Implementation of Grasshopper Optimization Algorithm for Controlling a BLDC Motor Drive

Devendra Potnuru and Ayyarao S. L. V. Tummala

Abstract The paper implemented a recently developed grasshopper optimization algorithm for speed response improvement during transient and steady-state conditions for a BLDC motor drive. An objective function is formulated to reduce the integral square error in such a way that the gains of speed control (PID) are optimally tuned. To know the validity of the present approach, simulation experiments are conducted extensively to get the proper tuning of PID gains in MATLAB/Simulink and then the same gains can be used in off-line for the hardware implementation.

Keywords BLDC · Grasshopper optimization · Speed controller PID tuning

1 Introduction

The brushless DC motor is nowadays well known for its superiority due to its kind of the characteristics such as noise-free operation, flexible speed control, and high-speed range. However, the closed-loop speed control is necessary for some applications where in the speed should be constant irrespective of sudden changes of load and/or supply. Hence, the gain tuning of speed controller designed using PID controller is a tedious job. To reduce the time of gain tuning and improve the performance of the speed control in closed loop, many researchers developed various techniques. The Ziegler–Nichols method is a usually employed method for determining the PID gains in many applications. However, in high-performance

D. Potnuru (✉) · A. S. L. V.Tummala
GMR Instutute of Technology, Rajam 532127, India
e-mail: devendra.p@gmrit.org; devendra.p.07@gmail.com

A. S. L. V.Tummala
e-mail: ayyarao.tslv@gmrit.org

J. Nayak et al. (eds.), *Soft Computing in Data Analytics*,
Advances in Intelligent Systems and Computing 758,
https://doi.org/10.1007/978-981-13-0514-6_37

applications the accurate closed-loop speed control with good steady-state and transient behavior is very much important.

In [2], BAT algorithm is implemented for speed control of BLDC motor. Ye et al. [7] have been implanted with modified PSO and PID controller in. In [6]. Bacteria foraging and PSO optimization has been applied to speed control DC motor. Multiobjective Optimization has been applied for Controller Tuning of a TRMS Process in. Fuzzy-Neural-Network Self-Learning Control Methods for Brushless DC Motor Drives has been implemented in [4]. DE and PSO algorithms have been applied for PID controller tuning in [3].

However, tuning of PID speed controller using grasshopper algorithm for closed-loop speed controller for a BLDC motor has been not yet implemented in the existing literature. Hence, the present paper is mainly devoted to implementation of recently developed nature-inspired grasshopper optimization algorithm for tuning of the PID gains. Integral speed error is considered as objective function optimization in the present work. The extensive simulations using MATLAB/Simulink has resulted in optimum gains of PID, and they are in off-line for real-time speed control of BLDC motor in laboratory arrangement.

The paper is organized as follows: In Sect. 2, the operation of BLDC drive is described. Formulation objective function and PID control design are elaborated in Sect. 3. Later, the experimental results are described in Sect. 4 and conclusions are given in the last section.

2 The Drive Scheme of BLDC Motor

The BLDC motor drive operation mainly consists of three-phase inverter, BLDC motor, DSP controller and speed and position sensors. As the BLDC motor drive run in self-control mode, the inverter IGBTs are switched on based on rotor position. The DSP controller is responsible for generating the PWM pulses from the given input conditions. Then, the gate drivers are used to drive the output pulses of the DSP controller and turn on the IGBTs. In the present work BLDC motor is a tetra square wave type and with inbuilt hall sensors to sense the rotor position.

The block diagram for PID tuning for speed control of BLDC motor drive using grasshopper optimization algorithm is shown in Fig. 2. For more details of the drive scheme, one can refer [1].

First, the MATLAB simulation file is converted into DSP-enabled code and dump in the DSP processor (dSPACE DS 1103). The LEM makes current, and voltage sensors are used to sense the current and voltages. In the closed-loop speed control, the torque reference is generated based on the speed error. The output of the PID controller is scaled down using the motor torque constant to obtain the reference currents of the hysteresis current controller. The current controller generates the required PWM pulses to the inverter switches.

