Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: 0005-1144 (Print) 1848-3380 (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

A PI controller optimized with modified differential evolution algorithm for speed control of BLDC motor

Huang Jigang, Fang Hui & Wang Jie

To cite this article: Huang Jigang, Fang Hui & Wang Jie (2019) A PI controller optimized with modified differential evolution algorithm for speed control of BLDC motor, Automatika, 60:2, 135-148, DOI: 10.1080/00051144.2019.1596014

To link to this article: https://doi.org/10.1080/00051144.2019.1596014

AUTOMATIKA
2019, VOL. 60, NO. 2, 135–148
https://doi.org/10.1080/00051144.2019.1596014

REGULAR PAPER

OPEN ACCESS Check for updates

A PI controller optimized with modified differential evolution algorithm for speed control of BLDC motor

Huang Jigang, Fang Hui and Wang Jie

School of Manufacturing Science and Engineering, Sichuan University, Chengdu, People's Republic of China

ABSTRACT
In this paper, a proportional integral (PI) controller that optimized with the modified different evolutional (DE) algorithm is proposed for speed control of brushless direct-current (BLDC) motor. The parameters of PI controller are tuned by the modified DE algorithm which based on adaptive mutation factor, multivariable fitness function and the starting rule for the modified algorithm. The performances of proposed controller, the conventional PI controller and the PI controller optimized with standard DE controller (PI-SDE controller) are investigated and compared in simulation. Also, the proposed controller is compared with other optimization controller in this study. The simulation results and the experimental verification show that the proposed controller leads to the smaller overshoot, less setting time and rising time compared to other controllers in this study. The results also show that the proposed controller can accelerate the response speed of BLDC motor, strengthen the robustness and guarantee motor runs smoothly as well as precisely. This work indicates the distinguished performance of proposed controller for the speed control of BLDC motor.

ARTICLE HISTORY
Received 16 July 2018
Accepted 13 March 2019

KEYWORDS
PI controller; DE algorithm; speed control; BLDC motor

1. Introduction

Brushless direct-current (BLDC) motors have famous speed versus torque characteristics, high power density, long operating life, noiseless operation and excellent control properties [1,2]. Due to these merits, BLDC motors have been widely used in the industries and different control methods were presented to control the speed of BLDC motor.

Proportional–integral–derivative (PID) controller is well known for its simplicity, clear functionality, and effectiveness, which has been commonly adopted in industrial control systems [3]. Generally, PID controller is used for the speed closed-loop control of BLDC motor in practical application. However, BLDC motor is a multivariable and strong coupling nonlinear system, the conventional PID controller using in this system always have deficiencies. It's so sensitive to the system uncertainties that the control performance can be seriously degraded under parameter variations [4,5]. Moreover, the conventional PID controller is also difficult to adjust the high precision and rapid speed system dynamic performance as well as static performance [6].

Recently, some modern intelligence algorithms have been studied to control motors and always showed better performance than the conventional PID controller. K. Premkumar et al. (2016) [7] studied a fuzzy proportional derivative (PD) controller optimized with bat algorithm for speed control of BLDC motor. In their work, the bat algorithm optimized fuzzy PD controller was compared with particle swarm, cuckoo search algorithm optimized fuzzy PID controller and fuzzy PID controller. It was proven that bat optimized fuzzy PD controller has superior performance than the other controllers considered. S. Bouallegue et al. (2012) [8] presented a PID-type fuzzy logic controller tuning based on particle swarm optimization (PSO). The performance of PSO-tuned PID-type fuzzy logic controller was compared with that of the fuzzy PID controller tuning based on genetic algorithm optimization. And the simulation results showed that their proposed controller has better performance for the speed control of DC motor in term of robustness and efficiency. Bharat Bhushan et al. (2011) [9] introduced a bacterial foraging algorithm (BFA) based speed controller for DC motor, and the performance of BFA based controller was studied in MATLAB. It was shown that the BFA based speed controller works effectively for tracking the desired trajectory with less computational time. A hybrid controller for the control of permanent magnet synchronous motor which includes neural network PID and conventional PID controller was researched by Bingyang Luo et al. (2017) [10]. The high stability and reliability of conventional PID were combined with the strong adaptive ability and robustness of neural network. As their results revealed, the speed response of the hybrid controller was faster than that of the

CONTACT Fang Hui hjigang@foxmail.com, jfh@scu.edu.cn; wangjie@scu.edu.cn School of Manufacturing Science and Engineering, Sichuan University, Chengdu, Sichuan Province 610065, People's Republic of China

single control strategy. Moreover. the recovery ability and robustness on the condition of sudden disturbance of hybrid controller were also stronger.

Previous works have demonstrated the prospect of intelligence algorithms for the motor control. In this paper, a proportional integral (PI) controller optimized with modified differential evolution (DE) algorithm was proposed for the better speed control of BLDC motor. The modified DE algorithm is designed with more varied individuals and bigger search space for the minimization problems, which enables the proposed controller more accurately and effectively for the speed control. The control parameters of PI controller are tuned by modified DE algorithm based on the speed error, the overshoot and the setting time of system. To investigate the performance of proposed controller, detailed simulations were conducted on different conditions. Meanwhile, the simulation results of proposed controller were compared with that of the conventional PI controller and standard DE based PI controller. The paper is organized as follows: section 2 introduces the model of BLDC motor, the standard DE algorithm and the proposed modified DE algorithm are described detailly in section 3, section 4 presents the proposed controller, the performance of the proposed controller is investigated in MATLAB/Simulink and the results are discussed in section 5, section 6 shows the experimental verification, then section 7 concludes the paper.

2. Model of BLDC

2.1. The control system of BLDC motor

The working principle of BLDC motor could be described as follows: electronic commutation controller receives the position signal of rotor by position sensors, and control the inverter bridge to change the power-up state of windings [11,12]. The continuous commutating will produce a rotating magnet at the stator windings and then drive the rotor to move. The BLDC motor in this paper has three stator windings connecting in star model and a permanent magnet rotor, driving by the model of three-phase six-step and two-phase breakover with 120 electrical angle. Figure 1 shows the BLDC motor control system [13].

2.2. Mathematical model of BLDC motor

From motor voltage equation $U = \mathrm{Ri} + L\frac{di}{dt} + E$, the voltage equation of BLDC motor can be characterized by the following equation.

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = \begin{bmatrix} R_a & 0 & 0 \\ 0 & R_b & 0 \\ 0 & 0 & R_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \times + \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} E_a \\ E_b \\ E_c \end{bmatrix} \quad (1)$$

where, u_a, u_b, u_c were the stator phase voltage, R_a, R_b, R_c were the stators phase resistance, i_a, i_b, i_c were stators phase current, L_{aa}, L_{bb}, L_{cc} were winding self-induction of three-phase stators, L_{ab}, L_{bc}, L_{ba}, L_{ac}, L_{ca}, L_{cb} were mutual inductance between three-phase stators winding, E_a, E_b, E_c were the back electromotive force (EMF) of three-phase stators.

As the resistance of each stator is equal, and the structure of three stator windings is totally symmetrical, so there are equations:

$$R_a = R_b = R_c = R \quad (2)$$

$$L_{aa} = L_{bb} = L_{cc} = L \quad (3)$$

$$L_{ab} = L_{bc} = L_{ba} = L_{ac} = L_{ca} = L_{cb} = M \quad (4)$$

For a three-phase star winding motor, there is an equation:

$$i_a + i_b + i_c = 0 \quad (5)$$

From (2) $\sim$ (5), the voltage equation (1) can be rewritten as:

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L-M & 0 & 0 \\ 0 & L-M & 0 \\ 0 & 0 & L-M \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} E_a \\ E_b \\ E_c \end{bmatrix} \quad (6)$$

The torque equation of BLDC motor is described as:

$$T_e = \frac{e_a i_a + e_b i_b + e_c i_c}{\omega} \quad (7)$$

where, T_e is the electromagnetic torque, ω is the motor mechanical angular velocity.

