Hindawi
International Journal of Photoenergy
Volume 2023, Article ID 2777505, 23 pages
https://doi.org/10.1155/2023/2777505

Research Article

An Improved Tuning of PID Controller for PV Battery-Powered Brushless DC Motor Speed Regulation Using Hybrid Horse Herd Particle Swarm Optimization

A. RamaKrishnan,[1] **A. Shunmugalatha**,[2] **and K. Premkumar**[3]

[1]*AR College of Engineering and Technology, Tamil Nadu, India*
[2]*Velammal College of Engineering and Technology, Tamil Nadu, India*
[3]*Rajalakshmi Engineering College, Chennai, Tamil Nadu, India*

Correspondence should be addressed to A. RamaKrishnan; ramakrishnan452022@gmail.com

Received 28 February 2023; Revised 26 April 2023; Accepted 4 July 2023; Published 18 July 2023

Academic Editor: Debabrata Barik

Copyright © 2023 A. RamaKrishnan 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.

In this study, speed control of PV battery-powered brushless DC motor (BLDC) is controlled by novel hybrid horse herd particle swarm optimization- (HHHPSO-) tuned proportional integral derivative (PID) controller. The optimal gain parameter of the PID controller is tuned by hybrid horse herd optimization algorithm. The purpose of the newly developed HHHPSO algorithm is to enhance the performance of the classic horse herd algorithm (HHA), specifically in two different ways. In the first place, it bolsters HHA's aptitude for exploratory learning related to the ageing issue. By doing so, it is possible to circumvent the phenomenon of the local minimum stagnation. Second, it permits HHA to have a superior capability of exploitation with the assistance of hybridization through the utilisation of particle swarm optimization. This hybrid technique helps improve the rate convergences of the HHA method. The time domain-based performance indices were considered as an objective function such as addition of integral of squared speed error, integral of squared current error, and integral of squared electromagnetic torque error for finding the optimal gain values for the PID controller using HHHPSO. The proposed HHHPSO-tuned PID controller for PV battery-powered BLDC motor is tested for various working conditions such as constant speed conditions, varying speed conditions, and varying load conditions and also compared with state-of-the-art method. The proposed method has quick rise time around 20-21 msec, quick settling time around 35-39 msec, zero steady-state error, and zero overshoot than state-of-the-art optimization method. The proposed control techniques were also tested in hardware to confirm the suitability for real-time applications.

1. Introduction

Packaging equipment, compressors, riders, elevators, conveyor belts, mills, fans, pumps, and many more have all included BLDC motors in recent years [1]. Other uses for BLDC motors include those in electric vehicles, medical devices, robotics, car industry, and aerospace. The BLDC motor is superior to conventional induction motors in several respects, including its versatility in speed regulation, its power density, its torque-to-weight ratio, its low maintenance requirements, and its efficiency [2]. Many applications need the usage of a three-phase BLDC motor in order to lessen their carbon footprint by decreasing their need for fuel and the complexity of their control systems. A BLDC motor is essentially the same as a permanent magnet synchronous motor (PMSM). For the purpose of generating a trapezoidal back EMF [3], permanent magnets are installed on the rotor and windings are placed on the stator. The BLDC's functioning is structured, and its key variable is the rotor's orientation.

The BLDC motor's control may be divided into two distinct categories: sensorless control and sensor-based control. In sensor-based control, the Hall effect sensors are used to determine where the rotor of the BLDC motor is located

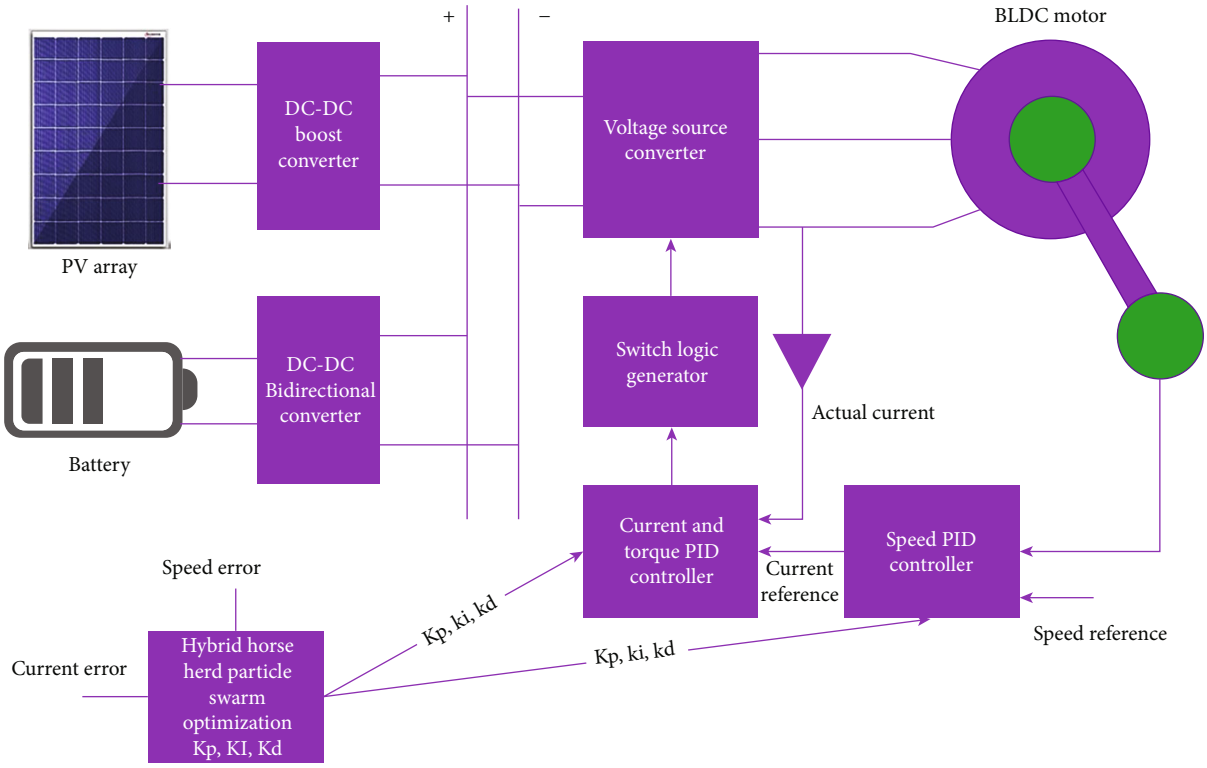


Figure 1: Analysis of control strategy of BLDC motor.

[4]. The control of speed of the BLDC motors is crucial for achieving ripple-free operation of speed and torque of the BLDC motor [5]. Because of the inductance effect of the winding, the rectangular stator current and trapezoidal back EMF of a BLDC motor take on their characteristic shapes. In order to calculate the speed variations caused by the stator current deviation, the ideal rectangular form is used [6–8]. The motor's vibration and sound are a result of its increased speed. The motor's speed regulation is vital factor for enhancing the efficiency of the BLDC motors [9]. Compared with proportional integral (PI) controllers, PID controllers are commonly used to control the BLDC motor's speed. In contrast to PI controllers, PID controllers may be fine-tuned to provide ideal speed control, noise suppression, and vibration damping via a combination of these features. Furthermore, BLDC motor speed is regulated by PID controllers [10]. Furthermore, a BLDC motor's speed controller's performance relies heavily on the fine-tuning of PID gains. Parameter tuning for a PID controller requires determining the best possible proportional, integral, and derivative gains for the PID controller to achieve the speed tracking performance [11].

In most cases, adjusting a PID controller is an extremely difficult process that is completed using a combination of rule-based and empirical approaches. PID control parameters are tuned using the rule-based Ziegler Nichols tuning technique [12]. The rule-based approach takes longer, is more time-consuming task, and causes hardware damage during control operations. Because of these properties, certain higher-order plants should not be supported by rule-based techniques, and again, it is increasing the tuning time as a result [13]. By using an integral square error (ISE) as the goal function for PID tuning, numerous researchers have created several optimization-based approaches for many applications at present. Tuning the PID controller to regulate the speed of the BLDC motor is the subject of a number of optimization techniques [14, 15]. Genetic algorithm (GA), fuzzy logic controller (FLC), bacterial foraging optimization (BFO), particle swarm optimization (PSO), artificial bee colony (ABC) optimization, firefly algorithm (FA), etc., are all examples of metaheuristic algorithms used for tuning of PID controller in application of speed control of BLDC motor. The next subsection provides the literature review on controllers and optimization methods used for control of BLDC motor and related applications.

1.1. Literature Review on Related Works. Comparison of model reference adaptive control (MRAC) with PID compensator and self-tuning fuzzy PID control for high-performance brushless DC motors is shown in [16]. The aim is to have the rotor speed tracked accurately regardless of load disturbance or changes in the parameters. MRAC with PID compensator outperforms self-tuning fuzzy PID control in simulations. High overshoot and a persistent steady-state inaccuracy, however, are the outcome of MRAC implementation alone.

In [17], we see how the flower pollination algorithm can be used to manage the speed of a brushless DC motor (BLDC) with optimal PID tuning. The flower pollination method mimics the process of pollination in flowers and is a metaheuristic optimization algorithm inspired by nature. Particle swarm optimization (PSO) and firefly algorithms,

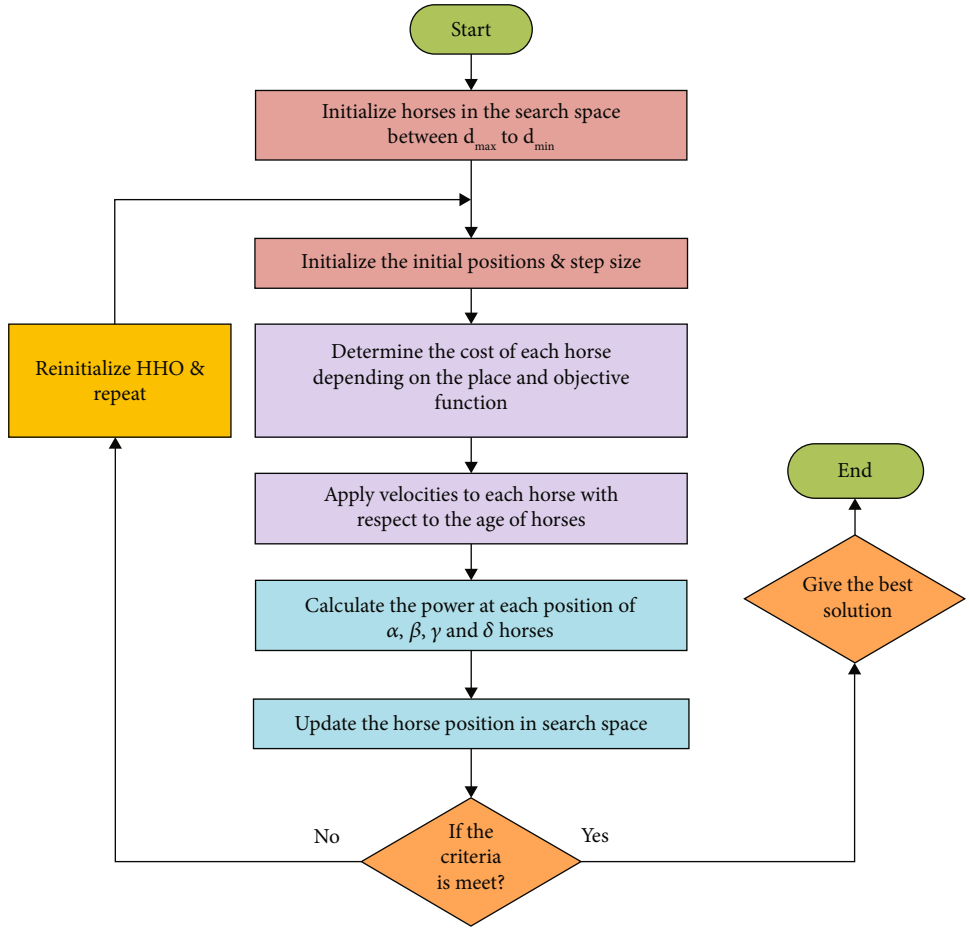


Figure 2: Flowchart of horse herd optimization.

both of which are inspired by natural processes, are used as benchmarks to evaluate the algorithm's performance. The system tests revealed that the BLDC motor's speed response has a chattering problem and a significant overshoot.

In [18], a PI+DF controller is constructed by fusing a PI control law with windup protection with a derivative path that employs a first-order low-pass filter. Next, the stochastic fractal search (SFS) technique, which is analogous to particle swarm optimization, is used to fine-tune the controller's gains, which include proportional, integral, derivative, and filter gains. There is significant overshoot in the speed response of the BLDC motor when using the SFO approach.

In [19], the fast terminal sliding mode control (FTSMC) strategy for controlling the speed of switching reluctance motors (SRMs) is described. By incorporating a fuzzy logic compensator into the FTSMC, this controller is able to dampen chattering effects and increase the sign function's value. The research shows that the proposed method is superior in the time and frequency domains compared to the proportional integral method and the traditional sliding mode control method. However, designing this controller is a time-consuming procedure due to its complexity.