3 Grasshopper Optimization

Grasshopper is one kind of insects which damages the agriculture. The swarm behavior of the grasshopper is modeled to develop an efficient nature-inspired algorithm. The size of the grasshopper swam is very large than any other creature in nature. Most noted thing is that swarm behavior can be observed in nymph and adulthood. They jump and move like rolling and eat the vegetation in their way and become nightmare for the farmers. Later, they form very big swarm in the air and migrated over large distances [5]. Figure 1 shows the life cycle of swarm, Fig. 2 shows the primitive corrective patterns in swarm of grasshoppers, and Fig. 3 shows the primitive corrective patterns in swarm of grasshoppers.

The implementation algorithm for grasshopper optimization is given as below [5].

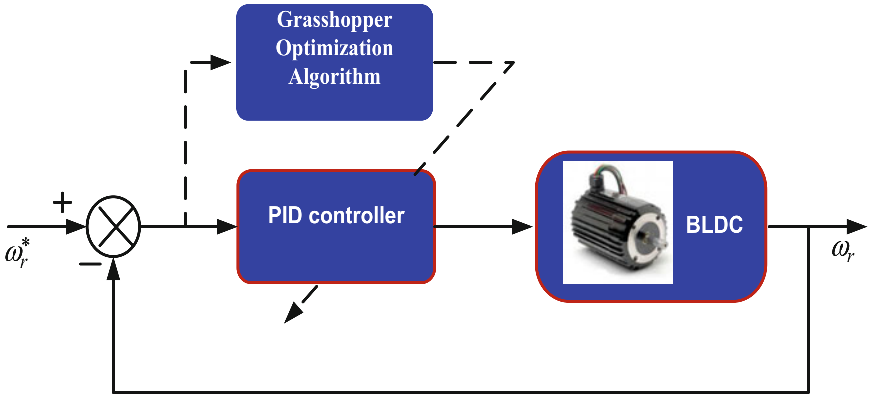

Fig. 1 Speed controller of BLDC motor using grasshopper algorithm

Fig. 2 Grasshopper and life cycle of grasshopper [5]

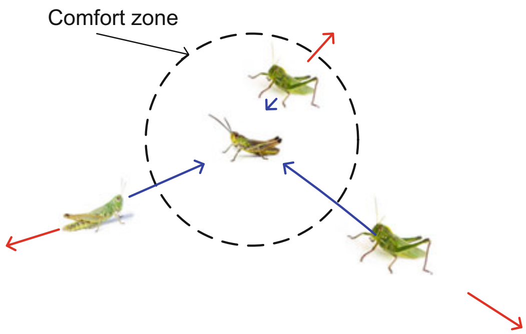

Fig. 3 Primitive corrective patterns in swarm of grasshoppers [5]

Grasshopper optimization Algorithm

Initialization

Initialize the population, Max-iteration λ, $\zeta_{\max}$, $\zeta_{\min}$
Calculate the objective function value from (ISE) for each solution
Obtain the best solution of X with minimum objective function -
-value F

While (i < λ) % main loop begins
Update the following equation (1)

$$\zeta = \zeta_{\min} - i\frac{\zeta_{\max} - \zeta_{\min}}{\lambda} \quad (1)$$

for j < population size % to obtain the new X
obtain normalized distance between grasshoppers
update the position
j=j+1
end for loop
Replace X with the new one if the objective function is less than F
i=i+1
end % end main loop

4 Result Analyses

The closed-loop speed control using grasshopper algorithm has been extensively tested for two reference speeds. Firstly, the algorithm is run for the closed-loop speed control for number of times by keeping the proper ranges for the PID gains. Then finally, the optimal gains are selected for implementation of closed loop speed control. The typical convergence characteristic of fitness function is as shown in