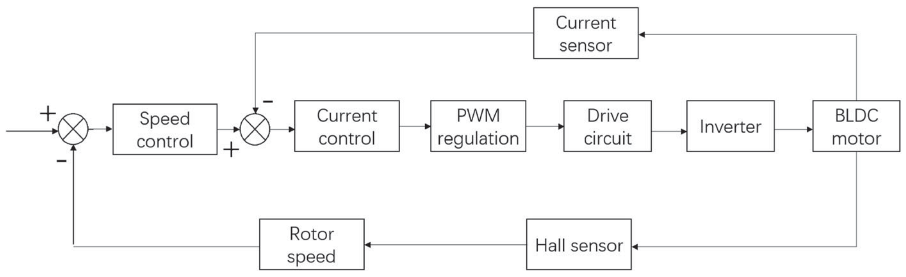


Figure 1. Block diagram of BLDC motor control system.

The motion equation of BLDC motor can be written as:

$$T_e - T_L = J\frac{d\omega}{dt} + B_v\omega \tag{8}$$

where, T_L is load torque, J is rotation inertia, B_v is viscous frictional coefficient.

3. Differential evolution algorithm

3.1. Standard DE algorithm

DE algorithm is an evolutionary algorithm based on the differential mutation, greedy principle and real-coded. The excellent properties of differential evolution algorithm, such as compact structure, high convergence characteristic, simple yet powerful and straightforward features, make it attractive for parameter optimization [13]. Similar to other evolution algorithms, the DE algorithm includes three operations, which are mutation, crossover and selection, for the global optimization over continuous spaces with only a few parameters. Furthermore, DE algorithm iterates generation by generation until the termination conditions have been met by operating on each individual [14–16]. Figure 2 shows the procedure of DE algorithm and the specific descriptions are as follows.

The standard DE starts with initializing a population of target individuals, and each individual represents a potential solution. Suppose the range of population is $[x_{i,\min}, x_{i,\max}]$, then the individuals are generated randomly within this range. Equation (9) shows the individuals and the initialization method is described in Equation (10).

$$P_G = \{X_{1,G}, X_{2,G}, \ldots, X_{j,G}\}, \quad j = 1, 2, \ldots, M;$$
$$X_{j,G} = (X_{1,j,G}, X_{2,j,G}, \ldots, X_{N,j,G}) \tag{9}$$

where, P_G is the population, G represents the times of iteration, and M is number of individuals. $X_{j,G}$ is an individual of N-dimensional vector and j represents the index of the individual.

$$X_{ij,0} = X_{i,\min} + \text{rand}(0,1)(X_{i,\max} - X_{i,\min}),$$
$$i = 1, 2, \ldots, N \tag{10}$$

where, $X_{ij,0}$ is the element of initial individual, rand (0,1) is a random number between 0 and 1.

The fitness of individual that calculated by individual fitness function is used to estimate the quality of individuals [17]. Usually, it is also a termination condition for DE algorithm. The DE algorithm ends when the fitness is satisfied, or proceeds to the next iteration.

The DE algorithm performs the mutation according to Equation (11). As Equation (11) shows, a mutant individual is generated based on three individuals that randomly selected from previous population, and the weighted difference is added between two individuals [18]. If an element of a mutant individual goes off the search space, then the value of this element is regenerated. Here, it should be noted that the individuals must be difference from each other, thus the number of individuals (M) is at least four [19,20].

$$V_{j,G+1} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}),$$
$$j \neq r1 \neq r2 \neq r3 \tag{11}$$

where, $V_{j,G+1}$ is the mutant individual. $X_{r1,G}$, $X_{r2j,G}$, and $X_{r3,G}$ are three individuals selected from population randomly. F is a real parameter, called mutation factor.

After the mutation, the operation of crossover is designed to enhance the diversity of population. For DE algorithm, Binomial crossover is the commonly used scheme, which can be expressed as Equation (12) [21,22]. The crossover is performed on each of elements whenever a randomly generated number between 0 and 1 is less than or equal to a pre-fixed parameter. From the crossover operation of the standard DE algorithm, it can be seen that a mutant individual is used to enhance the perturbation of a target individual to avoid premature convergence to non-global local optima selection [23–26].

$$U_{j,G+1} = (U_{1,j,G}, U_{2,j,G}, \ldots, U_{N,j,G});$$
$$V_{j,G+1} = (V_{1,j,G}, V_{2,j,G}, \ldots, V_{N,j,G});$$
$$U_{ij,G+1} = \begin{cases} V_{ij,G+1}, & \text{rand}\,(0,1) \leq CR \text{ or } i = \text{rand}\,(i) \\ X_{ij,G}, & \text{rand}\,(0,1)\ CR \end{cases} \tag{12}$$

where, $U_{j,G+1}$ is the trial individual after operation of crossover. $U_{ij,G+1}$, $V_{ij,G+1}$ are elements in individuals of $U_{j,G+1}$ and $V_{j},_{G+1}$ respectively. rand (0,1) is a random number between 0 and 1, rand(i) is a random number ranging from 1 to N, and CR is a per-fixed parameter, called the crossover factor.

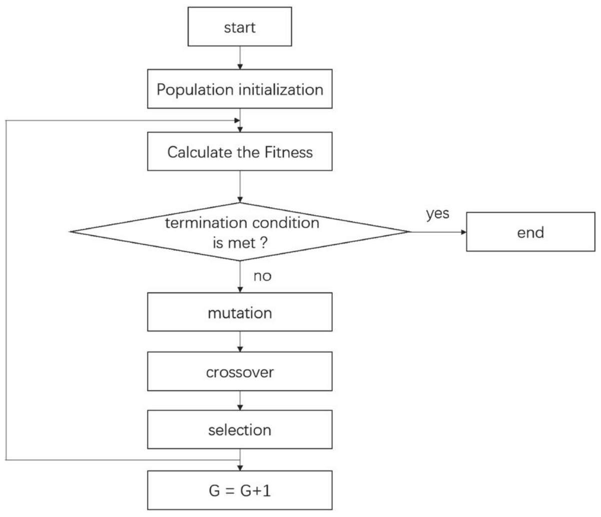


Figure 2. Procedure of DE algorithm.

To decide whether or not the trial individual $U_{j,G}$ should be a member of the next generation, a one to one greedy selection strategy is adapted in DE algorithm. The fitness value of each individual is evaluated for the selection. If the trial individual $U_{j,G+1}$ yields a better fitness value than $X_{j,G}$, then $U_{j,G+1}$ is selected for the next generation. Otherwise, the source individual is retained. For minimization problems, the selection scheme is designed as follows:

$$X_{j,G+1} = \begin{cases} U_{j,G+1}, & f\left(U_{j,G+1}\right) f\left(X_{j,G}\right), \\ X_{j,G}, & f\left(U_{j,G+1}\right) \geq f\left(X_{j,G}\right) \end{cases} \tag{13}$$

where, $f(U_{j,G+1})$, $f(X_{j,G})$ are the fitness functions of $U_{j,G+1}$ and $X_{j,G}$ respectively.

DE algorithm takes the operations of mutation, crossover, selection and calculate the fitness value of each individual repeatedly. However, the algorithm would not end until the termination condition is met or the iterations is reached [27].

3.2. *Modified differential evolution algorithm*

For standard DE algorithm, the mutation factor (F) and crossover factor (CR) are invariant constants that chosen by previous experience. Specifically, the range of F is from 0 to 2, and CR is in the range of 0 and 1 generally [28–30]. As F and CR are the primary control parameters of DE algorithm, it has already been reported that the suitable values of these parameters may result DE algorithm enjoy a superior performance. S. Das et al. (2005) [31] suggested decreasing F from 1.0 to 0.5 linearly, which encourages the individuals to sample diverse zones of the search space during the early stages. The mutation factor decayed during later stages helps DE algorithm to explore a relatively small space in which the suspected global optimum lies. J. Zhang et al. (2009) [32] proposed a JADE algorithm. In their algorithm, the F for each individual is sampled from a Cauchy probability distribution with a mean μ_F and the CR is sampled from a normal distribution with a mean μ_{CR}. However, the means μ_{F} and μ_{CR} are updated in each generation based on their previous records of success. S.M. Elsayed et al. (2013) [33] presented a variant of DE with an adaptive parameter control scheme. They defined two sets of values for F and CR, and each trial individual is generated by combination of the parameters randomly. Then, if a trial individual is successfully selected for the next generation, the value of corresponding parameters is increased by one. While the top half of the performing combinations is selected, their corresponding parameters are set to 0, and the others are discarded.

