURE 31: Variation of K_p with the number of iterations. FIGURE 32: Variation of K_i with the number of iterations.

the increasing number of iterations, the changes of the parameters K_p , K_i and the optimal value are shown in Figures 31–33.

Figure 34 shows the optimal values obtained under different inertia weight adjustment strategies. In the case that different requirements are set, the adjustment strategy of

TABLE 2: Performance comparison under different inertia weights adjustment strategies.

Inertia weight adjustment strategies	K_i	K_p	Optimal performance
Linear descending inertia weight	152.865	21.521	224.9267
Linear differential descending inertia weight	151.533	21.544	234.9818
Incremental-decremented inertia weight	395.768	45.034	278.6788
Nonlinear descending inertia weight with threshold	151.562	21.931	224.8972
Nonlinear descending inertia weight with control factor	316.710	49.273	199.6130

FIGURE 33: Variation of the optimal value with the number of iterations.

FIGURE 34: Optimal values under different inertia weight adjustment strategies.

the most suitable inertia weight can be compared to make the system optimal. When adopting the same input, the output current, speed, and voltage of the motor are obtained through optimization of the PSO algorithm. By using different adjusting strategy of inertia weights, different K_i , K_p and the optimal performance are listed in Table 2, which clearly shows the effectiveness of the adopted adjustment strategies of inertia weight on PSO algorithm.

In the simulation experiments, a parameter optimization method of PI controller based on PSO algorithm with five inertia weight adjustment strategies is applied on the speed regulation of the established brushless DC motor model. It can be seen from the simulation results (Figures 4–34 and Table 2) that the linear descending inertia weight has better optimization performance than linear differential descending inertia weight. The nonlinear descending inertia weight with control factor has best optimization performance in these five inertia weight adjustment strategies. On the other hand, the nonlinear inertia weights have better optimization performance than the linear inertia weights.

6. Conclusions

In this paper, an optimization method of PI controller parameters of speed regulation for brushless DC motor based on PSO algorithm with variable inertia weights is proposed. Inertia weight is a very important parameter affecting PSO algorithm's local development ability and global search ability. The local search and the global search of the PSO algorithm can be balanced by adjusting the magnitude of the inertia weight to some extent. By properly adjusting the inertia weight, the optimal performance of the PSO algorithm on the tuning of PI controller parameters can be improved, the convergence speed can be accelerated, and more accurate optimal solution is obtained. On the other hand, other metaheuristic techniques, such as ant colony optimization (ACO) algorithm, bat algorithm (BA), chicken swarm optimization (CSO) algorithm, and firefly algorithm (FA), will be explored in the optimization of PI controller parameters of brushless DC motor in our future work.

Data Availability

There are no data to be available for this paper.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Wei Xie participated in the draft writing and critical revision of this paper. Jie-Sheng Wang participated in the concept, design, and interpretation and commented on the manuscript. Hai-Bo Wang participated in the data collection, analysis, and algorithm simulation.

Acknowledgments

This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. 2017FWDF10) and the Project by Liaoning Provincial Natural Science Foundation of China (Grant No. 20180550700).