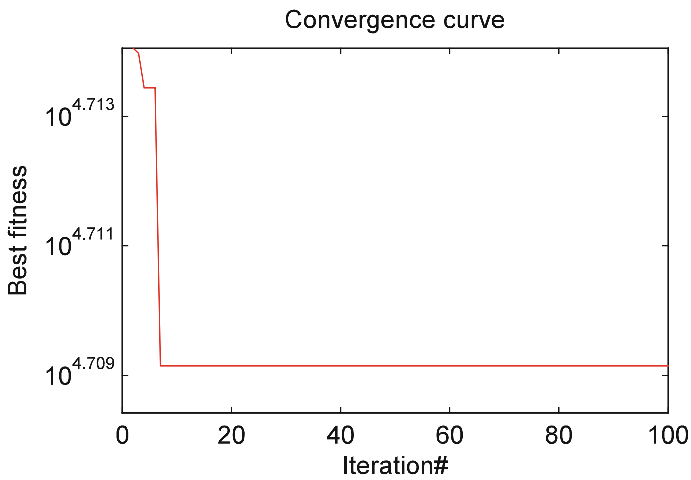

Fig. 4 Convergence curve of optimization

Fig. 4. The optimal PID gains for final selection are with kp = 23.418, ki = 13.8416, and kd = 0.001. These gains are applied to observe the dynamic performance of the drive. In first test case, reference speed consists of stepped speed command and second case is the combination of step and ramp speeds (hybrid speed reference).

Figure 5a shows the performance of the closed-loop speed control using stepped speed command, and Fig. 5b shows speed error with the stepped reference speed.

Figure 6a shows the performance of the closed loop speed control using hybrid speed reference, and corresponding speed error is shown in Fig. 6b.

One can observe that the closed-loop speed control performance is superior during transient and steady-state conditions. Speed error is very small during the steady-state conditions.

5 Conclusions

The closed-loop speed control of BLDC motor has been implemented successfully by using the PID gains obtained with the grasshopper optimization algorithm. It is observed that the closed-loop speed control performance is superior during transient and steady-state conditions. Speed error is very small during the steady-state conditions.

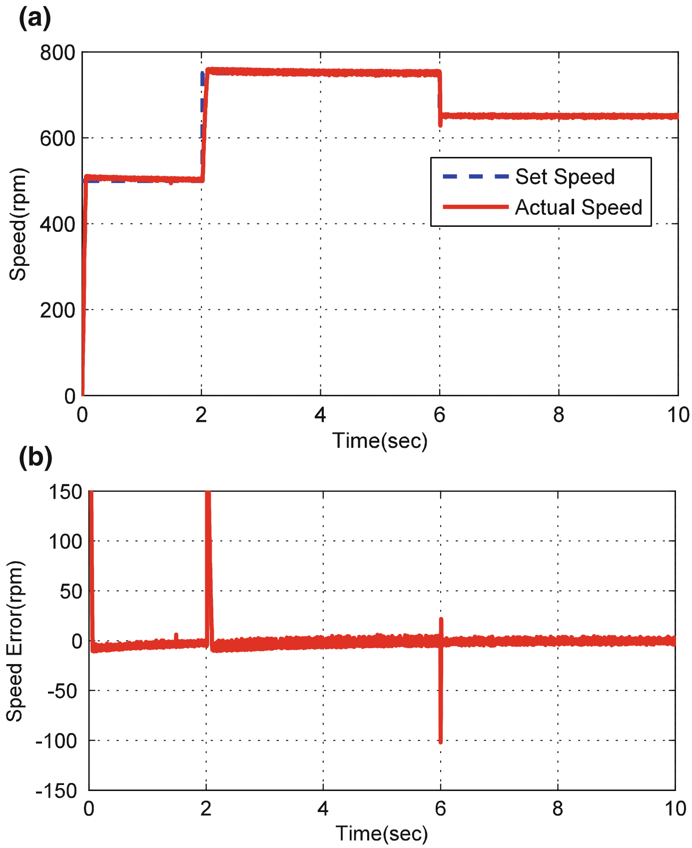

Fig. 5 **a** Closed-loop speed control using grasshopper algorithm. **b** Speed error of closed-loop speed control with stepped speed reference