In this paper, the modified DE algorithm is originated from the study of Aronb Ghosh. Aronb Ghosh et al. (2011) supported a fitness-base adaption scheme for the parameters of F and CR [34]. Equation (14) shows the scheme for F.

$$\begin{aligned} F_1 &= F_{\max} * \left(\frac{\Delta f_j}{\lambda + \Delta f_j}\right), \\ F_2 &= F_{\max} * (1 - e^{-\Delta f_j}) \\ F_j &= \max(F_1, F_2) \end{aligned} \tag{14}$$

where, $\lambda = (\Delta f_j/10) + 10^{-14}$, $\Delta f_j = |f(X_j) – f(X_{\text{best}})|$, f is the fitness function and $F_{\max} = 0.8$.

It's clear from Equation (14) that the F for each individual is related to its fitness value. Aronb Ghosh et al. also compared their modified DE algorithm with other DE algorithms, and the results indicated that the DE algorithm with their scheme for parameters had superior performance for most of the tested problem. Here, we propose a fitness-base adaption scheme for F, which is shown in Equation (15). While the CR is designed as a constant value of 0.6.

$$F_j = F_{\max} * (1 - e^{-f(X_j)}) \tag{15}$$

where, $f(\text{X}_j)$ is the fitness function of X_j, and $F_{\max} = 0.8$.

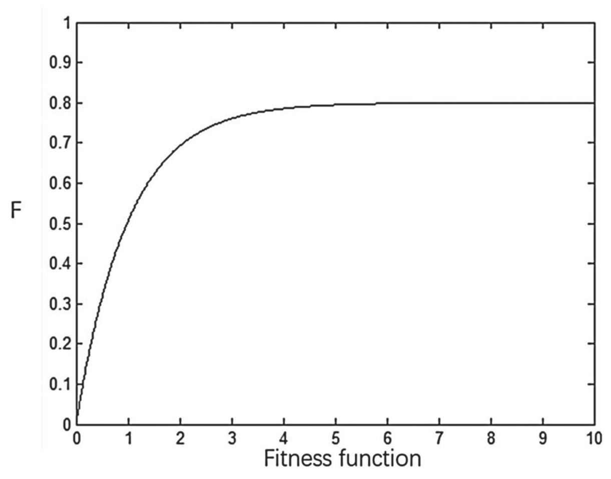


Figure 3. variation of F with $f(X_j)$ varying in scale of 0–10.

Figure 3 shows the variation of F with the fitness function. Obviously, the proposed scheme for F is applicable for the minimization problems. When $f(\text{X}_j)$ is great, the great value of F enhances the diversity of individuals and enlarges the search space. However, the little value of F guarantees the DE algorithm to search a small space in which the global optimum lies. Moreover, it should be note that the proposed modified DE algorithm is designed to optimize the conventional PI controller for the speed control of BLDC motor.

4. Design of the proposed controller

4.1. *The conventional PI controller*

The PI controller is a simplified PID controller, which removes the derivative control. The letter P, I stand for proportion and integral. Figure 4 shows the structure

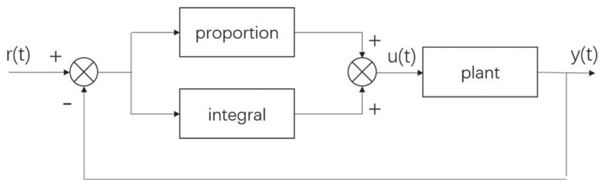


Figure 4. Structure of conventional PI controller.

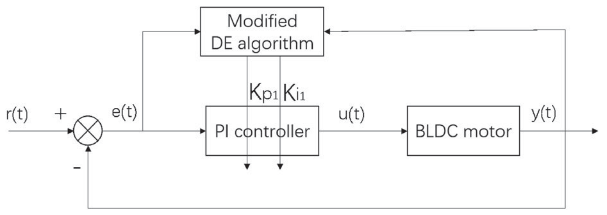


Figure 5. Structure of the proposed controller.

of the convention PID controller, and the output of a conventional PID controller can be described as the following equation:

$$u(t) = \mathrm{Kp} \cdot e(t) + \mathrm{Ki} \cdot \int_0^t e(t)\mathrm{d}t, \tag{16}$$

where, u(t) is the output of PID controller, Kp is proportional gain, Ki is integral gain, e(t) is the speed error.

In modern control systems, for the sake of reliability, the continuous PID controller should not be applied directly and a discrete processing is needed [35]. Then the digital conventional PID controller can be expressed as:

$$u(k) = \mathrm{Kp} \cdot e(k) + \mathrm{Ki} \cdot \sum_{j=0}^{k} e(j), \tag{17}$$

where, e(k) is the error at the time of k.

4.2. The proposed controller

As mentioned above, the proposed controller is a conventional PI controller optimized with modified DE algorithm. Figure 5 describes the structure of proposed controller. The inputs of modified DE algorithm are the speed error (“e”) and the speed of BLDC motor (“s”), while the outputs are variables of Kp_1 and Ki_1. The parameters of PI controller, which are Kp and Ki, are tuned by the modified DE algorithm according to the real-time system.

The fitness function of DE algorithm is the criterion to evaluate the quality of individuals during evolution. A suitable fitness function is the basis of modified DE algorithm applied to control systems. However, it is difficult to meet the requirements that the response, stability, and robustness of the speed control for BLDC motor according to the speed error [36]. To optimize the dynamic and static characteristics of BLDC motor, a fitness function that contains multiple system parameters needs to be established. Here, the fitness function for the proposed modified DE algorithm is designed and shown as follows:

$$f = \omega_1|\sigma| + \omega_2 t_s + \omega_3 \sum_{k=1}^{n} |e(k)|, \tag{18}$$

where, $\omega_1, \omega_2, \omega_3$ are weight coefficients that ω_1 is 10.0, ω_2 is 5.0 and ω_3 is 0.1, σ is the overshoot, t_s is the setting time and $e(k)$ is the speed error at the moment of k.

The overshoot, settling time and error of the speed are contained in this fitness function. While the requirements for the speed response, stability and accuracy could be changed by adjusting the value of weights. The weight for overshoot tends to be increased if a small overshoot is required, and when the rapid speed response is required, the weight for setting time should be augmented. The value of fitness function is expected to be minimum by the modified DE algorithm, which leads to good speed control of BLDC motor. However, the difference of parent and offspring is subtle after numbers of iterations and the individual is close to the optimal, so the proposed modified DE algorithm is also optimized with the rule as follows:

If $f > f_U$, then the modified DE algorithm works;
If $-0.12 \leq f' < 0$ for the first time and after five more iterations, or G = 50, then the modified DE algorithm ends.

where, f_U is the threshold of fitness function that f_U is 1.2 and f' is the derivative of fitness function.

5. Simulation results

In this section, the performances of three controllers, which are the proposed controller, PI controller optimized with the standard DE algorithm (PI-SDE controller) and the conventional PI controller, for the speed control of BLDC motor are researched respectively and compared. The simulations are conducted on different operating conditions such as sudden/gradual change in load, start with different load, varying set speed in MATLAB/Simulink. Here, the proportional gain and integral gain of the conventional PI controller are 0.011 and 28, respectively. Table 1 shows the parameters of the proposed modified DE algorithm and the standard DE algorithm.

Figure 6 shows the Simulink model of BLDC motor. In the simulation model, the proposed controller was used to control the speed of BLDC motor, the load torque was controlled by a Lookup Table function module, and the stator current, rotor speed and electromagnetic torque were observed by using Scope modules. The Simulink model of proposed controller is shown

Table 1. The parameters of DE algorithm.