In [20], an adaptive sliding mode observer (SMO) is studied for sensorless control of a brushless DC motor. The boundary layer is adaptively adjusted for changes in speed using fuzzy algorithms, and the observer is built with a sinusoidal saturation switching function. As a result, the observer is better able to make accurate predictions of low- and high-speed EMF values, rotor positions, and rotor speeds. Due to its straightforward design, high computing efficiency, and resilience, the SMO method has largely supplanted the extended Kalman filter (EKF) and model reference adaptive systems (MRAS). However, EKF and MRAS are both inadequate for widespread use in industry because of their restrictions.

In [21], we see how the sine cosine algorithm (SCA) can be used to tune a PID controller for DC motor speed regulation. The SCA is a recently developed optimization method that use sine and cosine functions to generate a set of initial candidate solutions then iteratively refine them until the optimal one is reached. The authors evaluate their SCA/PID strategy against conventional methods. The results of the tests showed that their method produced a lot of overshoots, a long rise time, and a long settling time.

In [22], the authors offer a method for controlling the speed of a DC motor by combining the Henry gas solubility optimization (HGSO) algorithm and adversarial learning (OBL). An OBL/HGSO-tuned PID controller is used to reduce the integral of time multiplied absolute error (ITAE) in this method. Robustness and disturbance rejection

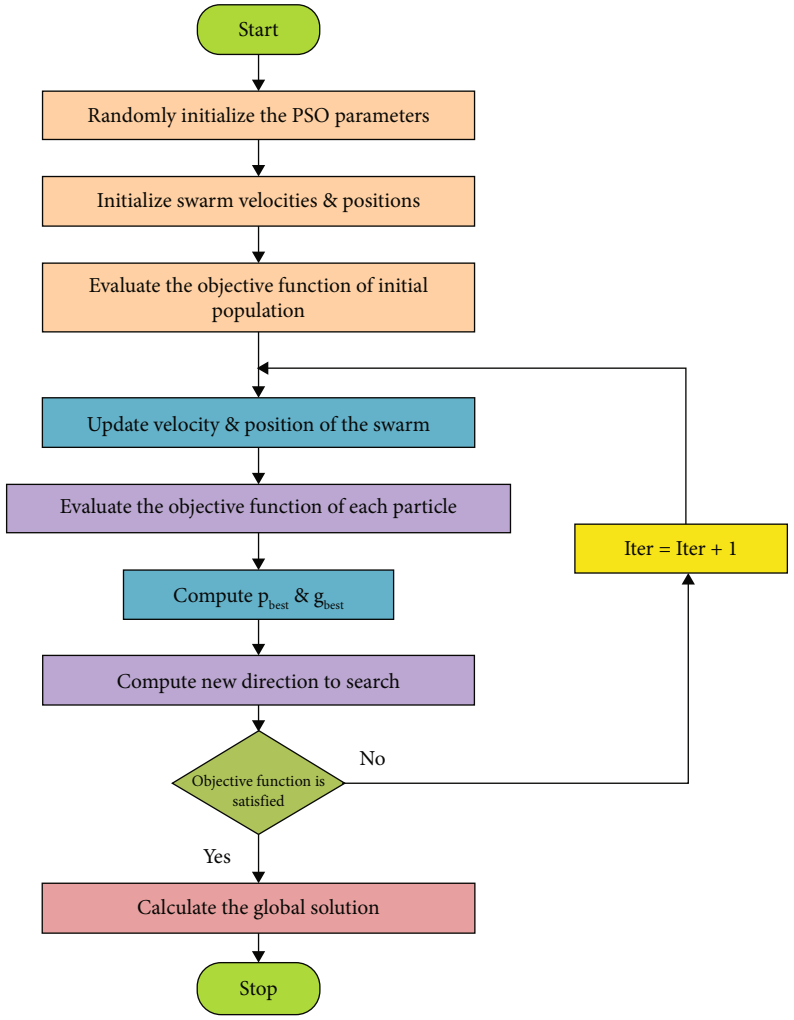


Figure 3: Flowchart of particle swarm optimization.

abilities of the proposed OBL/HGSO-PID controller are then evaluated, and the controller is compared to others modified with various optimization strategies. These optimization algorithms are not without their drawbacks, though, as they require more processing power, converge too quickly, get stuck at a local minimum, have poor parameter tuning flexibility, and take too much time to run.

A stochastic fractal search (SFS) technique is proposed in [23] for approximation and control of linear time-invariant (LTI) systems. The diffusion property seen in random fractals is exploited by SFS, a nature-inspired algorithm, to uncover the search space, making it accessible for usage in control systems. Low-order systems (LOS) are obtained from high-order systems (HOS) using SFS in approximation of LTI systems, with the original HOS's transient and steady-state features preserved. Simulation results show that SFS combined with a PID controller provides better control of a DC motor than a traditional PID controller. These optimization algorithms are not without their drawbacks, though, as they require more processing power, converge too quickly, get stuck at a local minimum, have poor parameter tuning flexibility, and take too much time to run.

The optimal tuning of fractional order proportional +integral+derivative (FOPID) controllers for DC motor speed control is provided in [24], along with the atom search optimization (ASO) algorithm and a novel chaotic variant termed the chaotic atom search optimization (ChASO) algorithm. The convergence rate and resulting precision of the ChASO algorithm are improved over those of ASO thanks to its foundation in logistic map chaotic sequences. The ASO and ChASO algorithms are useful because they are straightforward, can be quickly implemented, and can be used to a wide range of optimization problems. Consequences of using heuristic optimization techniques include sluggish convergence or early convergence to inferior solutions, both of which can result from using inappropriate parameter values.

In [25], the grey wolf optimizer (GWO) algorithm is used to fine-tune the settings of a PID controller used to regulate the rotational velocity of DC motors. The GWO algorithm is based on the cooperative hunting strategies of grey wolves and is aimed at striking a happy medium between discovery and exploitation. The study gives a comparison and robustness analysis of the GWO/FOPID method,

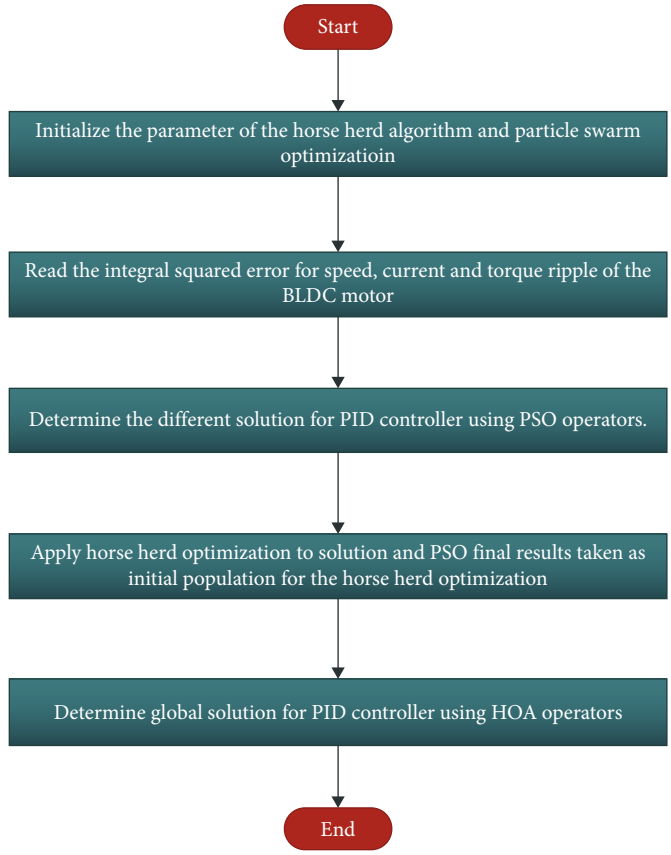


Figure 4: Hybrid horse herd particle swarm optimization.

contrasting its settling and rising times and finding that it has comparable overshoot to other methods already in use. These optimization algorithms are not without their drawbacks, though, as they require more processing power, converge too quickly, get stuck at a local minimum, have poor parameter tuning flexibility, and take too much time to run.

Brushless DC (BLDC) motor speed management using a fractional order fuzzy PI (FOFPI) controller is studied in [26]. The FOFPI controller uses the whale optimization algorithm (WOA) for optimization, and it solves these problems. The robustness of the suggested controller was confirmed through an examination of the control system's performance at varying speeds. However, in nonlinear and unpredictable systems, this controller performs poorly.

To optimize the parameters of a FOPID controller for speed control of a DC motor, [27] uses a hybrid of manta ray foraging optimization (MRFO) and simulated annealing (SA) algorithm (OBL-MRFO-SA). Time domain and frequency domain simulations, robustness, and load disturbance rejection assessments are only some of the ways that the algorithm is tested.

In [28–30], an improved slime mould algorithm (ISMA) is described for managing the speed of a direct current (DC) motor and keeping the terminal voltage level of an automated voltage regulator (AVR) constant. By combining a modified opposition-based learning scheme with the Nelder-Mead simplex search method, the suggested approach enhances the exploration and exploitation capabilities of the original slime mould algorithm.

The simulated annealing (SA) algorithm is embedded within an enhanced version of the atom search optimization (ASO) algorithm to improve its search capability in [31]. This enhanced version is called hASO-SA. Multilayer perceptron (MLP) training and proportional integral derivative (PID) controller design for regulating DC motor speed are just two examples of the linear and nonlinear issues that benefit from the hybrid algorithm's optimization. Unimodal, multimodal, hybrid, and composition benchmark functions are all put through their paces using this approach.

The PID controller used to regulate DC motor speed is tuned with a hybrid algorithm (LFDNM) described in [32]. The technique combines the robust local search capabilities of the Nelder-Mead (NM) algorithm with the exploratory nature of the Levy flight distribution (LFD) algorithm. Simulations were used to check the effectiveness of the suggested method in comparison to the cuckoo search algorithm, genetic algorithm, and original LFD algorithm.

When examining the above literature reviews, some of the optimization and control methods have high overshoot, high settling time, high rise time, and high steady-state error. Some methods are very complex and not applicable to industrial application. Some optimization takes longer convergence time and trap into local minima and requires lot more trials for getting optimal values for the controllers. These are to be considered as research gap in the literature review. This is considered as problem statement for this paper, and it is going to address using hybrid horse herd

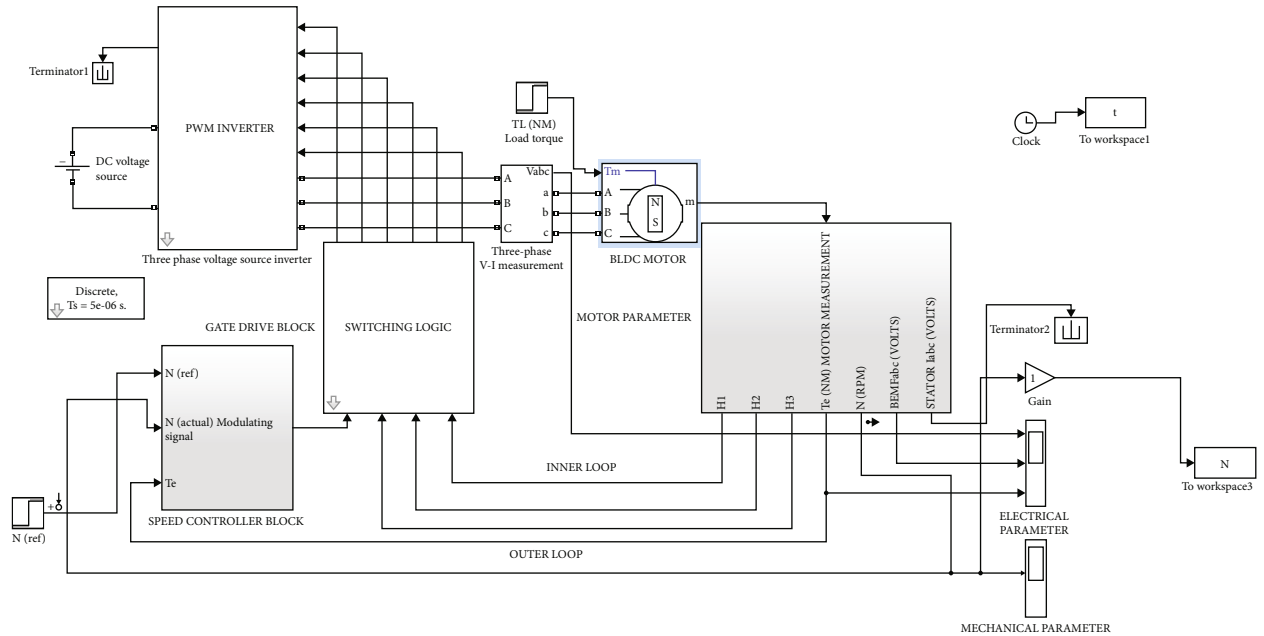


Figure 5: Overall Simulink model of the hybrid horse herd particle swarm-optimized brushless DC motor.

particle swarm optimization. The PSO provides quick convergence results and works well in exploration phase, and HHA works well in exploitation phase from the results of the PSO.

The main contribution of this article is presented as follows:

(i) To optimize the PID controller's gain parameter to regulate the speed of a BLDC motor using a hybrid horse herd particle swarm technique

(ii) MATLAB simulation developed for implementing proposed hybrid horse herd particle swarm optimization-tuned PID-controlled PV-powered BLDC motor

(iii) The proposed controller is compared with SCA [21], GWA-FOPID [22], CASOA [23], SFSA [24], HGSOA [25], WOA [26], and FA [33]

(iv) Experimental verification of the proposed method realized for checking the suitability of the proposed work

The rest of the article is organized as follows: Section 2 detailed the BLDC motor's modelling, and Section 3 outlined the proposed methods for regulating the motor's speed using a PID controller, and controller optimization technique and procedure are described. In Section 4, simulation and experimental findings based on our proposed technique are described. Finally, in Section 5, we discuss the inferences of our work and work for future.