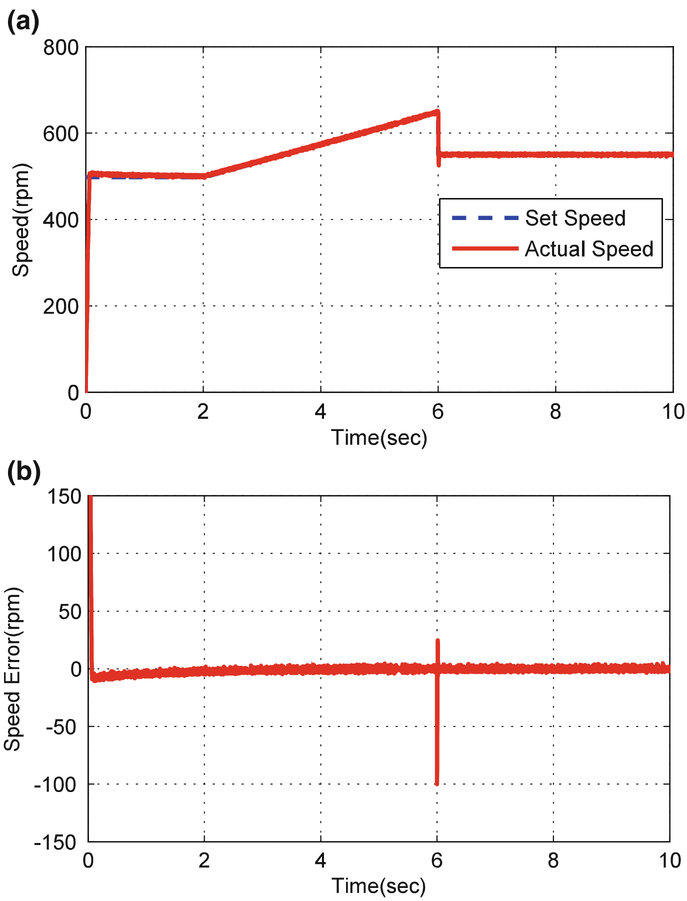

Fig. 6 **a** Closed-loop speed control hybrid step and ramp speed command. **b** Speed error of the closed-loop speed control for hybrid step and ramp speed command

References

1. Potnuru, D., et al.: Design and implementation methodology for rapid control prototyping of closed loop speed control for BLDC motor. J. Electr. Syst. Inf. Technol., 4–16 (2017)
2. Premkumar, K., Manikandan, B.V.: Bat algorithm optimized fuzzy PD based speed controller for brushless direct current motor. Eng. Sci. Technol. an Int. J. **19**(2), 818–840 (2016)
3. Puralachetty, M.M., Pamula, V.K.: Differential evolution and particle swarm optimization algorithms with two stage initialization for PID controller tuning in coupled tank liquid level system. In: ICARM 2016–2016 International Conference on Advanced Robotics and Mechatronics, pp. 507–511 (2016)
4. Rubaai, A., Young, P.: Hardware/software implementation of fuzzy-neural-network self-learning control methods for brushless DC motor drives. IEEE Trans. Ind. Appl. **52**(1), 414–424 (2016)

5. Saremi, S., et al.: Grasshopper optimisation algorithm: theory and application. Adv. Eng. Softw. **105**, 30–47 (2017)
6. Thangavelusamy, D., Ponnusamy, L.: Study and comparison of PI controller tuning techniques using bacteria foraging and bacteria foraging based particle swarm optimization. Int. J. Electr. Eng. **5**(5), 571–586 (2012)
7. Ye, Y., et al.: Position control of nonlinear hydraulic system using an improved PSO based PID controller. Mech. Syst. Signal Process. **83**, 241–259 (2017)