DE algorithm/parameters	Range of individuals	*M*	*G*	*F*	CR	*f*
Modified	$0 \leq \text{Kp} \leq 1$	30	50	$0.8 * (1 - e^{-f(X_j)})$	0.6	$f = \omega_1\sigma + \omega_2 t_s + \omega_3 \sum_{k=1}^{n} \lvert e(k)\rvert$
	$0 \leq \text{Ki} \leq 35$					
Standard	$0 \leq \text{Kp} \leq 1$	30	50	0.6	0.6	$f = \sum_{k=1}^{n} \lvert e(k)\rvert$
	$0 \leq \text{Ki} \leq 35$					

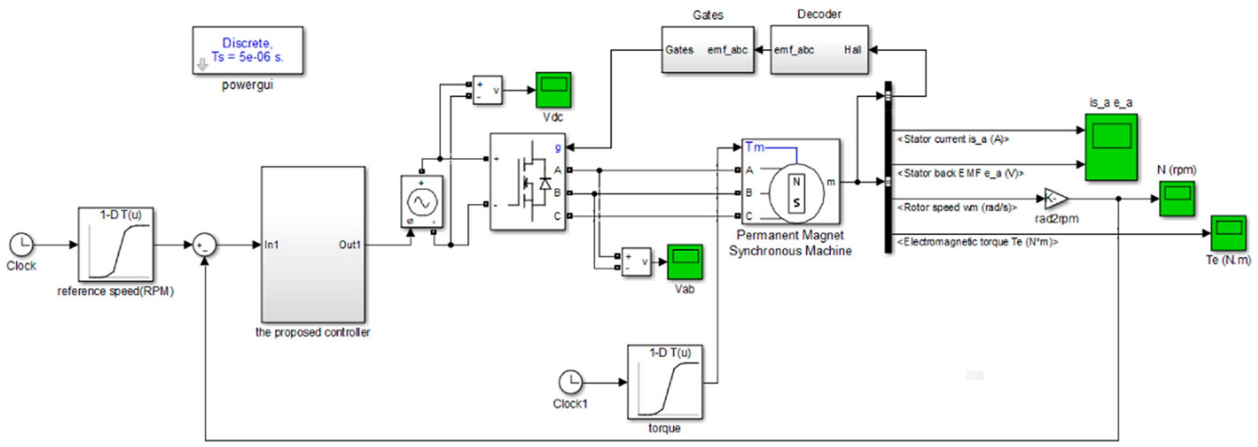


Figure 6. Simulink model of BLDC motor.

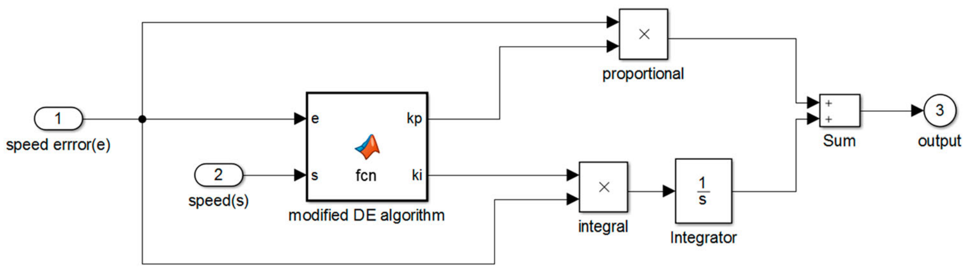


Figure 7. Simulink model of the proposed controller.

in Figure 7. To improve the optimization efficiency of DE algorithm, the space of individuals should be determined properly. Thus, the range of individuals is selected according to previous experience, which the ranges of Kp and Ki are from 0 to 1, and from 0 to 35 respectively.

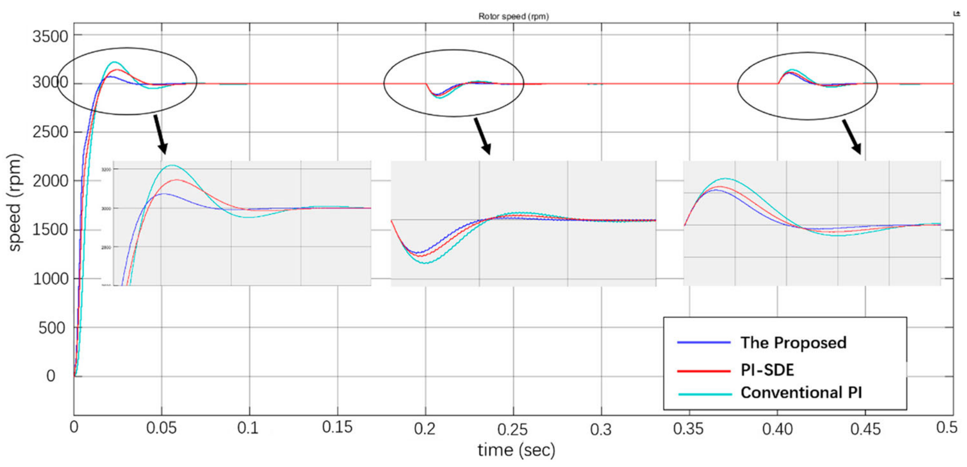


Figure 8. Speed response curve of BLDC motor of sudden change in load.

Table 2. Results of speed response under sudden change in load, (a) the conventional PI controller, (b) PI controller optimized with the standard DE algorithm controller, (c) the proposed controller.

Time (s)/results	Overshoot (%)	Settling time (s)	Rising time (s)	Steady state error (rpm)
		(a)		
0 ~ 0.2	7.37	0.055	0.015	0
0.2 ~ 0.4	−4.87	0.043	–	2
0.4 ~ ~ 0.5	4.83	0.044	–	3
		(b)		
0 ~ 0.2	6.83	0.045	0.016	0
0.2 ~ 0.4	−4.03	0.039	–	0
0.4 ~ ~ 0.5	4.03	0.040	–	0
		(c)		
0 ~ 0.2	2.47	0.031	0.014	0
0.2 ~ 0.4	−3.70	0.020	–	0
0.4 ~ ~ 0.5	3.67	0.035	–	0

5.1. Sudden change in load

The performance of proposed controller, PI-SDE controller and the conventional PI controller for the speed control of BLDC motor are investigated respectively and compared on the condition of sudden application and removal of load. The set speed of BLDC motor is 3000 rpm, during the time from 0 to 0.2 s, the motor is running with no load, then a step load of 5 Nm is applied at the time of 0.2 s, and is suddenly removed at the time of 0.4 s. Figure 8 shows the speed response curve and the simulation results are exhibited in Table 2.

During the starting of BLDC motor, the overshoot is 2.47%, the setting time is 0.031 s and the rising time is 0.014 s with the proposed controller. While the overshoot is 6.83% with the setting time of 0.045 s as well as the rising time of 0.016 s as the PI-SDE controller using for the speed control during the starting. And for that of the conventional PI controller, the overshoot is 7.37%, the setting time is 0.055 s with the rising time of 0.015 s. When the sudden load is applied, the overshoot is −3.70% and the setting time is 0.020 s with the proposed controller. However, the overshoots of PI-SDE controller and conventional PI controller are −4.03%, −4.87% with the setting times are 0.039 s and 0.043 s respectively. Similarly, when the load is removed suddenly, the proposed controller also shows good performance with a small overshoot of 3.67% and a short setting time of 0.035 s. While the PI-SDE controller leads to a greater overshoot of 4.03% and a longer setting time of 0.040 s, thus the conventional PI controller leads to the greatest overshoot of 4.83% as well as the longest setting time of 0.044 s. Meanwhile, the proposed controller and the PI-SDE controller also eliminate the steady state error which is 2 rpm with the conventional PI controller when the load is suddenly changed. From the results, it’s clear that the proposed controller has best performance for the speed control of BLDC motor on the condition of sudden change in load, and the performance of PI-SDE controller is better than that of the conventional PI controller.

Table 3. Results of speed response under gradual change in load, (a) the conventional PI controller, (b) PI controller optimized with the standard DE algorithm, (c) the proposed controller.