2. Design of the Speed Control of BLDC Motor

Because of its high-power density, extended working life, low noise, high efficiency, and excellent speed vs. torque characteristics [21], the brushless DC motor (BLDC) is gaining popularity and is employed in a variety of applications. Despite the decades of steady progress in adjustable speed driver control, microprocessors, and semiconductors, these applications need effective control of the speed in all kinds of operating conditions. Therefore, the BLDC motor's speed regulation is crucial for optimizing the motor's performance. Figure 1 depicts the suggested speed control for the BLDC motor. BLDC motor was powered by PV and battery system.

Table 1: Specification parameters of BLDC motor.

Description	Values
Stator phase resistance (Rs)	0.2 (ohm)
Stator phase inductance (Ls)	8.5*e*-3
Flux linkage	0.175
Back EMF flat area	120 (degrees)
Inertia	0.089 J (kg·m^2)

An encoder is used to determine the BLDC motor's rotational velocity or speed. The speed is then compared to the standard reference speed as per application, and the speed error is calculated. The PID controller receives the speed error. The developed hybrid horse herd particle swarm optimization approach is used to fine-tune the gains of the PID controller. The tuning of the PID controller is subjected to minimize the fitness function or objective function of the BLDC motor, and it is framed as a multiobjective function. In this work, multiobjective function has additions of the integral of the squared speed error value (ISSE), integral of squared current error value (ISCE), and integral of squared torque error value (ISTE). The Mut objective function is represented by the following equations:

$$\vartheta = \min\left(\Delta_1 + \Delta_2 + \Delta_3\right), \tag{1}$$

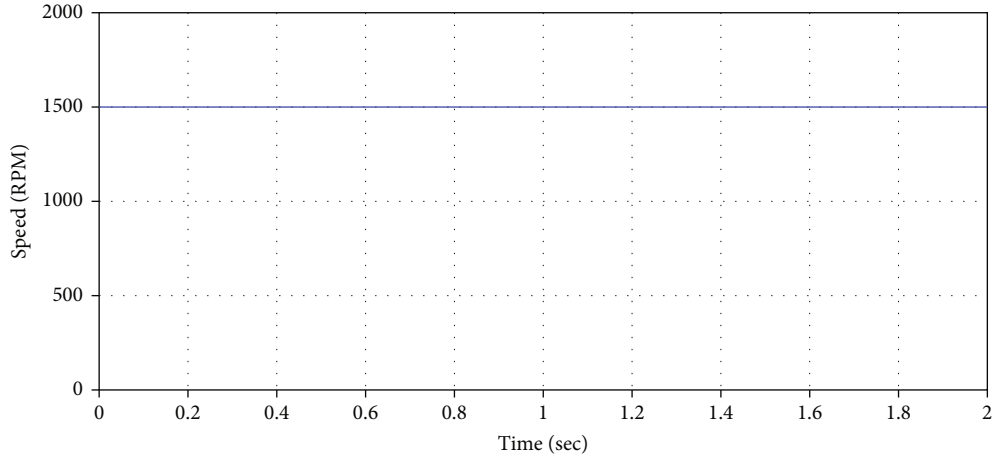


Figure 6: Constant reference speed condition.

where $\Delta_1 = \vartheta_1$, $\Delta_2 = \vartheta_2$, and $\Delta_3 = \vartheta_3$.

$$\Delta_1 = \int_0^T (N_{\text{ref}} - N_{\text{bldc}})^2, \tag{2}$$

$$\Delta_2 = \int_0^T (I_{\text{ref}} - I_{\text{bldc}})^2, \tag{3}$$

$$\Delta_3 = \frac{1}{T_{\text{bldc}}} \int_0^T (T_{\max} - T_{\min})^2. \tag{4}$$

From these equations, Δ_1, Δ_2, and Δ_3 are represented as ISSE, ISCE, and ISTE, correspondingly. The multiobjective function of ϑ_1, ϑ_2, and ϑ_3 is represented as ISE current error, ISE speed error, and ISE torque ripple, correspondingly. The error values are determined and pass to the proposed controller and optimization algorithms. In PID regulator, the speed and current control are accomplished by the optimal choice of gain parameters. In operating condition, the actual speed (N_{bldc}), current (I_{bldc}), and torque (T_{bldc}) are evaluated from the BLDC motor. Afterward, the actual speed, current, and torque values are contrasted by means of the reference values (N_{ref}, I_{ref}, and $T_{\max} - T_{\min}$). The reduction of error values ϑ is minimized by the tuning of PID regulator using HHHPSO method. The controller gain parameters are calculated in the exploitation of hybrid horse herd particle swarm optimization algorithm. The brief depiction of PID controller model and design is presented in Section 2.2. The next section explains the modelling of the BLDC motor.

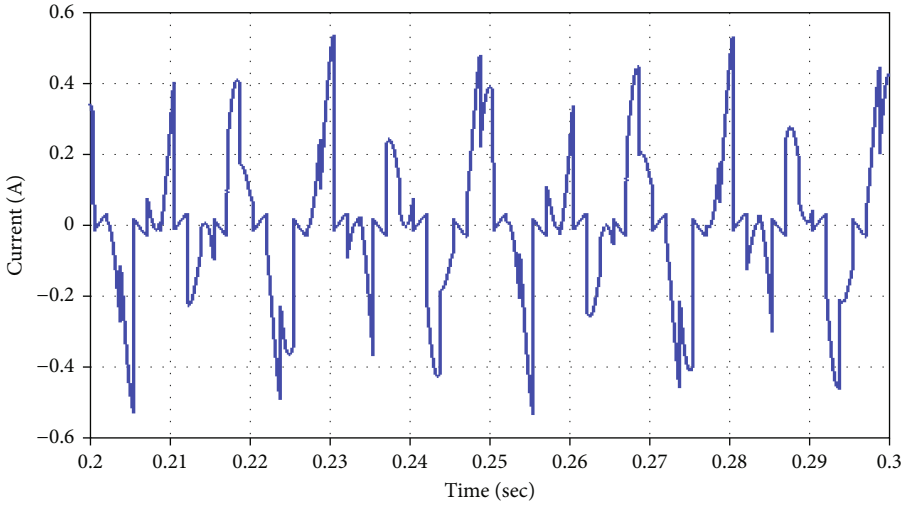
2.1. Mathematical Modelling of the BLDC Motor. Modelling a BLDC motor is analogous to modelling a synchronous machine. At first, a BLDC motor is connected to a three-phase power supply. The BLDC motor is unique in that it can operate with a nonsinusoidal power supply. The peak voltage should not be higher than the motor's maximum safe operating voltage. The following equations are used for denote the voltage of the BLDC motor armature windings:

$$\begin{aligned} u_a &= i_a \times r_a + \frac{di_a}{dt} \times L_a + e_a, \\ u_b &= i_b \times r_b + \frac{di_b}{dt} \times L_b + e_b, \\ u_c &= i_c \times r_c + \frac{di_c}{dt} \times L_c + e_c, \end{aligned} \tag{5}$$

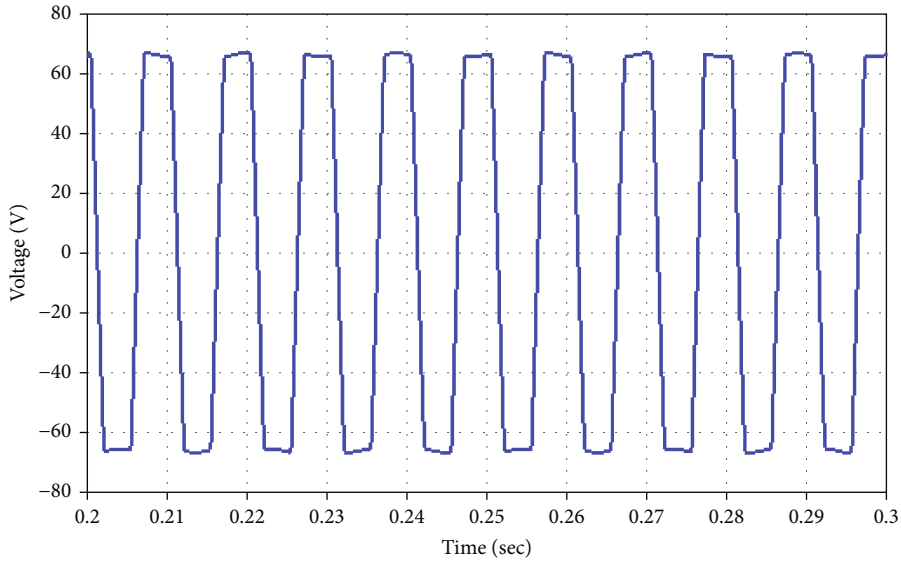
where e_a, e_b, and e_c can be described as the trapezoidal back EMF of different phases such as phases a, b, and c; i_a, i_b, and i_c can be described as the motor input current of phases a, b, and c; u_a, u_b, and u_c can be described as the terminal voltage of phases a, b, and c; r_a, r_b, and r_c can be referred as the armature resistance of stator phase winding; and L_a, L_b, and L_c can be represented as the armature self-inductance of the BLDC motor [22]. Based on the motor specifications, the voltage equation matrix is derived which is presented in the below equation:

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = \begin{bmatrix} r_a & 0 & 0 \\ 0 & r_b & 0 \\ 0 & 0 & r_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} L_a & L_{ab} & L_{ac} \\ L_{ba} & L_b & L_{bc} \\ L_{ca} & L_{cb} & L_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}. \tag{6}$$

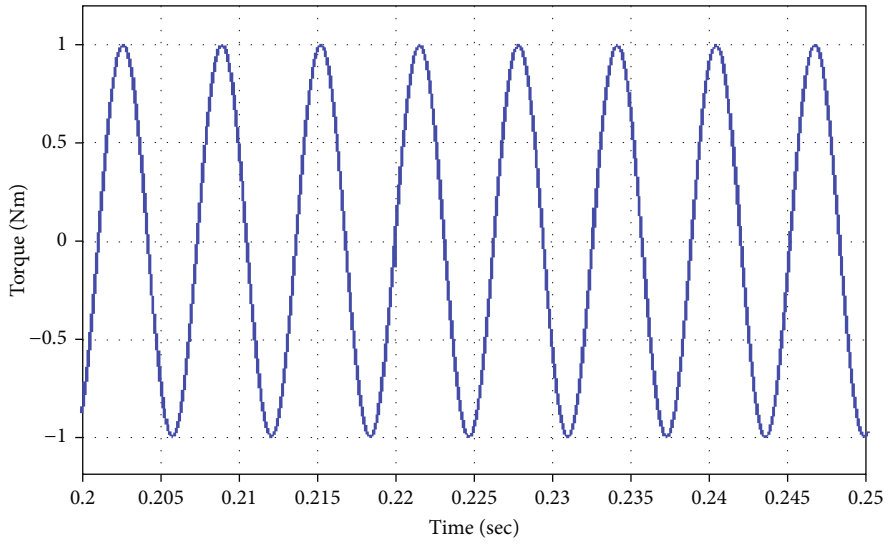
Iron loss and saturation loss are almost nonexistent in a BLDC motor. Moreover, there is no difference between the self-inductance and stator winding resistance of all three



(a)

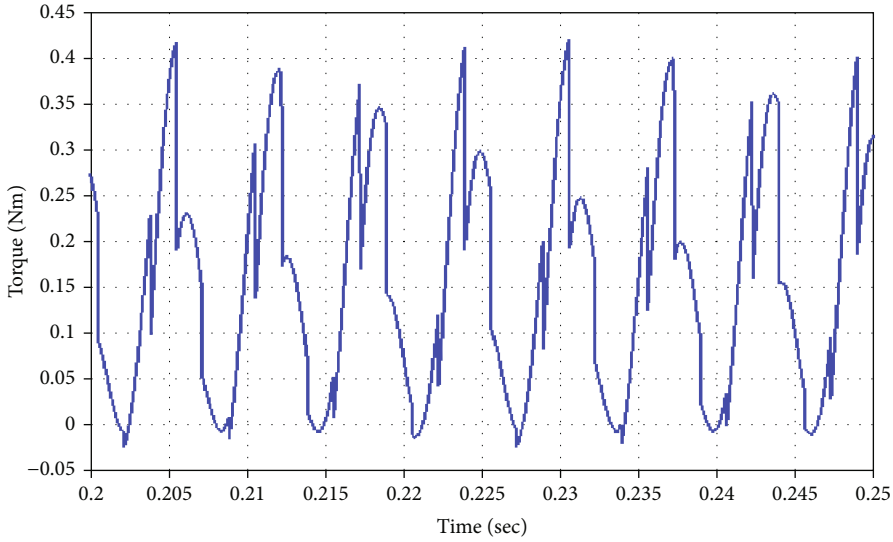


(b)



(c)

Figure 7: Continued.