References

- [1] A. Rubaai and P. Young, "Hardware/software implementation of fuzzy-neural-network self-learning control methods for brushless DC motor drives," *IEEE Transactions on Industry Applications*, vol. 52, no. 1, pp. 414–424, 2016.
- [2] W. Chen, Y. Liu, X. Li et al., "A novel method of reducing commutation torque ripple for brushless DC motor based on Cuk converter," *IEEE Transactions on Power Electronics*, vol. 32, no. 7, pp. 5497–5508, 2017.
- [3] C. Kiree, D. Kumpanya, S. Tunyasirirut, and D. Puangdownreong, "PSO-based optimal PI(D) controller design for brushless DC motor speed control with back EMF detection," *Journal of Electrical Engineering & Technology*, vol. 11, no. 3, pp. 715–723, 2016.
- [4] A. Sabir and M. Kassas, "A novel and simple hybrid fuzzy/PI controller for brushless DC motor drives," *Automatika*, vol. 56, no. 4, pp. 424–435, 2015.
- [5] C. Yu Jin, K. H. Ryu, S. W. Sung et al., "PID auto-tuning using new model reduction method and explicit PID tuning rule for a fractional order plus time delay model," *Journal of Process Control*, vol. 24, no. 1, pp. 113–128, 2014.
- [6] K. J. Åström and T. Hägglund, "The future of PID control," *Control Engineering Practice*, vol. 9, no. 11, pp. 1163–1175, 2001.
- [7] B. K. Sarkar, P. Mandal, R. Saha et al., "GA-optimized feedforward-PID tracking control for a rugged electrohydraulic system design," *ISA Transactions*, vol. 52, no. 6, pp. 853–861, 2013.
- [8] G.-Q. Zeng, J. Chen, M.-R. Chen et al., "Design of multivariable PID controllers using real-coded population-based extremal optimization," *Neurocomputing*, vol. 151, no. 3, pp. 1343–1353, 2015.
- [9] J. Zhang and S. Yang, "An incremental-PID-controlled particle swarm optimization algorithm for EEG-data-based estimation of operator functional state," *Biomedical Signal Processing and Control*, vol. 14, pp. 272–284, 2014.
- [10] T. Kumbasar and H. Hagsras, "Big bang-big crunch optimization based interval type-2 fuzzy PID cascade controller design strategy," *Information Sciences*, vol. 282, pp. 277–295, 2014.
- [11] A. Rajasekhar, R. Kumar Jatoth, and A. Abraham, "Design of intelligent PID/PI^λD^μ speed controller for chopper fed DC motor drive using opposition based artificial bee colony algorithm," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 13–32, 2014.
- [12] S. A. Taher, M. Hajiakbari Fini, and S. Falahati Aliabadi, "Fractional order PID controller design for LFC in electric power systems using imperialist competitive algorithm," *Ain Shams Engineering Journal*, vol. 5, no. 1, pp. 121–135, 2014.
- [13] R. K. Sahu, S. Panda, and U. K. Rout, "DE optimized parallel 2-DOF PID controller for load frequency control of power system with governor dead-band nonlinearity," *International Journal of Electrical Power & Energy Systems*, vol. 49, no. 1, pp. 19–33, 2013.
- [14] F. Cao and W. Wang, "Harmony search based particle swarm optimisation approach for optimal PID control in electrosag remelting process," *International Journal of Modelling, Identification and Control*, vol. 15, no. 1, pp. 20–27, 2012.
- [15] J. Kennedy, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, vol. 4, no. 8, pp. 1942–1948, Perth, Australia, 2011.
- [16] Y. Tang and X. Guan, "Parameter estimation for time-delay chaotic system by particle swarm optimization," *Chaos, Solitons & Fractals*, vol. 40, no. 3, pp. 1391–1398, 2009.
- [17] B. Gordan, D. J. Armaghani, M. Hajihassani, and M. Monjezi, "Prediction of seismic slope stability through combination of particle swarm optimization and neural network," *Engineering with Computers*, vol. 32, no. 1, pp. 85–97, 2015.
- [18] B. Yao, B. Yu, P. Hu et al., "An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot," *Annals of Operations Research*, vol. 242, no. 2, pp. 303–320, 2014.
- [19] R. P. Singh, V. Mukherjee, and S. P. Ghoshal, "Particle swarm optimization with an aging leader and challengers algorithm for the solution of optimal power flow problem," *Applied Soft Computing*, vol. 40, pp. 161–177, 2016.
- [20] Z. Chen, R. Xiong, J. Cao et al., "Particle swarm optimization-based optimal power management of plug-in hybrid electric vehicles considering uncertain driving conditions," *Energy*, vol. 96, pp. 197–208, 2016.
- [21] S. Chatterjee, S. Sarkar, S. Hore, N. Dey, A. S. Ashour, and V. E. Balas, "Particle swarm optimization trained neural network for structural failure prediction of multistoried RC buildings," *Neural Computing and Applications*, vol. 28, no. 8, pp. 2005–2016, 2017.
- [22] D. Tang, M. Dai, M. A. Salido, and A. Giret, "Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization," *Computers in Industry*, vol. 81, pp. 82–95, 2016.
- [23] S.-M. Chen and Z. C. Huang, "Multiattribute decision making based on interval-valued intuitionistic fuzzy values and particle swarm optimization techniques," *Information Sciences*, vol. 397, 2017.
- [24] M. A. Rahman, S. Anwar, and A. Izadian, "Electrochemical model parameter identification of a lithium-ion battery using particle swarm optimization method," *Journal of Power Sources*, vol. 307, pp. 86–97, 2016.

- [25] Y. Li, X. Bai, L. Jiao, and Y. Xue, "Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation," *Applied Soft Computing*, vol. 56, pp. 345–356, 2017.
- [26] K. Jagatheesan, V. E. Balas, A. S. Ashour et al., "Particle swarm optimisation-based parameters optimisation of PID controller for load frequency control of multi-area reheat thermal power systems," *International Journal of Advanced Intelligence Paradigms*, vol. 9, no. 5-6, pp. 464–489, 2017.
- [27] K. Jagatheesan, B. Anand, N. Dey, and M. A. Ebrahim, "Design of proportional-integral-derivative controller using stochastic particle swarm optimization technique for single-area AGC including SMES and RFB units," in *Proceedings of the Second International Conference on Computer and Communication Technologies*, pp. 299–309, Springer, New Delhi, India, 2016.
- [28] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation and IEEE World Congress on Computational Intelligence*, pp. 69–73, Anchorage, Alaska, USA, May 1998.
- [29] J.-x. Hu and J.-c. Zeng, "A particle swarm optimization model with stochastic inertia weight," *Computer Simulation*, vol. 23, no. 8, pp. 164–166, 2006.
- [30] H.-m. Cui and Q.-b. Zhu, "Convergence analysis and parameter selection in particle swarm optimization," *Computer Engineering and Applications*, vol. 43, no. 23, pp. 89–91, 2007.
- [31] L. Wang and X.-k. Wang, "Modified particle swarm optimizer using non-linear inertia weight," *Computer Engineering and Applications*, vol. 43, no. 4, pp. 47–48, 2007.
- [32] S.-l. Shui-li Chen, G.-r. Cia, G. W.-z. et al., "Study on the nonlinear strategy of acceleration coefficient in particle swarm optimization PSO algorithm," *Journal of Yangtze University*, vol. 4, no. 4, pp. 1–4, 2007.
- [33] J. Kallannan, A. Baskaran, N. Dey, and A. S. Ashour, *Bio-Inspired Algorithms in PID Controller Optimization*, CRC Press, 2018.