Time (s)/results	Overshoot (%)	Settling time (s)	Rising time (s)	Steady state error (rpm)
		(a)		
0 ~ 0.1	7.37	0.055	0.015	0
0.1 ~ 0.2	−0.97	–	–	–
0.2 ~ 0.3	0	–	–	2
0.3 ~ 0.4	0.87	–	–	–
0.4 ~ 0.5	0	–	–	2
		(b)		
0 ~ 0.1	6.83	0.045	0.016	0
0.1 ~ 0.2	−0.47	–	–	–
0.2 ~ 0.3	0	–	–	0
0.3 ~ 0.4	0.50	–	–	–
0.4 ~ 0.5	0	–	–	0
		(c)		
0 ~ 0.1	2.47	0.031	0.014	0
0.1 ~ 0.2	−0.43	–	–	–
0.2 ~ 0.3	0	–	–	0
0.3 ~ 0.4	0.43	–	–	–
0.4 ~ 0.5	0	–	–	0

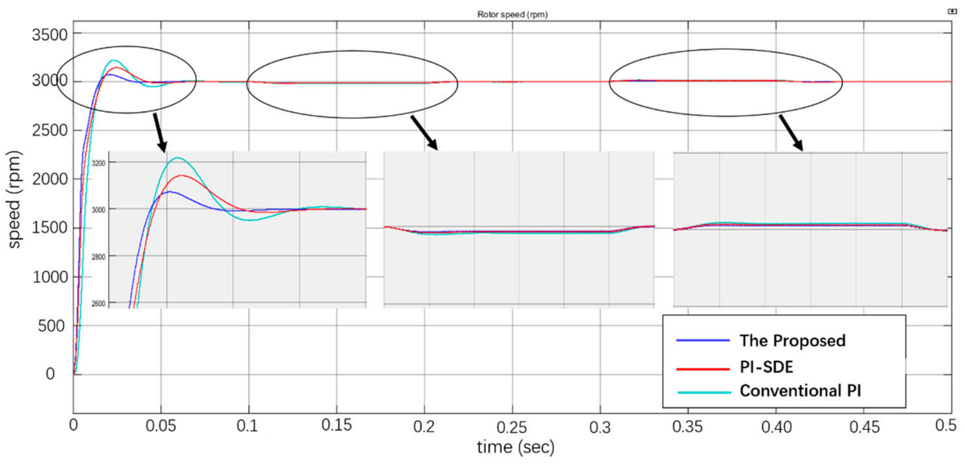
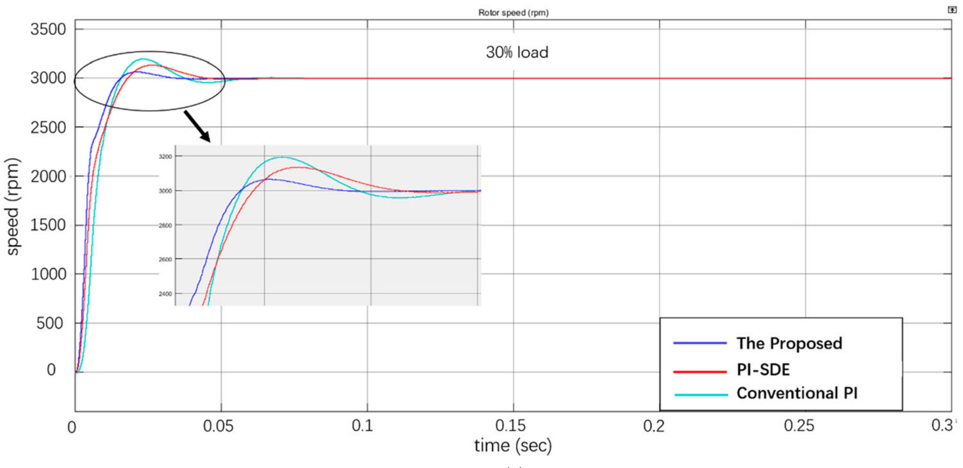


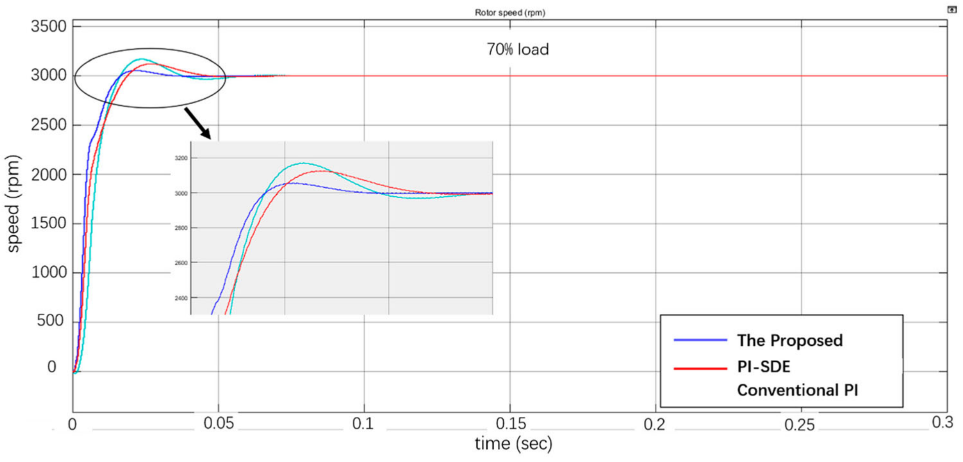
Figure 9. Speed response curve of BLDC motor of gradual application and removal of load.

5.2. Gradual change in load

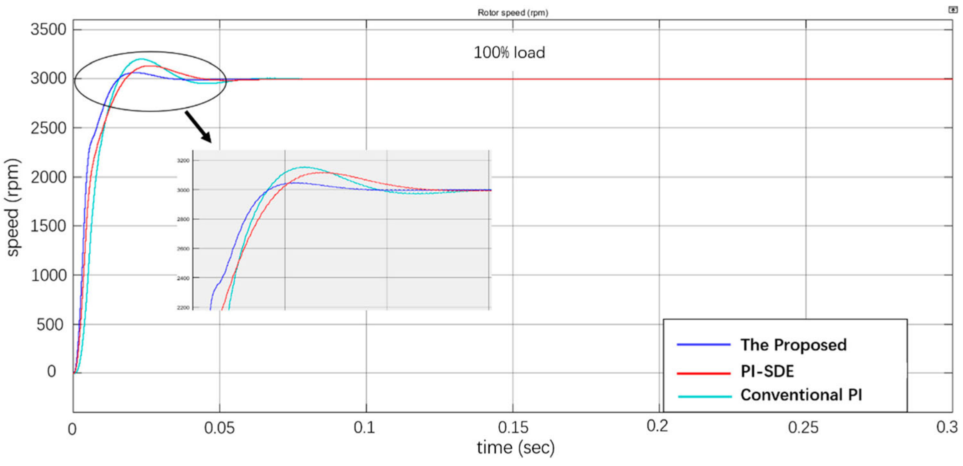
Figure 9 shows the speed response of BLDC motor with the proposed controller, PI controller optimized with the standard DE algorithm and the conventional PI controller on the condition of gradual change in load. From 0 to 0.1 s, the motor runs without any load, then from 0.1 s to 0.2 s, the load linearly increases from 0 Nm to 3 Nm, and maintains during the time from 0.2 s to 0.3 s. However, the load linearly decreases from 3 Nm back to 0 Nm start at the time of 0.3 s and end at the time of 0.4 s. Table 3 shows the results of three controllers for the speed control of BLDC motor. From the results, the proposed controller shows the better performance than the other two controllers for the speed control of BLDC motor. Specifically, the overshoot is −0.43% with the proposed controller while the load increases linearly, and −0.47% with the PI-SDE controller as well as −0.97% for that of the conventional PI controller. As



(a)



(b)



(c)

Figure 10. Speed curves of BLDC motor starts on different loads, (a)30% of full load, (b)70% of full load, (c) 100% of full load,

Table 4. Results of BLDC motor starts on different loads, (a) 0% of full load, (b) 30% of full load, (c) 70% of full load, (d) 100% of full load.

Controllers/results	Overshoot (%)	Settling time (s)	Rising time (s)	Steady state error (rpm)
		(a)		
Conventional PI	7.37	0.055	0.015	0
PI-SDE	6.83	0.045	0.016	0
The proposed	2.47	0.031	0.014	0
		(b)		
Conventional PI	6.50%	0.056	0.016	2
PI-SDE	4.53%	0.045	0.018	0
The proposed	2.23%	0.032	0.015	0
		(c)		
Conventional PI	5.67%	0.058	0.016	2
PI-SDE	4.53%	0.046	0.018	0
The proposed	2.23%	0.032	0.015	0
		(d)		
Conventional PI	5.03%	0.058	0.017	2
PI-SDE	4.53%	0.047	0.019	0
The proposed	2.23%	0.033	0.016	0

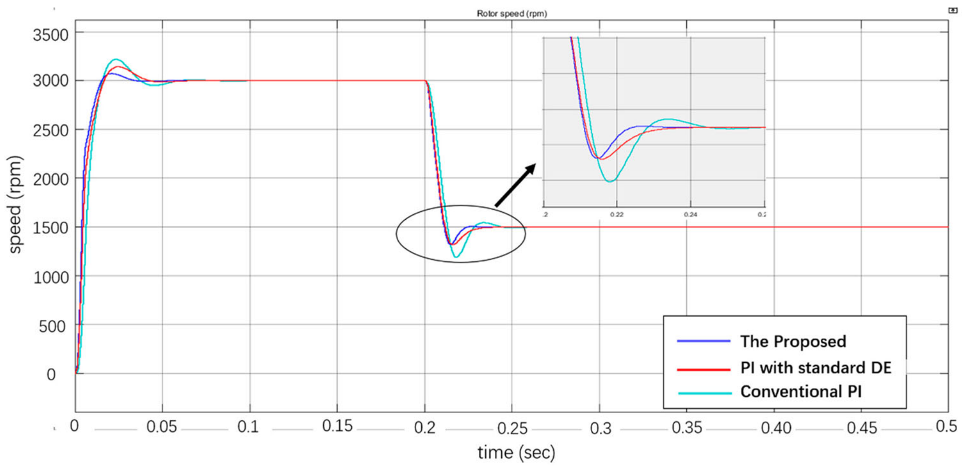
the load decreases, the results are also expected, that the overshoot is smallest with the proposed controller and the conventional PI controller leads to the greatest overshoot.