(d)

Figure 7: Analysis of BLDC at 1500 rpm: (a) current, (b) voltage, (c) mechanical torque, and (d) motor torque.

phases. The following equations represent the BLDC motor with mutual inductance form:

$$\begin{aligned} r_a &= r_b = r_c = r, \\ L_a &= L_b = L_c = L, \\ L_{ab} &= L_{ac} = L_{ba} = L_{bc} = L_{ca} = L_{cb} = L_m, \\ \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} &= \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} L & L_m & L_m \\ L_m & L & L_m \\ L_m & L_m & L \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}. \end{aligned} \tag{7}$$

The back EMF phase equations are computed based on rotor position of the motor and speed of the motor and represented in the following equations:

$$\begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} = \begin{bmatrix} K_v \times f_{as}(\theta) \times \omega \\ K_v \times f_{bs}\left(\theta + \frac{2\pi}{3}\right) \times \omega \\ K_v \times f_{cs}\left(\theta + \frac{4\pi}{3}\right) \times \omega \end{bmatrix}, \tag{8}$$

where ω can be represented as the speed of rotor (rad/sec); f_{as}, f_{bs}, and f_{cs} can be represented as the trapezoidal functions; θ can be represented as the electrical rotor angle; and K_v can be represented as back EMF constant of one phase (v/rad/sec). The trapezoidal functions are described in the following equations:

$$f_{as}(\theta) = \begin{cases} \frac{6\theta}{\pi} & 0 < \theta \le \frac{\pi}{6}, \\ 1 & \frac{\pi}{6} < \theta \le \frac{5\pi}{6}, \\ -1 - \frac{(0 - (5\pi/6))}{\pi/6} & \frac{5\pi}{6} < \theta \le \frac{7\pi}{6}, \\ -1 & \frac{7}{6} < \theta \le \frac{11\pi}{6}, \\ -1 + \frac{(\theta - (11\pi/6))}{\pi/6} & \frac{1\pi}{6} < \theta \le 2\pi. \end{cases} \tag{9}$$

Similarly, the functions of the f_{bs} and f_{cs} can be described in the following equations:

$$\begin{aligned} f_{bs} &= f_{as}\left(\theta + \frac{2\pi}{3}\right), \\ f_{cs} &= f_{as}\left(\theta + \frac{4\pi}{3}\right). \end{aligned} \tag{10}$$

There is a correlation between the mechanical rotor angle and the electrical rotor angle, as well as the number of pole pairs. The mathematical relation of the electrical and mechanical rotor angles is described in the following equation:

$$\theta = \frac{P}{2}\theta_m, \tag{11}$$

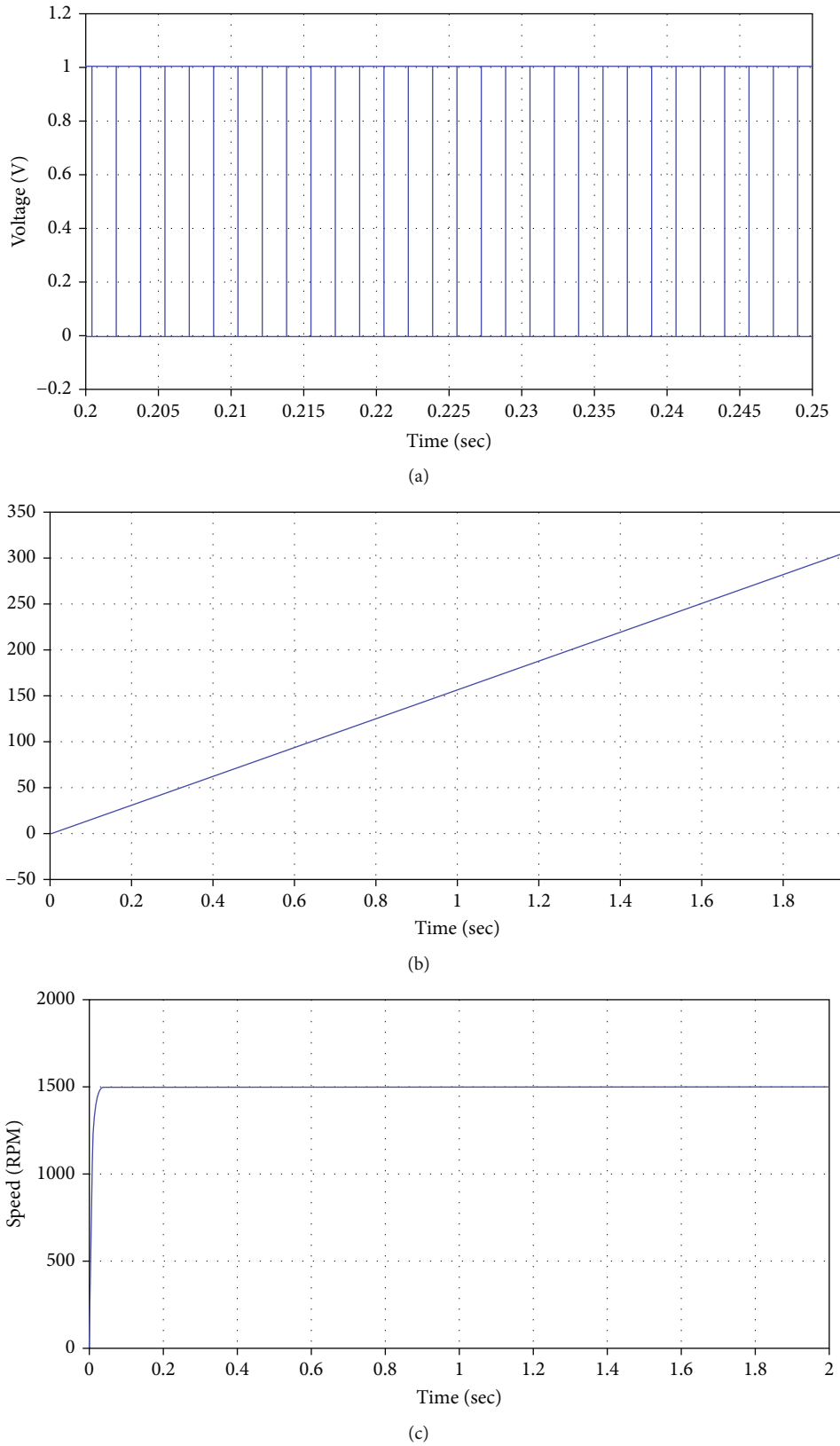


Figure 8: Analysis of BLDC at 1500 rpm: (a) pulses of inverter, (b) rotor angle, and (c) output speed.

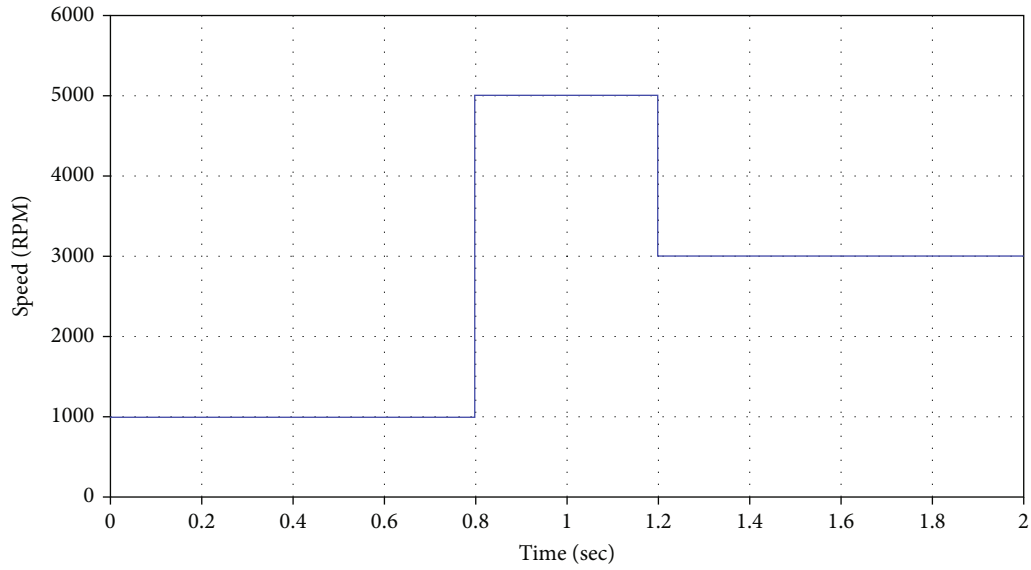


Figure 9: Speed reference variation.

where θ_m is represented as angle of rotor in mechanical degree in the motor. T_{bldc} is the electromagnetic torque which can be described as the below equation:

$$T_{\text{bldc}} = \frac{e_a \times i_a + e_b \times i_b + e_c \times i_c}{\omega}. \tag{12}$$

From the motor torque, equation with inertia, load torque, and friction coefficient can be represented in the below equation:

$$T_{\text{bldc}} - T_L = J\frac{d\omega}{dt} + B \times \omega, \tag{13}$$

where T_L is the load torque of the BLDC motor, J is the moment of inertia of the BLDC motor, and B is the friction coefficient of the BLDC motor.

2.2. Description of the PID Controller. The gain parameter of the PID controller used in the speed, current, and torque controller is optimized using HHHPSO to minimized objective function mentioned in Equations (1) to (4), and also for minimizing the torque ripple, the following constraints should meet during optimization, and it is described as the following equations:

$$\max(\vartheta_2) < \xi. \tag{14}$$

From Equation (14), the maximum current limit is represented by the constraints ξ. Here, the $\xi = 0.9$ value is calculated by the trial error method. The gain limitations for PID controllers, current limitation, speed limitation, and rotor angle limitation are described in the following equation to obtain optimal values of the PID controllers:

$$\begin{aligned}
K_{P,\min} &\le K_P \le K_{P,\max},\\
K_{I,\min} &\le K_I \le K_{I,\max},\\
K_{D,\min} &\le K_D \le K_{D,\max},\\
V_{\min} &\le V \le V_{\max},\\
I_{\min} &\le I \le I_{\max},\\
\theta_{\text{on},\min} &\le \theta \le \theta_{\text{on},\max},\\
\theta_{\text{off},\min} &\le \theta \le \theta_{\text{off},\max}.
\end{aligned} \tag{15}$$

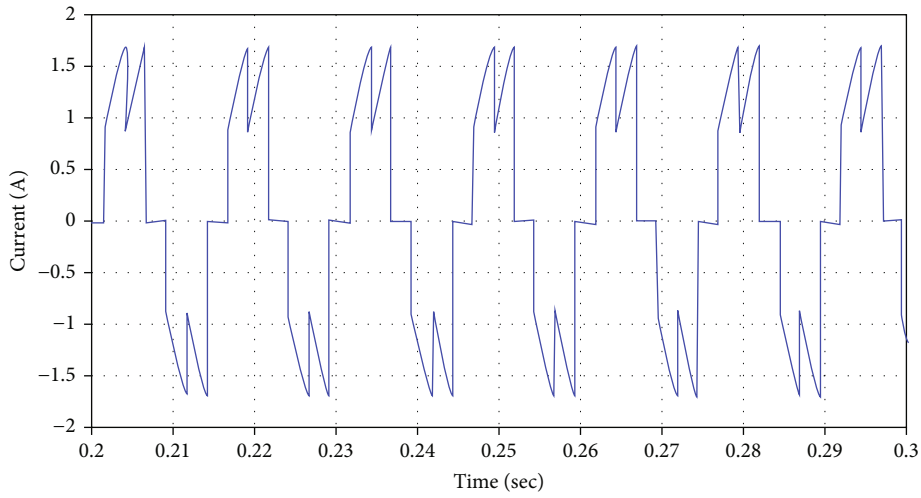
The gain parameters are calculated by means of the procedure of HHPSO. Normally, the tuning of gain parameters is accomplished by the speed, current, torque, and turn on and turn off angle control of BLDC. With the use of a HHPSO technique, the best gain parameters are determined. In the following section, the detailed steps taken by the HHPSO techniques are provided.

3. Hybrid Horse Herd Particle Swarm Optimization

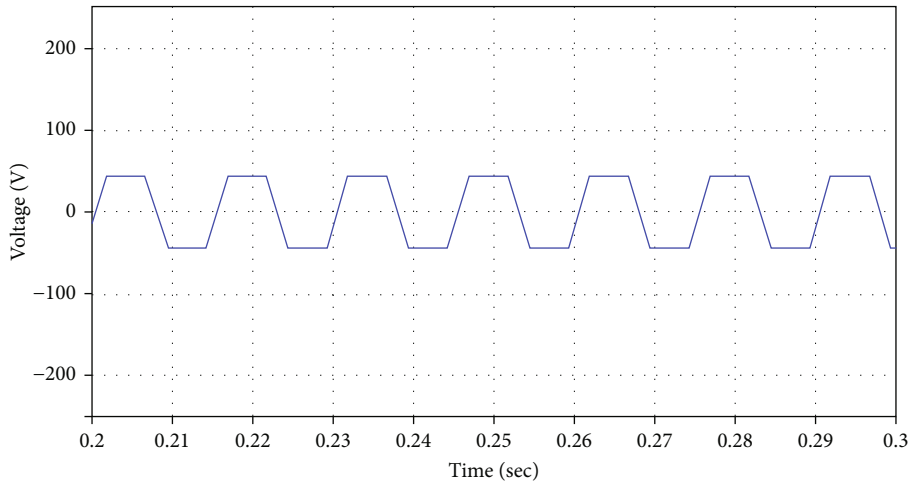
Particle swarm optimization, horse herd optimization, and hybrid horse herd particle swarm optimization are discussed in this section.

3.1. Horse Herd Optimization Algorithm. To develop the HHO algorithm, we studied equine behaviour in its natural settings. Horses often display the grazing (A), hierarchical (B), sociable (C), imitative (E), and defensive (D) behaviours (F) [34]. Because of this, our algorithm was inspired by the six behaviours of stallions over many ages. The horse movement equation is described below:

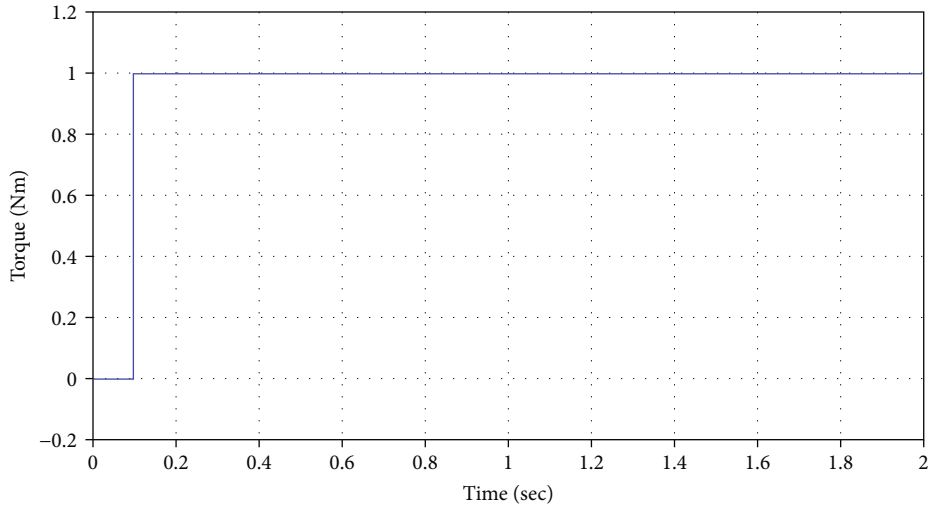
$$Z_i^{\text{iter,AGE}} = S_i^{\text{iter,AGE}} + Z_i^{(\text{iter}-1),\text{AGE}}, \text{AGE} = \alpha, \beta, \gamma, \delta. \tag{16}$$



(a)

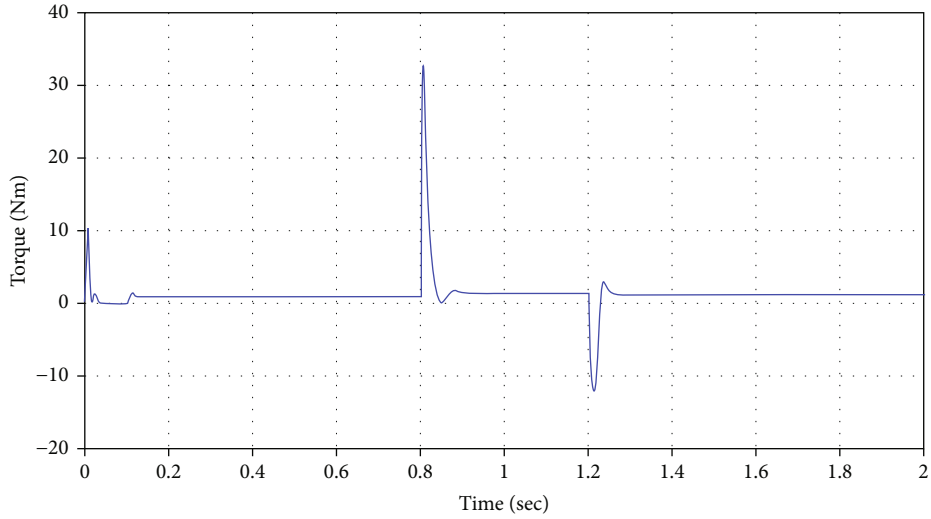


(b)



(c)

Figure 10: Continued.



(d)

Figure 10: Analysis of BLDC for varying speed conditions: (a) current, (b) voltage, (c) mechanical torque, and (d) motor torque.

The ith stallion's location is shown by $Z_x^{\text{iter,AGE}}$, the stallion's age range is shown by AGE, the current repetition is indicated by Iter, and the horse's velocity is shown by $S_i^{\text{iter,AGE}}$. Horses' personalities and habits change as they mature. It is estimated that the average life expectancy of a horse is between 25 and 30 years. Equine ages 0–5 years, 5–10 years, 10–15 years, and above 15 years are represented as α, β, γ, and δ. Iteratively sifting through a whole matrix of responses is the best strategy for determining the appropriate age for stallions. Using a best-response ranking, we may choose 10% of the horses in the whole matrix as "horses" for further examination. That group comprises the next 20% of survey takers. The remaining 30% are γ horses, while 40% are just plain horses.

For food, they graze on grass, forage, and other plants. They graze for 16–20 hours a day, with very little downtime in between. Continuous eating describes the condition in which a person eats without stopping. Maybe you have seen the mares and their young in the pasture. The HHO algorithm calculates an estimated grazing space for each horse by using a coefficient of grass. Horses of all ages are often seen frolicking on the grass. The following formula is a mathematical implementation of grazing:

$$A_i^{\text{iter,AGE}} = a_i{}^{\text{iter,AGE}} \times (U + R \times L) \times \left(Z_i^{(\text{iter}-1)}\right), \text{AGE} = \alpha, \beta, \gamma, \delta,$$
$$a_i{}^{\text{iter,AGE}} = a_i{}^{(\text{iter}-1),\text{AGE}} \times w_a, \tag{17}$$

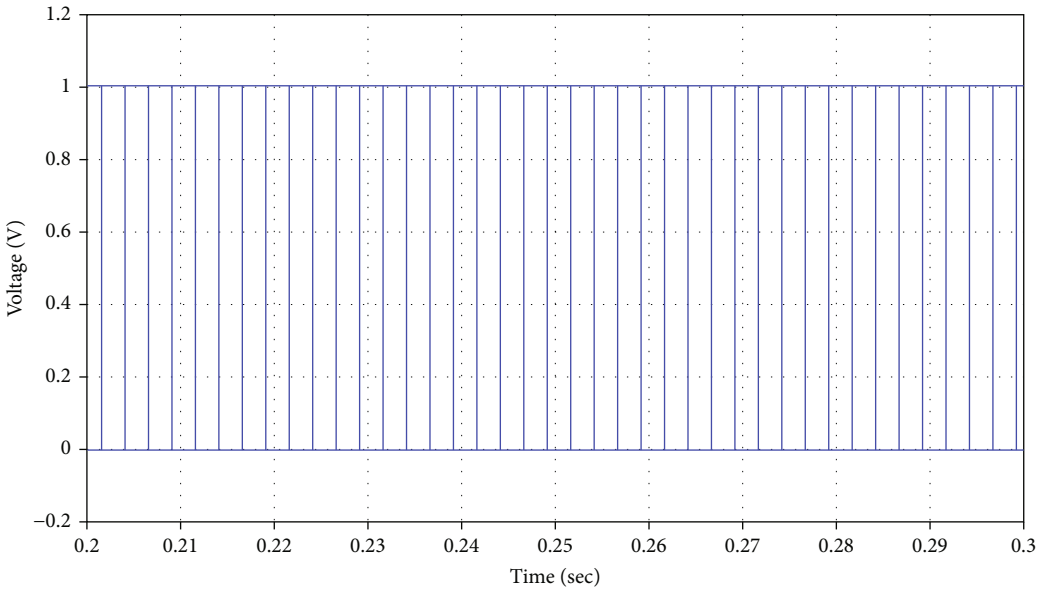
where w_a is the horse's motion parameter and $A_i^{\text{iter,AGE}}$ demonstrates the horse's inclination to graze. With this consideration, in the linearity of$a_i{}^{\text{iter,AGE}}$, each iteration is reduced. The results of this formula vary between zero and one, with R being a random value between zero and one. Taking all ages into account, set U and L to 1.05 and 0.95, respectively, and set the constant w_a to 1.5.

In the wild, horses have no place. They behave similarly to humans in that they obey someone in authority. Adult stallions and mares, following the law of hierarchy, are also responsible for leading herds of wild horses. Coefficient h of HHO considers herd behaviour in which animals tend to flock to the leader. Young horses (aged 5–15) have been the subject of substantial research, and the extent to which they conform to the law of hierarchy has been quantitatively defined as

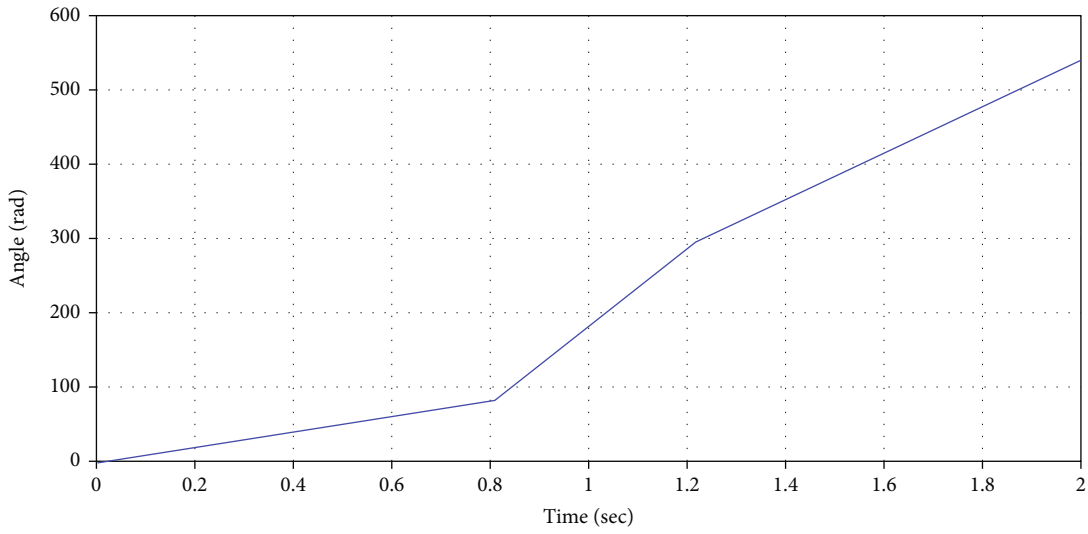
$$B_i^{\text{iter,AGE}} = b_{\text{iter}}{}^{\text{iter,AGE}} \times \left(Z_*^{(\text{iter}-1)} - Z_i^{(\text{iter}-1)}\right), \text{AGE} = \alpha, \beta, \gamma,$$
$$b_i^{\text{iter,AGE}} = b_{\text{i}}{}^{(\text{iter}-1),\text{AGE}} \times w_b. \tag{18}$$

$B_i^{\text{iter,AGE}}$ demonstrates that the best horse location has a large effect on the velocity parameter, and $Z_*^{(\text{iter}-1)}$ reveals the precise position of the best horse.

Horses thrive in social environments and may even be housed among other animals. Consequently, the stallions are being targeted by predators, while the herd environment previously made them feel secure. Being part of a diverse community increases your chances of staying alive and breaking free. Due to their extroverted nature, horses often engage in violent altercations with one another. It is because of the horse's individuality that they gets so excited. It seems that some of the horses would rather not be alone, since they get along rather well with the cattle and sheep. The horse

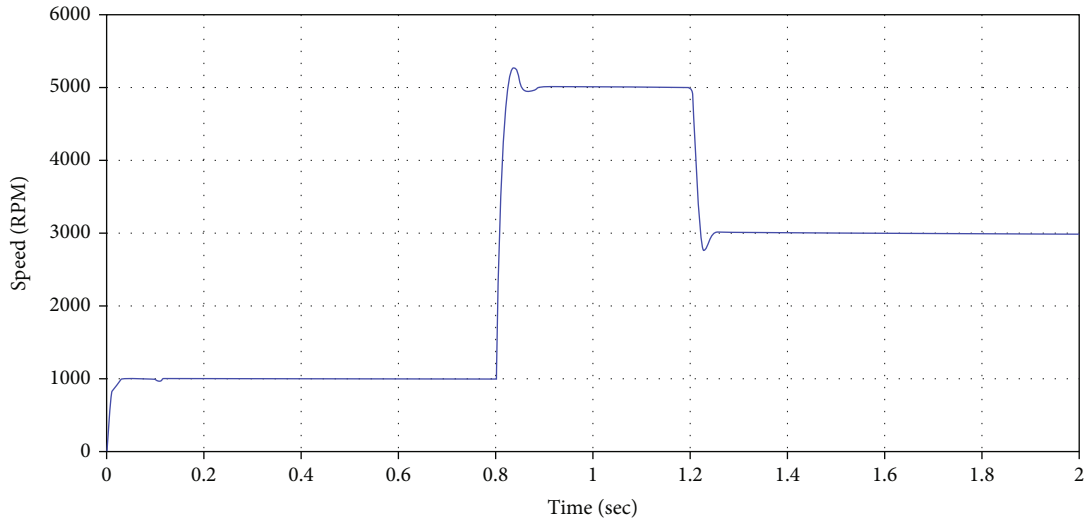


(a)



(b)

Figure 11: Continued.



(c)

Figure 11: Analysis of BLDC for varying speed conditions: (a) pulses of inverter, (b) rotor angle, and (c) output speed.

Table 2: PID gains for speed control of BLDC motor.