5.3. Starting characteristics

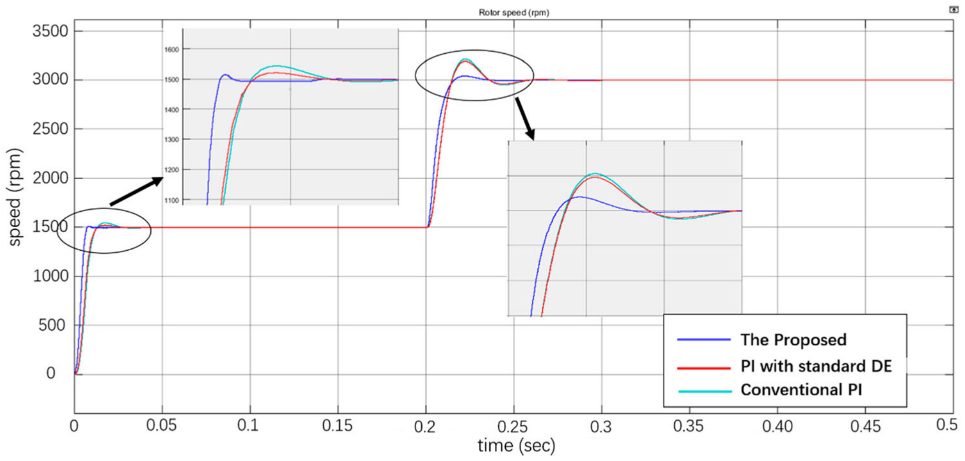
The starting characteristics of BLDC motor with the two controllers is investigated in Simulink by starting on four different loads, which are 0%, 30%, 70% and 100% of full load. The simulation results are shown in Figure 10 and Table 4. The simulations of BLDC motor start without load with these controllers have been researched in section 5.1. Here, the results are

Table 5. Results of speed response under the speed changes suddenly from 3000 rpm to 1500 rpm, (a) with the conventional PI controller, (b) with PI-SDE controller, (c) with the proposed controller.

Time (s)/results	Overshoot (%)	Settling time (s)	Rising time (s)	Steady state error (rpm)
		(a)		
0 ~ 0.2	7.37	0.055	0.015	0
0.2 ~ 0.5	−11.33	0.032	0.014	2
		(b)		
0 ~ 0.2	6.83	0.045	0.016	0
0.2 ~ 0.5	−7.38	0.032	0.013	2
		(c)		
0 ~ 0.2	2.47	0.031	0.014	0
0.2 ~ 0.5	−7.33	0.022	0.010	0



(a)



(b)

Figure 11. Speed response curve of BLDC motor when speed changes suddenly, (a) speed changes suddenly from 3000 rpm to 1500 rpm, (b) speed changes suddenly from 1500 rpm to 3000 rpm.

Table 6. Results of speed response under the speed changes suddenly from 1500 to 3000 rpm, (a) with the conventional PI controller, (b) with PI-SDE controller, (c) with the proposed controller.

Time (s)/results	Overshoot (%)	Settling time (s)	Rising time (s)	Steady state error (rpm)
		(a)		
0 ~ 0.2	3.47	0.048	0.015	2
0.2 ~ 0.5	3.54	0.050	0.015	2
		(b)		
0 ~ 0.2	1.67	0.044	0.014	0
0.2 ~ 0.5	3.26	0.048	0.014	0
		(c)		
0 ~ 0.2	1.65	0.022	0.012	0
0.2 ~ 0.5	1.57	0.028	0.013	0

rewritten in Table 4. The results show that the starting characteristics with proposed controller are more excellent than that of the PI-SDE controller and the conventional PI controller. For example, while the BLDC motor starts on 70% of full load, the overshoot, the setting time and the rising time are 2.23%, 0.032 s and 0.015 s respectively with the proposed controller. However, the overshoot of the PI-SDE controller is 4.53% with the setting time of 0.046 s on this condition, and the overshoot is 5.67% with the setting time of 0.058 s as the conventional PI controller is used for the speed control of BLDC motor. On the whole, the overshoot, setting time and rising time of the system with the proposed controller are all less than those of the other two controllers while BLDC motor starts on different loads. Furthermore, the overshoot and setting time of the system with the PI-SDE controller are less than that of the conventional PI controller, yet the rising time of PI-SDE controller is a litter longer. It also demonstrates that the overshoot is greater and the setting time as well as the rising time are longer as the load increases. The results indicate that the outstanding performance of proposed controller for the speed control of BLDC motor while the motor starts with different loads.

5.4. Varying set speed

Figure 11 presents the speed response of BLDC motor with different controller on the condition of sudden change in speed. There are two schemes which are designed as following:

Scheme 1. The motor starts with the speed of 3000 rpm and the speed changes suddenly from 3000 rpm to 1500 rpm at the time of 0.2 s.

Scheme 2. The motor starts with the speed of 1500 rpm and the speed suddenly increases from 1500 rpm to 3000 rpm at the time of 0.2 s.

As shown in Figure 11, the proposed controller exhibits a better performance when the speed of BLDC motor is changed in a sudden. It also can be found that the speed response of the system is faster and smoother with the proposed controller in both simulation experiments. Table 5 lists the results of simulation conducted with the Scheme 1, and the results of the other simulation were shown in Table 6.

Table 5 shows that the overshoots of both controllers increase significantly as the speed decreases from 3000 rpm to 1500 rpm suddenly, specifically, the overshoot is −11.33% with conventional PI controller, −7.38% with PI-SDE controller and −7.33% with the proposed controller. The overshoot, settling time and rising time at the starting speed of 1500 rpm are less than these at the starting speed of 3000 rpm. From Table 6, the overshoot is 3.47% at the starting speed of 1500 rpm and 3.54% when the speed suddenly changes with conventional PI controller. While the overshoot

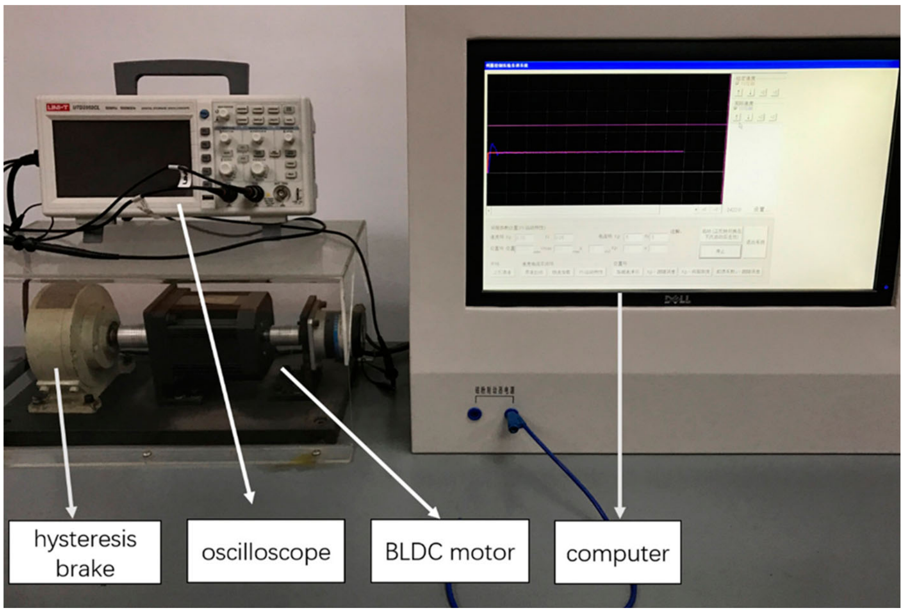
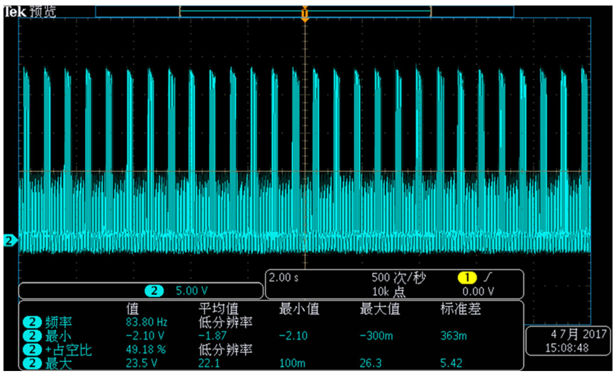