S. no.	Tuning method	Gain parameters of the PID		
		Proportional	Integral	Derivative
1	Proposed HHHPSO	0.35	1.25	0.05
2	SCA [21]	11.3163	0.5544	1.8072
3	GWA-FOPID [22]	18.328	4.9418	3.2612
4	CASOA [23]	11.9437	2.0521	2.4358
5	SFSA [24]	1.6315	0.2798	0.2395
6	HGSOA [25]	13.4430	1.2059	2.2707
7	WOA [26]	12.3	1.4	1.7
8	FA [33]	11.4	2.3	2.64

Table 3: Transient response criteria for PID control in BLDC motor.

S. no.	Tuning method	Transient response criteria		
		Maximum overshoot	Rise time	Settling time
1	Proposed HHHPSO	No overshoot	0.021	0.039
2	SCA [21]	0.192	0.0833	0.138
3	GWA-FOPID [22]	0.51	0.058	0.1172
4	CASOA [23]	No overshoot	0.0253	0.0405
5	SFSA [24]	0	0.638	1.06
6	HGSOA [25]	0	0.0798	0.0457
7	WOA [26]	0	0.035	0.0498
8	FA [33]	0	0.033	0.0476

seems to be moving toward the middle of the pack. Equating a horse's social behaviour to the following formula, we can see that young horses (aged 5-15) are very social:

$$C_i^{\text{iter,AGE}} = c_{\text{iter}}{}^{\text{iter,AGE}} \times \left(\left(\frac{1}{N} \sum_{k=1}^{N} Z_k^{(\text{iter}-1)} \right) - Z_i^{(\text{iter}-1)} \right), \text{AGE} = \beta, \gamma,$$

$$c_i{}^{\text{iter,AGE}} = c_i{}^{(\text{iter}-1),\text{AGE}} \times w_c. \tag{19}$$

The social velocity vector, $C_i^{\text{iter,AGE}}$, of the ith stallion, and the direction, $C_i^{\text{iter,AGE}}$, towards the herd, are both defined in terms of age. Each iteration, $C_i^{\text{iter,AGE}}$, decreases by a factor of w_c. Both the total number of horses and their age distribution are shown in the N field. To do a parameter sensitivity analysis, we determine the c coefficients for both and horses. It is commonly known that horses learn from one another, both good and harmful habits like where to locate grass. The current algorithm also takes equine

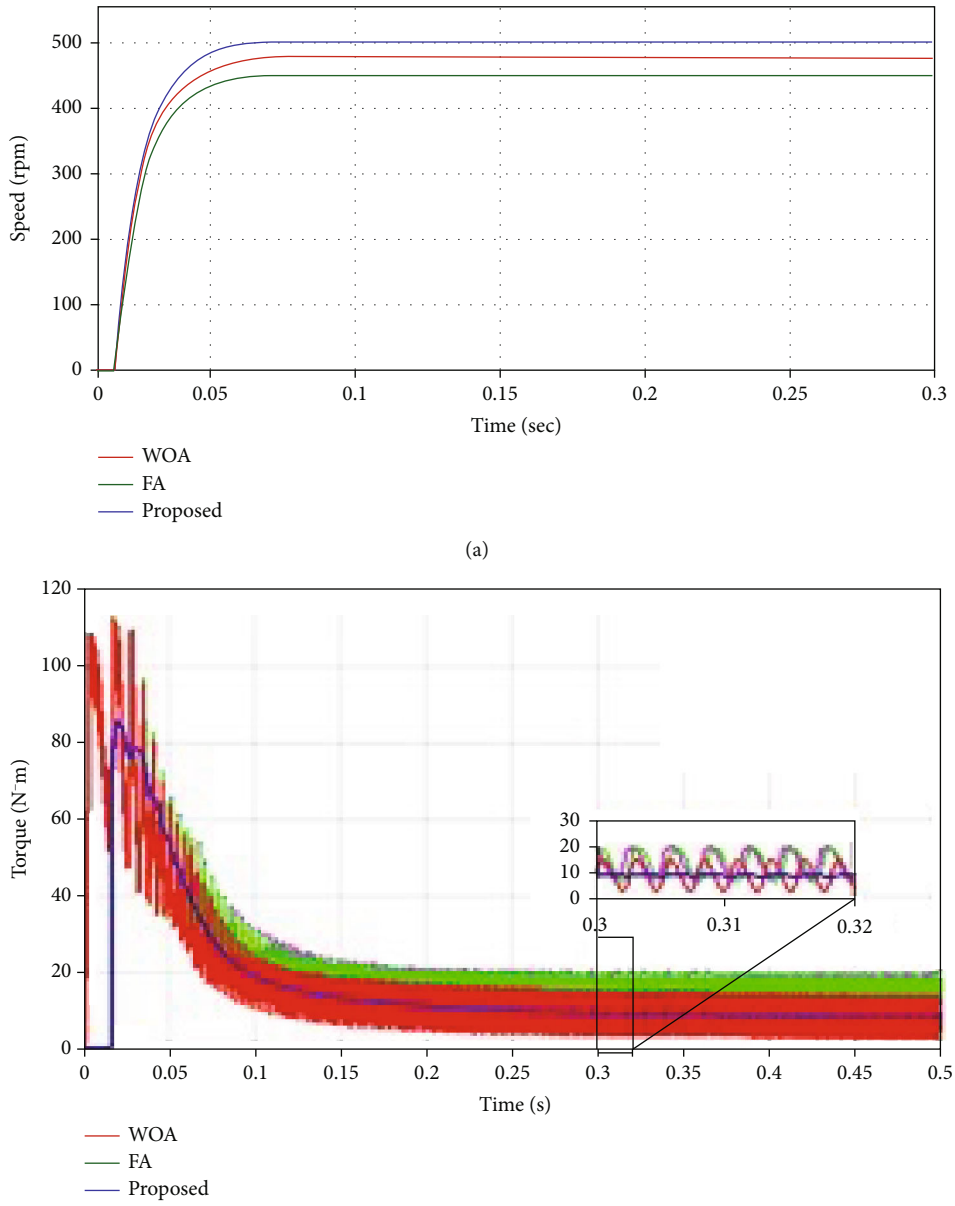


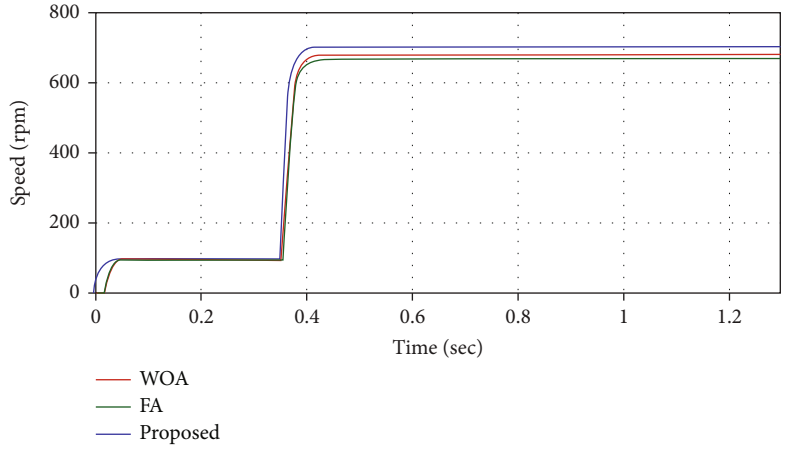
Figure 12: Comparison analysis of case 1: (a) speed and (b) torque.

mimicry into account. The following equation expresses the lifelong tendency of a young horse to mimic its elders:

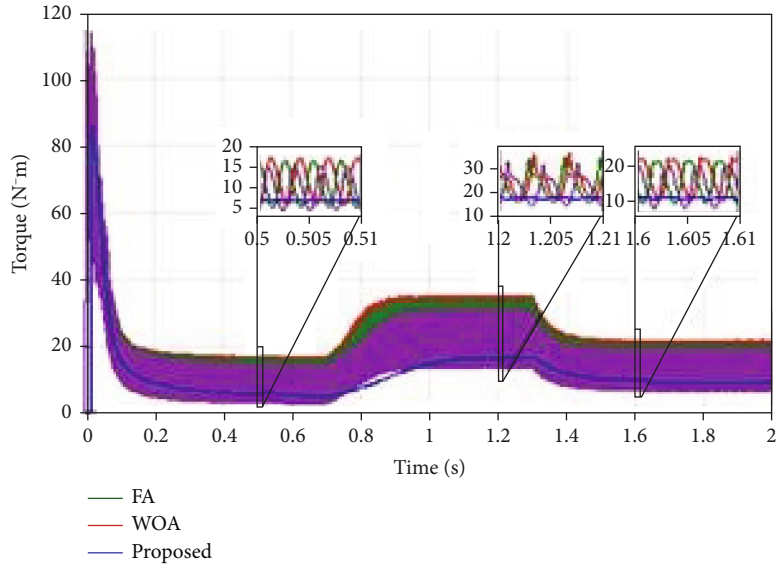
$$E_i^{\text{iter,AGE}} = e_{\text{iter}}{}^{\text{iter,AGE}} \times \left(\left(\frac{1}{pN} \sum_{k=1}^{pN} Z_k^{(\text{iter}-1)} \right) - Z_i^{(\text{iter}-1)} \right), \text{AGE} = \gamma,$$
$$e_i{}^{\text{iter,AGE}} = e_i{}^{(\text{iter}-1),\text{AGE}} \times w_e. \tag{20}$$

In order to get to where the best horses in the region average, $E_i^{\text{iter,AGE}}$ indicates $Z_k^{(\text{iter}-1)}$ positions. pN is the total number of top-finishing horses. Roughly 10% of the horses have been put up as p. If we apply the above example, we see that the reduction factor is denoted by the symbol w_e.

Horses' abusive treatment at the hands of predators may be seen in the animals' subsequent behaviour. For self-preservation, they exhibit the fight-or-flight response. The need to run away is ingrained in them. Furthermore, if they are captured, they will buck. Horses instinctively defend their territory and territory resources from competitors and predators like wolves. The protective mechanism of a horse in the HHO algorithm is to avoid collision with other horses that provide incorrect replies. Essential to their defence is factor d. In the face of danger, horses have two options: flee away or stand their ground. A horse, whether young or old, will always choose this safety mechanism if given the chance. A



(a)



(b)

Figure 13: Comparison analysis of case 2: (a) speed and (b) torque.

negative coefficient represents the horse's protective mechanism, which protects it from being positioned incorrectly.

$$D_i^{\text{iter,AGE}} = -d_{\text{iter}}^{\text{iter,AGE}} \times \left(\left(\frac{1}{qN} \sum_{k=1}^{qN} Z_k^{(\text{iter}-1)} \right) - Z_i^{(\text{iter}-1)} \right), \text{AGE} = \alpha, \beta, \gamma,$$
$$d_i^{\text{iter,AGE}} = d_i^{(\text{iter}-1),\text{AGE}} \times w_d, \tag{21}$$

where $D_i^{\text{iter,AGE}}$ is the outflow vector of the ith stallion from the worst-case placements of the $Z_k^{(\text{iter}-1)}$ vector. In addition, the worst-case total number of horses is provided in qN. One study found that q accounts for 20% of the world's horses. The reducing factor is denoted by w_d.

Horses may go from pasture to pasture in the countryside for hours at a time in search of food. While there are few wild horses, the vast majority of horses nowadays are kept in stables. The urge to graze might cause a horse to suddenly move to a new location. Horses are naturally curious animals who are always on the lookout for new territory. This fulfils the horses' innate curiosity in their environment, as they can observe one other through the walls. Calculated by multiplying a random motion model by a significant value. Wild behaviour in horses often begins at a young age and gradually fades as the horse becomes older.

$$F_i^{\text{iter,AGE}} = f_{\text{iter}}^{\text{iter,AGE}} \times \mathscr{P}\left(Z_i^{(\text{iter}-1)} \right), \text{AGE} = \gamma, \delta,$$
$$f_i^{\text{iter,AGE}} = f_i^{(\text{iter}-1),\text{AGE}} \times w_f. \tag{22}$$

A horse's velocity is chosen at random for use in local search and evasion of minimums, where $F_i^{\text{iter,AGE}}$ represents the random velocity vector of ith horse for local search and w_f displays the reduction factor of $f_i^{\text{iter,AGE}}$ every cycle.

The age-specific velocity vector of horses during each iteration of the algorithm may be expressed by the following equation, and flowchart of HHO is shown in Figure 2.

$$S_i^{\text{iter},\alpha} = A_i^{\text{iter},\alpha} + D_i^{\text{iter},\alpha},$$
$$S_i^{\text{iter},\beta} = A_i^{\text{iter},\beta} + B_i^{\text{iter},\beta} + C_i^{\text{iter},\beta} + D_i^{\text{iter},\beta},$$
$$S_i^{\text{iter},\gamma} = A_i^{\text{iter},\gamma} + B_i^{\text{iter},\gamma} + C_i^{\text{iter},\gamma} + E_i^{\text{iter},\gamma} + D_i^{\text{iter},\gamma} + F_i^{\text{iter},\gamma},$$
$$S_i^{\text{iter},\delta} = A_i^{\text{iter},\delta} + E_i^{\text{iter},\delta} + F_i^{\text{iter},\delta}. \tag{23}$$

3.2. Particle Swarm Optimization. PSO, or particle swarm optimization, was developed in 1995 by Kennedy and Eberhart. It takes its cues from the cooperative behaviours of flocking birds and schooling fish. The simulation of social behaviour provided the impetus for the development of particle swarms. PSO employs just the simplest mathematical operators and is very light on both memory and processing speed in comparison to other algorithms. In PSO, a swarm is made up of numerous individual particles. Each particle position for a potential answer to the problem statement of optimization. One may argue that every single position represents a possible answer. Each particle moves to a new position using its new velocity and the movement vectors in accordance with the previous best solution and the global best solution. Thereafter, the optimal solution is saved, and each particle speeds up in the direction of both the optimal global and optimal local solutions. More particles will approach a location where a particle has discovered a potential answer. The flowchart for PSO is shown in Figure 3.