Figure 12. Experimental setup of speed control for BLDC motor.

is 1.67% at the starting and 3.26% when the speed changes. For the proposed controller, the overshoot is 1.65% at the starting speed of 1500 rpm and 1.57% on the condition of speed changes from 1500 rpm to 3000 rpm suddenly. The settling time and rising time with the proposed controller are obviously less than these with the other two controllers. For example, the setting time is 0.022 s and the rising time is 0.012 s with the proposed controller at the starting speed of 1500 rpm. However, the setting time is 0.044 s with the rising time of 0.014 s as the PI-SDE controller is used, and the setting time of 0.048 s as well as rising time of 0.015 s for the conventional PI controller. The results also demonstrate the distinguished performance of proposed controller for speed control of BLDC motor.

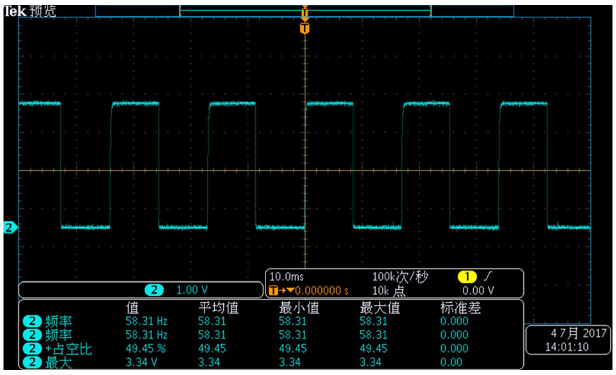
6. Experimental verification

Experiments are conducted and the results are provided in this section to verify the performance of proposed controller. As Figure 12 shows, the speed control of BLDC motor, which the rated parameters are 3000 rpm, 24 V, 210 W and 0.7 N•M respectively, is implemented based on STM32. The rotor position of BLDC motor is



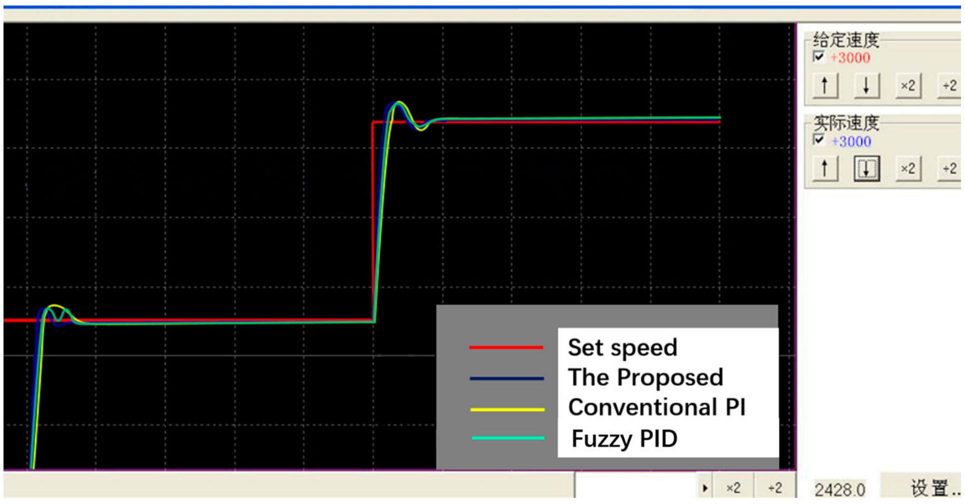
	值	平均值	最小值	最大值	标准差
2 频率	83.80 Hz	低分辨率			
2 最小	−2.10 V	−1.87	−2.10	−300m	363m
2 +占空比	49.18 %	低分辨率			
2 最大	23.5 V	22.1	100m	26.3	5.42

(a)



	值	平均值	最小值	最大值	标准差
2 频率	58.31 Hz	58.31	58.31	58.31	0.000
2 频率	58.31 Hz	58.31	58.31	58.31	0.000
2 +占空比	49.45 %	49.45	49.45	49.45	0.000
2 最大	3.34 V	3.34	3.34	3.34	0.00

(b)



(c)

Figure 13. (a) Experimental phase voltage waveform of BLDC motor, (b) output of Hall sensor, (c) speed response curve when set speed suddenly changes from 1500 rpm to 3000 rpm.

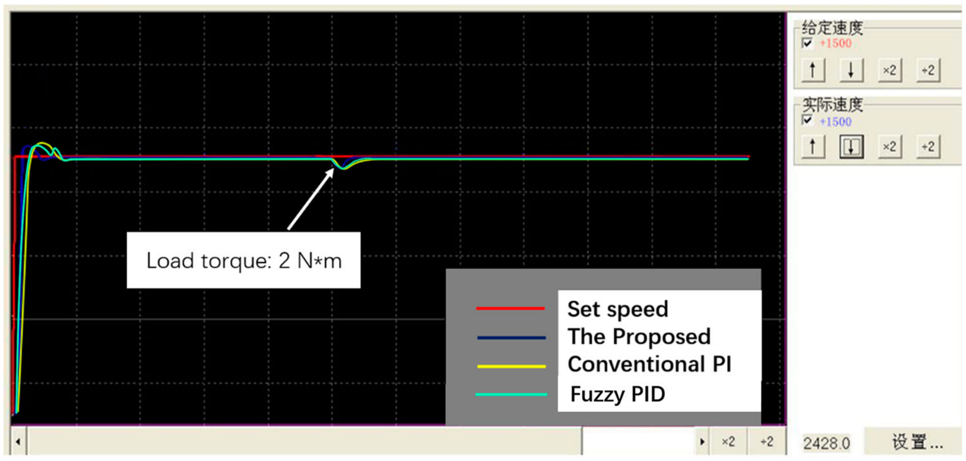


Figure 14. Speed response for the load suddenly changes.

measured by Hall sensor, and the rotor position is then converted into actual speed by derivative algorithm. The output of Hall sensor and the phase voltage waveform are detected by an oscilloscope. A hysteresis brake is connected with BLDC motor by coupling to provide the load for the tests. To prove the validity of the proposed controller, the performances of conventional PI controller and fuzzy PID controller, which is a well-known optimization controller, are taken into consideration as the comparison. The details of fuzzy PID controller are designed from R. Arulmozhiyal's work [37].

Figure 13 (a) and (b) show the phase voltage waveform and output of Hall sensor respectively at the speed of 1500 rpm, while (c) depicts the speed response curve on the condition that the BLDC motor starts at 1500 rpm and the speed suddenly change from 1500 rpm to 3000 rpm without any load. It's clear that the rotor speed with proposed controller increases to track the set speed and reach the stability more rapidly than which with the conventional PI controller as well as that with fuzzy PID controller when the set speed changes suddenly. While the performance of fuzzy PID controller is slightly better than that of conventional PI controller according to Figure 13(c). The speed response characteristic with load changes suddenly at the speed of 1500 rpm is depicted in Figure 14. when the load is applied suddenly, the speed decreases lightly and recovers quickly. Specifically, the proposed controller lead to less overshoot and shorter setting time than the other two controllers, which means the good robustness of proposed controller. The experimental results indicate the speed response with proposed controller is faster than that with fuzzy PID controller as well as the conventional PI controller. Also, it can be seen from Figure 13(c) and Figure 14 that the proposed controller leads to shorter rising time, less setting time and less overshoot that the other two controllers during the starting. While the fuzzy PID controller does not show visibly better performance than conventional PI controller, which may be due to the inappropriate fuzzy parameters. From the experimentation studies, it is proved that the proposed controller has excellent performance for the speed control of BLDC motor, which could accelerate the response speed of BLDC motor, strengthen the robustness and guarantee motor runs smoothly as well as precisely.