In the y-dimension, x particles travel at a predetermined speed. All particles modify their positions based on a comparison of their individual greatest location and the best position of other particles. ith swarm particle location is expressed as $\bar{z}_i = (z_{i1}, z_{i2}, \cdots z_{iy})$, where $1 \le i \le x$ and x are the swarm particle size. ith swarm particle speed is expressed by $\bar{s}_i = (s_{i1}, s_{i2}, \cdots s_{iy})$, where $1 \le i \le y$ and y are searching space dimension of all swarm particle. The finest position in history for the ith swarm (Q_{best}) is $\bar{Q}i = (Q_{i1}, Q_{i2}, \cdots Q_{iD})$. The whole swarm optimal location (L_{best}) is $\bar{L}g = (L_{g1}, L_{g2}, \cdots L_{gx})$ $g \in \{1, 2, \cdots x\}$.

The following formulas can be used to update the particle's speed and location:

$$s_{iy}(t+1) = s_{iy}(t) + c_1 r_1 \left(Q_{iy}(t) - z_{iy}(t)\right) + c_2 r_2 \left(L_{gy}(t) - z_{iy}(t)\right),$$
$$z_{iy}(t+1) = z_{iy}(t) + s_{iy}(t+1). \tag{24}$$

Learning factors c_1 and c_2 are both positive constants, $z_{iy}(t)$ is the location vector associated with the ith particle, and $s_{iy}(t)$ is the related speed vector. This capacity to self-sufficiency and learning from one of the group's most talented members is what allows particles to get near to both their historical greatest position and the group's. In most cases, c_1 and c_2 are set to 2. Randomly distributed in the range [0,1] are the values of r_1 and r_2. Particles can only travel at a maximum speed of $S_{\max}$. For better algorithm astringency, Shi and Eberhart created the concept of inertia weight, which is shown in the following equation:

Figure 14: The experimental setup for hybrid horse herd particle swarm optimization-based improved tuning of PID controller speed controller for BLDC motor drive.

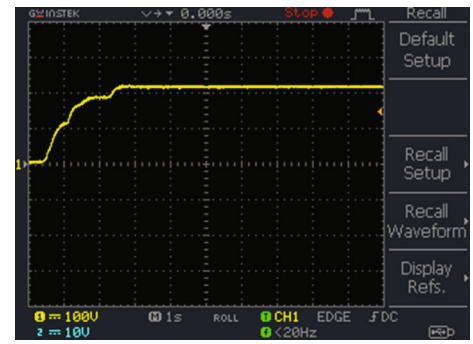
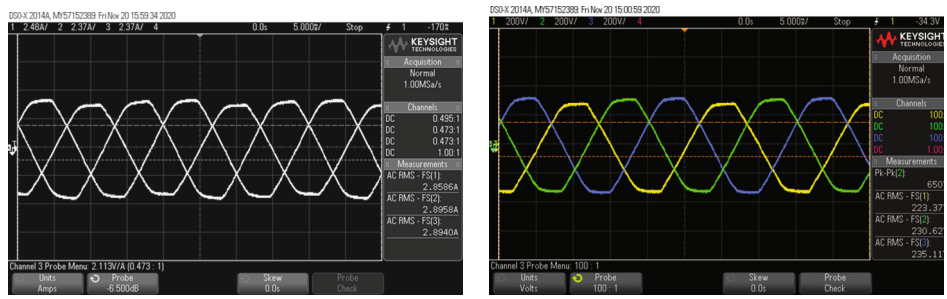


Figure 15: The DC bus voltage of PV battery system.

$$s_{iy}(t+1) = \alpha_\omega \times \left(s_{iy}(t) + c_1 r_1 \left(Q_{iy}(t) - z_{iy}(t)\right) + c_2 r_2 \left(L_{gy}(t) - z_{iy}(t)\right)\right). \tag{25}$$

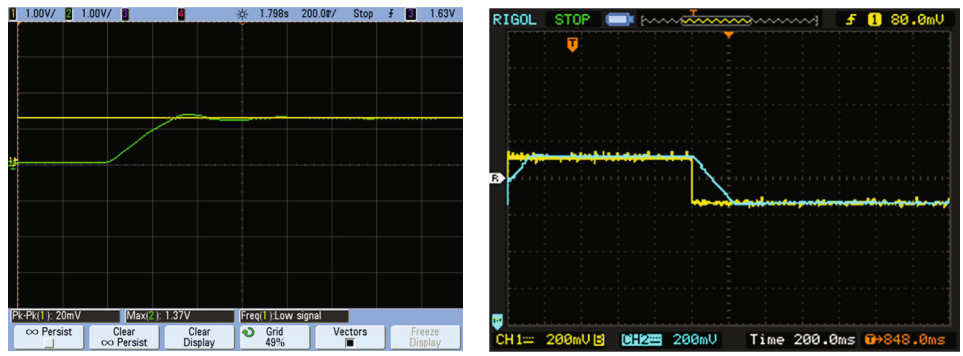
The inertia weight (α_ω) determines how much of the particle's current speed is inherited. If the particle is picked correctly, it will be able to exploit and develop at the same time.

3.3. Hybrid Horse Herd Particle Swarm Optimization. The goal of PSO is to find the optimal solution to a given problem. Compared to the PSO, the HHA has a quicker convergence rate. Based on the same population for both algorithms, the HHA and PSO each represent a unique solution. High performance may be achieved by combining these two algorithms in a hybrid approach that takes use of both their strengths and their distinctiveness. Hybrid horse herd particle swarm optimization for tuning of the PID controller parameter is shown in Figure 4. To begin, set the population, number of generations, social parameter, cognitive parameter, initial velocity, and initial weight for HHA and PSO, respectively. The PSO operator is used to find many viable solutions for the first population by reading the integral square error for the speed, current, and torque ripple of



(a) (b)

Figure 16: The performance analysis of three-phase (a) current and (b) voltage waveform.



(a) (b)

Figure 17: The performance analysis of change in speed: (a) case 1 and (b) case 2.

the brushless DC motor under various operating circumstances. The PSO's solutions make up the initial population of the HHA. HHA starts to fine-tune the parameter of the PID controller from best value obtained from the PSO, and after the final iteration, it provides optimal values for the PID controllers.

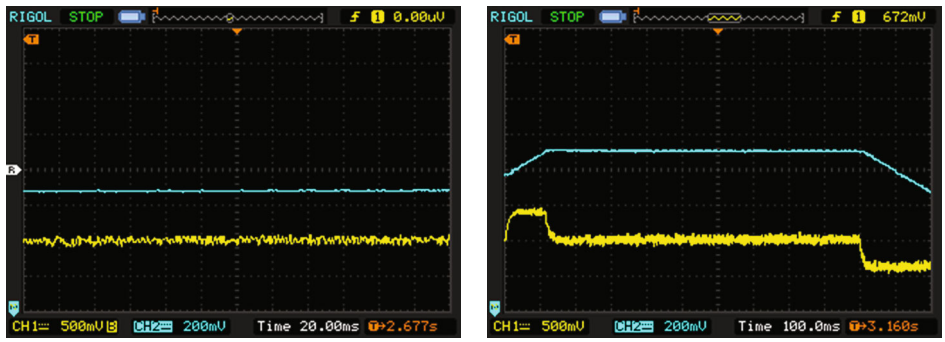
4. Results and Discussion

In this study, the PID controller parameters are optimized for regulating the speed and current of a brushless DC motor (BLDC) using a HHHPSO techniques. In this chapter, the effectiveness of the proposed HHHPSO technique is examined using simulation. The HHHPSO method has been tested on a computer with an Intel(R) core (TM) i5 CPU, 4 GB of RAM, and the MATLAB/Simulink (R2015a) environment. Figure 5 shows the simulation model of the HHHPSO-optimized PID-controlled BLDC motor. The difference between the BLDC motor's reference speed and its measured speed is used to determine the speed error. The PID control receives the speed error, and it is used to optimize the system's performance by minimising objective function in Equations (1) to (4). The HHHPSO technique regulates the proportional, integral, and derivative (PID) gains. The HHHPSO PID controller-assessed results are compared to those obtained from other controllers, such as SCA [21], GWA-FOPID [22], CASOA [23], SFSA [24], HGSOA [25], WOA [26], and FA [33].

4.1. Performance Analysis. The BLDC motor specification use for test the proposed HHHPSO techniques is shown in Table 1. The HHHPSO method is tested out on two different working conditions in simulation and compared to state-of-the-art standard methods such as SCA [21], GWA-FOPID [22], CASOA [23], SFSA [24], HGSOA [25], WOA [26], and FA [33]. The two working conditions are explained in the following sections.

4.1.1. Case 1: Constant Speed Condition. The performance of the proposed HHHPSO PID controller is examined with constant reference speed condition. The constant speed reference is shown in Figure 6. The reference speed of the BLDC motor is maintained at 1500 rpm from zero to two seconds. And corresponding results are presented in Figures 7 and 8.

Figure 7 depicts the motor's voltage, current, mechanical torque, and motor torque at a constant motor speed. The BLDC motor stator current is maintained at 0.55 A. The BLDC motor back EMF is maintained at 60 volts. Peak load torque is maintained at 1 Nm, and peak electromagnetic torque of the motor is maintained at 0.4 Nm for constant speed reference conditions. Figure 8 depicts the BLDC motor's inverter pulses, rotor angle, and output speed. From Figure 8(c), the BLDC motor speed reaches the 1500 rpm around 20 msec; it has no overshoot, zero steady-state error, and quick rise time.



(a) (b)

Figure 18: The performance torque analysis of (a) case 1 and (b) case 2.

4.1.2. Case 2: Variation of Speed Reference Condition. In this condition, motor speed reference is varied at random time step, and corresponding results are measured and analysed. The speed reference command variations are shown in Figure 9. The speed reference is maintained at 1000 rpm from 0 to 0.8 sec. The speed reference is maintained at 5000 rpm from 0.8 to 1.2 sec, and speed reference is maintained at 3000 rpm from 1.2 sec to 2 sec. The corresponding results are depicted in Figures 10 and 11.

Figure 10 depicts the voltage, current, mechanical torque, and motor torque as a function of changing motor speed. The BLDC motor stator current is maintained at 1.6 A. The BLDC motor back EMF is maintained at 50 volts. Load torque is maintained at 1 Nm, and electromagnetic torque of the motor is maintained at 1 Nm for varying speed reference conditions. Figure 11 depicts the inverter pulses, rotor angle, and output speed of the BLDC motor. Figure 11(c) displays the BLDC motor's output speed, which demonstrates the effectiveness of the HHHPSO-optimized PID controller; that is, speed of the BLDC motor is tracked effectively; it has quick rise time, settling time, zero overshoot, and low steady-state error from the simulation results under reference speed varying conditions.

4.2. Comparison Analysis. The HHHPSO-tuned PID controller is compared with SCA [21], GWA-FOPID [22], CASOA [23], SFSA [24], HGSOA [25], WOA [26], and FA [33] to demonstrate the efficacy of the proposed approach. PID controller gains are tuned using the HHHPSO technique. Optimizing the proportional, integral, and derivative gains improves the BLDC motor's performance by keeping the motor at a constant speed. The performance of the suggested technique is evaluated across two distinct scenarios, including both reference speed variation and constant motor speed. A comparison of the proposed method's optimal gain parameters with other considered approaches is shown in Table 2.

Table 3 shows a comparison of the transient response criteria such as maximum overshoot, rise time, and settling time are compared for proposed HHHPSO methods with other considered methods. From these table results, the HHHPSO-tuned PID controller has no overshoot, rise time around 21 msec, and settling time around 39 msec, but these parameters are not favour for other methods.

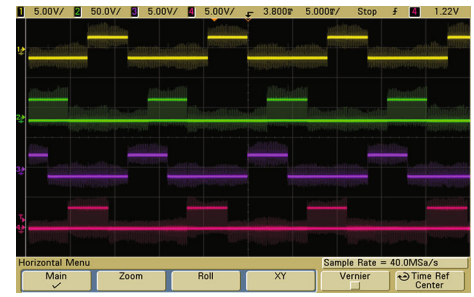


Figure 19: The performance analysis of inverter's PWM pulses.

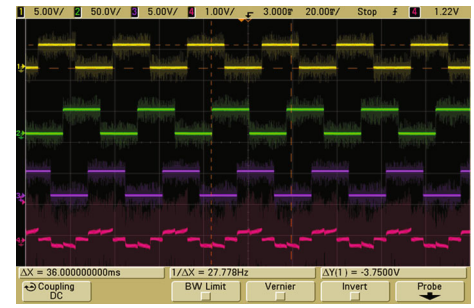
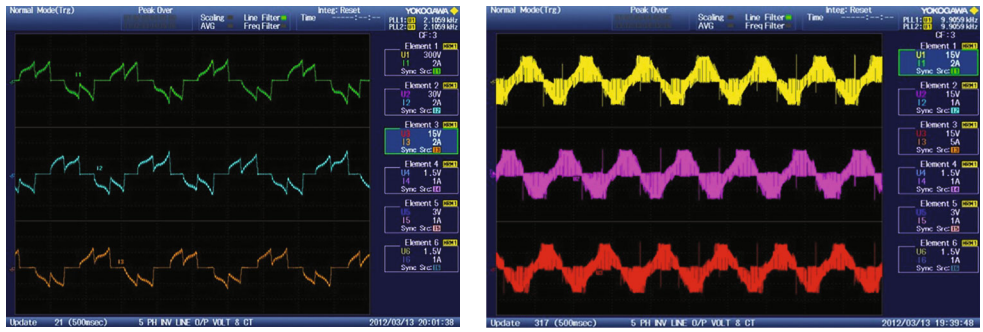


Figure 20: The performance analysis of Hall sensor and current signal.