7. Conclusion

The paper presented a PI controller optimized with the modified DE algorithm for the speed control of BLDC motor. The modified DE algorithm is based on adaptive mutation factor, multivariable fitness function and the rule for starting in real time. Simulations for the speed control of BLDC motor were conducted on different conditions in MATLAB/Simulink, and performance of the proposed controller was compared with the conventional PI controller and the PI-SDE controller. The simulation results indicated that the proposed controller for the speed control of BLDC motor has better performance than the conventional PI controller as well as the PI controller optimized with standard DE algorithm. Furthermore, the experiments were taken to validate the performance of proposed speed control algorithm by comparing with conventional PI controller as well as Fuzzy PID controller. From the results, it was easy to reach the conclusion that the proposed controller could accelerate the response speed of the motor and strengthen the robustness of the system. However, the values of threshold of fitness function (f_U) and weight coefficients (ω_1, ω_2, ω_3) need to be set deliberately, which always need necessary tests or empirical data to determine these values. In this paper, we provided an effective speed controller for BLDC motor, and we expect this work can advise researchers to work on the speed control of BLDC motor.

Acknowledgements

The authors thank Professor Placid M Ferreira, Department of Mechanical Engineering of University of Illinois at Urbana-Champaign, for his suggestions of this article.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This paper was supported by the National Natural Science Foundation of China (General Program, No. 51175356) and Sichuan Province Science and Technology Support Program (2017GZ0092)

ORCID

Huang Jigang http://orcid.org/0000-0002-8756-2262

References

[1] Li C-L, Li W, Li F-D. Chaos induced in brushless DC motor via current time-delayed feedback. Optik. 2014;125:6589–6593.
[2] Rahideh A, Korakianitis T, Ruiz P, et al. Optimal brushless DC motor design using genetic algorithms. J Magn Magn Mater. 2010;322:3680–3687.
[3] Sharkawy AB. Genetic fuzzy self-tuning PID controllers for antilock braking systems. Eng Appl Artif Intell. 2010;23(7):1041–1052.
[4] Jung J-W, Leu VQ, Do TD, et al. Adaptive PID speed control design for permanent magnet synchronous motor drives. IEEE Trans Power Electron. 2015;30(2):900–908.
[5] Premkumar K, Manikandan BV. Fuzzy PID supervised online ANFIS based speed controller for brushless DC motor. Neurocomputing. 2015;157:76–90.
[6] Kim CH, Yang JH, Lim DG, et al. An enhanced PID controller for speed control of brushless DC motors based on convex set optimization. IFAC Workshop on Intelligent Control Systems. 2010;8(1):75–80.
[7] Premkumar K, Manikandan BV. Bat algorithm optimized fuzzy PD based speed controller for brushless direct current motor. Eng. Sci Technol Int J. 2016;19:818–840.
[8] Bouallegue S, Haggege J, Ayadi M, et al. PID-type fuzzy logic controller tuning based on particle swarm optimization. Eng Appl Artif Intell. 2012;25:484–493.
[9] Bhushan B, Singh M. Adaptive control of DC motor using bacterial foraging algorithm. Appl Soft Comput. 2011;11:4913–4920.
[10] Luo B, Chi S, Fang M, et al. Modeling and simulation of permanent magnet synchronous motor based on neural network control strategy. Advance in Materials, Machinery, Electronics I, AIP Conf. Proc.1820, 070010-1 – 070010-7, 2017.
[11] Singh B, Singh S. State of the art on permanent magnet brushless DC motor drives. Journal of Power Electronics. 2009;9(1):3–17.
[12] Krishman R. Permanent magnet synchronous and brushless DC motor Drives. Beijing: China Machine Press, 2011.
[13] Jigang H, Jie W, Hui F. An anti-windup self-tuning fuzzy PID controller for speed control of brushless DC motor. Automatika. 2017;58(3):321–335.
[14] Lu X, Tang K, Sendhoff B, et al. A new self-adaptation scheme for differential evolution. Neurocomputing. 2014;146:2–16.
[15] Storn R, Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous space. J Global Optim. 1997;11:341–359.
[16] Mohamed AW, Sabry HZ, Khorshid M. An alternative differential evolution algorithm for global optimization. J Adv Res. 2012;3:149–165.
[17] Onan A, Korukoglu S, Bulut H. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. Expert Syst Appl. 2016;62:1–16.
[18] Tsai J-T, Fang J-C, Chou J-H. Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. Comput Oper Res. 2013;40:3045–3055.
[19] dos Santos Coelho L, Pessoa MW. A tuning strategy for multivariable PI and PID controllers using differential evolution combined with chaotic Zaslavskii map. Expert Syst Appl. 2011;38:13694–13701.
[20] Ouyang P, Pano V. Comparative study of DE, PSO and GA for position domain PID controller tuning. Algorithms. 2015;8:697–711.
[21] Weber M, Tirronen V, Neri F. Scale factor inheritance mechanism in distributed differential evolution. Soft Comput. 2010;180(12):2405–2420.
[22] Das S, Mullick SS, Suganthan PN. Recent advances in differential evolution – an updated survey. Swarm Evol Comput. 2016;27:1–30.
[23] Zorarpaci E, Ozel SA. A hybrid approach of differential evolution and artificial bee colony for feature selection. Expert Syst Appl. 2016;62:91–103.
[24] Pan Q-K, Suganthan PN, Wang L, et al. A differential evolution algorithm with self-adapting strategy and control parameters. Computers and Operations Research. 2011;38:394–408.
[25] Moharam A, EI-Hosseini MA, Ali HA. Design of optimal PID controller using hybrid differential evolution and particle swarm optimization with an aging leader and challengers. Appl Soft Comput. 2016;38: 727–737.
[26] Lianghong W, Yaonan W, Shaowu Z, et al. Design of PID controller with incomplete derivation based on differential evolution algorithm. J Syst Eng Electron. 2008;19(3):578–583.
[27] Pedro JO, Dangor M, Dahunsi OA, et al. Differential evolution-based PID control of nonlinear full-car electrohydraulic suspensions. Math Probl Eng. 2013;10:1155–1168.
[28] Zheng W, Pi Y. Study of the fractional order proportional integral controller for the permanent magnet synchronous motor based on the differential evolution algorithm. ISA Trans. 2016;63:387–393.
[29] Storn R, Price K. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces[R]. Berkeley: University of California, 2006.
[30] Ali MM, Torn A. Population set-based global optimization algorithms: some modifications and numerical studies. Computers and Operations Research. 2004;31:1703–1725.
[31] Pan Q-K, Wang L, Qian B. A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems. Computers and Operations Research. 2009;36:2498–2511.
[32] Das S, Konar A, Chakraborty UK. Two improved differential evolution schemes for faster global search. ACM-SIGEVO Proceedings of GECCO, Washington DC, 2005, pp.991–998.
[33] Zhang J, Sanderson AC. JADE: adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. 2009;13(5):945–958.

[34] Elsayed SM, Sarker RA, Ray T. Differential evolution with automatic parameter configuration for solving the CEC 2013 competition on real-parameter optimization. Proceedings of the IEEE Congress on Evolutionary Computation. 2013 Cancun, Mexico, pp.1932–1937.

[35] Ghosh A, Das S, Chowdury A, et al. An improved differential evolution algorithm with fitness-based adaptation of the control parameters. Inf Sci. 2011;181:3749–3765.

[36] Gundogdu T, Komurgoz G. Self-tuning PID control of a brushless DC motor by adaptive interaction. IEEJ Trans Electr Electron Eng. 2014;9:384–390.

[37] Arulmozhiyal R, Kandiban R. Design of fuzzy PID controller for brushless DC motor. 2012 ICCCI, Jan.10–12, Coimbatore, India.