Case 1. The comparative speed and torque response for the BLDC motor with HHHPSO and WOA and FA are analysed in this section for constant speed reference. Figure 12 shows a comparison response of the speed and torque with HHHPSO, WOA, and FA methods.

Figure 12(a) depicts the speed of the BLDC motor. The reference speed is 500 rpm maintained at all time. The FA algorithm attains the constant speed at 0.07 s. The WOA attains the constant at 0.075 s. The HHHPSO attains the 500 rpm at 0.05 s. The HHHPSO method only reaches the reference speed but WOA and FA having steady-state error of 10 to 20 rpm. Figure 12(b) depicts torque of the BLDC motor. The FA has the torque ripple value which is 10 Nm, and WOA has the torque ripple value which is 15 Nm. The HHHPSO has the torque ripple value which is approximately 3 or 5 Nm. From this analysis, HHHPSO-tuned



(a) (b)

Figure 21: The performance analysis of (a) current and (b) voltage signal.

PID controller has superior performance than WOA- and FA-tuned PID controller.

Case 2. The comparative speed and torque response for the BLDC motor with HHHPSO and WOA and FA are analysed in this section for speed reference varying conditions. The speed and torque response for HHHPSO, WOA, and FA are shown in Figure 13.

The speed response comparison with HHHPSO, WOA, and FA is depicted in Figure 13(a). Initially, speed reference is maintained at 100 rpm, and after 0.35 seconds, speed reference was changed to 700 rpm. The speed error is almost zero for the HHHPSO-tuned PID controller, but WOA- and FA-tuned PID controller has 10 to 20 rpm steady-state error in speed reference changing conditions also. Figure 13(b) depicts output torque of the BLDC motor with HHHPSO, WOA, and FA methods. The FA has the torque ripple value which is 10 Nm, and WOA algorithm has the torque ripple value which is 15 Nm. The HHHPSO-tuned PID controller has the torque ripple value which is approximately 3 or 5 Nm. From this analysis, HHHPSO-tuned PID controller has superior performance than WOA- and FA-tuned PID controller.

4.3. Experimental Results. Experimental tests of the proposed method are first simulated and then implemented in the laboratory to verify in various conditions. Figure 14 depicts the experimental setup for the improved tuning of a PID controller speed controller for a BLDC motor drive using a HHHPSO. Figure 15 shows the DC bus voltage of the PV battery system.

The performance analysis of three-phase current and voltage waveform is presented in Figure 16. The performance analysis of change in speed for case 1 and case 2 is presented in Figure 17.

The performance torque analysis of case 1 and case 2 is presented in Figure 18. The performance analysis of inverter's PWM pulses is illustrated in Figure 19. The performance analysis of Hall sensor and current signal is presented in Figure 20. The performance analysis of current and voltage signal is illustrated in Figure 21. The experimental results for speed regulation in the BLDC motor with HHHPSO-, WOA-, and FA-tuned PID controller tested in real time and corresponding details are presented in Figure 22. The results from the real-time analysis are

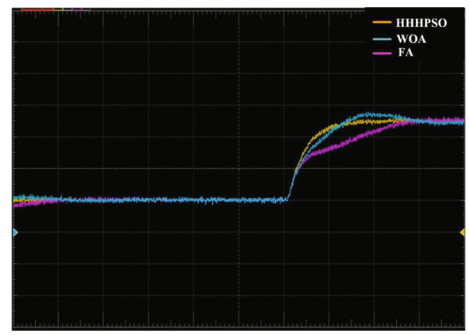


Figure 22: Experimental BLDC motor speed comparisons of HHHPSO-, WOA-, and FA-tuned PID controller.

matched with simulation results. From this analysis, the HHHPSO-tuned PID-controlled BLDC motor is performed well than WOA and FA methods and also suitable for real-time applications.

5. Conclusion

This work presents an approach for controlling the torque and speed of a PV and battery-powered BLDC motor using HHHPSO-tuned PID controller. The HHHPSO was used to optimize the parameter of the PID controller with reducing integral square of speed error, current error, and torque error. The proposed method has been created and tested in MATLAB simulation. The proposed method was tested for constant speed conditions and varying speed conditions and was also compared with the following methods: SCA, GWA-FOPID, CASOA, SFSA, HGSOA, WOA, and the FA. From the test results, the proposed HHHPSO-tuned PID controller has rise time of 20 to 21 msec, settling time of 35 to 39 msec, zero overshoot, and zero steady-state error in both operating case, but these parameters are not favour of other methods considered for comparisons. The hardware experimentation on the proposed HHHPSO-tuned PID-controlled PV battery-powered BLDC motor is presented and performed well in real time also. The proposed HHHPSO can be applied to tune the parameter of fuzzy logic control of PV battery-powered BLDC motor, and this is considered to be the scope for future work.

Data Availability

Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] P. Yedamale, "Brushless DC (BLDC) motor fundamentals," *Microship Technology Inc*, vol. 20, pp. 3–15, 2003.

[2] R. Nadolski, K. Ludwinek, J. Staszak, and M. Jaskiewicz, "Utilization of BLDC motor in electrical vehicles," *Przeglad Elekrotechniczny*, vol. 88, no. 4, pp. 180–186, 2012.

[3] C. P. Swamy, B. Singh, and B. P. Singh, "Investigations on dynamic behavior of permanent magnet brushless DC motor drive," *Elecric Machines and Power System*, vol. 23, no. 6, pp. 689–701, 1995.

[4] C. W. Hung and C. L. Hsu, "A simulation model of Hall sensor misalignment in BLDC motors," *Journel of Robatics, Nteworking and Artificial Life*, vol. 3, no. 3, pp. 152–157, 2016.

[5] S. Ushakumari, P. C. Nair, and R. Sankaran, "Closed loop performance of a permanent magnet brushless dc motor incorporating the nonlinearity in torque-balance equation (transient operation)," *Electric Power Componments and Systems*, vol. 30, no. 12, pp. 1249–1260, 2002.

[6] K. Xia, J. Lu, C. Bi, Y. Tan, and B. Dongs, "Dynamic commutation torque-ripple reduction of brushless DC motor based on quasi-Z-source net," *IET Electric Power Applications*, vol. 10, no. 9, pp. 819–826, 2016.

[7] K. Premkumar and B. V. Manikandan, "Bat algorithm optimized fuzzy PD based speed controller for brushless direct current motor," *Engineering Science and Technology, an International Journal*, vol. 19, no. 2, pp. 818–840, 2016.

[8] M. Tariq, T. K. Bhattacharya, N. Varshney, and D. Rajapan, "Fast response Antiwindup PI speed controller of brushless DC motor drive: modeling, simulation and implementation on DSP," *Journal of Electrical System and Information Technology*, vol. 3, no. 1, pp. 1–13, 2016.

[9] A. Varshnev, D. Gupta, and B. Dwivedi, "Speed response of brushless DC motor using fuzzy PID controller under varying load condition," *Journal of Electrical System and Information Technology*, vol. 4, no. 2, pp. 310–321, 2017.

[10] M. Sumega, S. Zossak, P. Varecha, and P. Rafadus, "Sources of torque ripple and their influence in BLDC motor drives," *Transportation Researches Procedia*, vol. 40, pp. 519–526, 2019.

[11] M. Nasri, H. Nezamabadi-Pour, and M. Maghfoori, "A PSO-based optimum design of PID controller for a linear brushless DC motor," *World Acadamy of Science, Enginnering and Technology*, vol. 26, no. 40, pp. 211–215, 2017.

[12] G. R. Sekhar and B. Banakara, "An internal current controller BLDC motor drive supplied with PV fed high voltage gain DC-DC converter," *International Jounnal of Elecrical and Computer Engineering*, vol. 8, no. 2, p. 1262, 2018.

[13] R. G. Rajesh and C. Blalaji, "Speed control of BLDC motor using PID controller," *International Journal of Advanced Research in Electrical, Electronics and Sustainable Development*, vol. 22, no. 1, pp. 34–46, 2014.

[14] T. K. Hassan and M. B. Abdulmelik, "Optimal controller for vector-controlled multilevel invereter fed brushless DC motor," *Journal of Engineering and Sustainable Development*, vol. 2018, no. 1, pp. 34–46, 2018.

[15] A. B. Yildiz, "Electrical equivalent circuit based modeling and analysis of direct current motors," *International Journal of Electrical Power & Enengy Systems*, vol. 43, no. 1, pp. 1043–1047, 2012.

[16] A. A. El-samahy and M. A. Shamseldin, "Brushless DC motor tracking control using self-tuning fuzzy PID control and model reference adaptive control," *Ain Shams Engineering Journal*, vol. 9, no. 3, pp. 314–352, 2018.

[17] D. Potnuru, K. A. Mary, and C. S. Babu, "Experimental implementation of flower pollination algorithm for speed controller of a BLDC motor," *Ain Shams Engineering Journal.*, vol. 10, no. 2, pp. 287–295, 2019.

[18] E. Celik and H. Gor, "Enhanced speed control of a DC servo system using PI + DF controller tuned by stochastic fractal search technique," *Journal of the Franklin Institute*, vol. 356, no. 3, pp. 1333–1359, 2019.

[19] M. Divandari, B. Rezaie, and A. R. Noei, "Speed control of switched reluctance motor via fuzzy fast terminal sliding-mode control," *Computers& Electrical Enginnering*, vol. 80, article 106472, 2019.

[20] M. G. Gan, M. Zhang, C. Y. Zheng, and J. Chen, "An adaptive sliding mode observer over wide speed range for sensorless control of a brushless DC motor," *Control Engineering Practice*, vol. 77, pp. 52–62, 2018.

[21] J. Agarwal, G. Parmar, and R. Gupta, "Application of sine cosine algorithm in optimal control of DC motor and robustness analysis," *Wulfenia Journal*, vol. 24, pp. 77–95, 2017.

[22] S. Ekinci, B. Hekimoglu, and D. Izci, "Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor," *An International Journal Engineering Science and Technology*, vol. 24, no. 2, pp. 331–342, 2021.

[23] R. Bhatt, G. Parmar, R. Gupta, and A. Sikander, "Application of stochastic fractal search in approximation and control of LTI systems," *Microsystem Technologies*, vol. 25, no. 1, pp. 105–114, 2019.

[24] B. Hekimoglu, "Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm," *IEEE Access*, vol. 7, pp. 38100–38114, 2019.

[25] J. Agarwal, G. Parmar, R. Gupta, and A. Sikander, "Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor," *Microsystem Technologies*, vol. 24, no. 12, pp. 4997–5006, 2018.

[26] M. A. Labbaf Khaniki, S. Esfandiari, and M. Manthouri, "Speed control of brushless DC motor using fractional order fuzzy PI controller optimized via WOA," in *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 431–436, Mashhad, Iran, 2020.

[27] S. Ekinci, D. Izci, and B. Hekimoğlu, "Optimal FOPID speed control of DC motor via opposition-based hybrid manta ray foraging optimization and simulated annealing algorithm," *Arabian Journal for Science and Engineering*, vol. 46, no. 2, pp. 1395–1409, 2021.

[28] D. Izci, S. Ekinci, H. L. Zeynelgil, and J. Hedley, "Performance evaluation of a novel improved slime mould algorithm for direct current motor and automatic voltage regulator systems," *Transactions of the Institute of Measurement and Control*, vol. 44, no. 2, pp. 435–456, 2022.

[29] D. Izci, H. Serdar Ekinci, L. Zeynelgil, and J. Hedley, "Fractional order PID design based on novel improved slime mould algorithm," *Electric Power Components and Systems*, vol. 49, no. 9-10, pp. 901–918, 2021.

[30] D. İzci and S. Ekinci, "Comparative performance analysis of slime mould algorithm for efficient design of proportional–integral–derivative controller," *Electrica*, vol. 21, no. 1, pp. 151–159, 2021.

[31] E. Eker, M. Kayri, S. Ekinci, and D. Izci, "A new fusion of ASO with SA algorithm and its applications to MLP training and DC motor speed control," *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 3889–3911, 2021.

[32] D. Izci, "Design and application of an optimally tuned PID controller for DC motor speed regulation via a novel hybrid Lévy flight distribution and Nelder–Mead algorithm," *Transactions of the Institute of Measurement and Control*, vol. 43, no. 14, pp. 3195–3211, 2021.

[33] C. K. Karthika and P. Abraham, "Speed and torque control of BLDC motor using firefly algorithm technique," *AIP Conference Proceedings*, vol. 2222, article 040013, 2020.

[34] F. M. Naeimi, "Horse herd optimization algorithm: a nature-inspired algorithm for high- dimensional optimization problems," *Knowledge-Based Systems*, vol. 213, article 106711, 2021.