

Article

# Black Widow Optimization-Based Optimal PI-Controlled Wind Turbine Emulator

K. Premkumar <sup>1</sup>, M. Vishnupriya <sup>2</sup>, Thanikanti Sudhakar Babu <sup>3,\*</sup>, B. V. Manikandan <sup>4</sup>,  
T. Thamizhselvan <sup>1</sup>, A. Nazar Ali <sup>1</sup>, Md. Rabiul Islam <sup>5,\*</sup>, Abbas Z. Kouzani <sup>6</sup>  
and M. A. Parvez Mahmud <sup>6,\*</sup>

<sup>1</sup> Department of EEE, Rajalakshmi Engineering College, Tamilnadu 602105, India; prem.kamaraj@gmail.com (K.P.); thamizhselvan.t@rajalakshmi.edu.in (T.T.); nazarali.a@rajalakshmi.edu.in (A.N.A.)

<sup>2</sup> Department of ECE, Saveetha School of Engineering, Tamilnadu 602105, India; vish0790@gmail.com

<sup>3</sup> Institute of Power Engineering, Department of Electrical Power Engineering, Universiti Tenaga Nasional, Selangor 43000, Malaysia

<sup>4</sup> Department of EEE, Mepco Schlenk Engineering College, Tamilnadu 626005, India; bvmani@mepcoeng.ac.in

<sup>5</sup> School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia

<sup>6</sup> School of Engineering, Deakin University, Geelong, VIC 3216, Australia; abbas.kouzani@deakin.edu.au

\* Correspondence: sudhakarbabu66@gmail.com (T.S.B.); mrislam@uow.edu.au (M.R.I.); m.a.mahmud@deakin.edu.au (M.A.P.M.); Tel.: +91-738-216-3424 (T.S.B.)

Received: 27 November 2020; Accepted: 8 December 2020; Published: 11 December 2020



**Abstract:** In this article, the parameters of the proportional-integral (PI) controller of the wind turbine (WT) emulator, i.e., proportional and integral gain of the PI controller, are optimized using a black widow optimization algorithm (BWOA). The proposed system is developed and analyzed using MATLAB/Simulink environment. The performance of the BWOA optimized PI controller is compared with a BAT algorithm, particle swarm optimization, and genetic algorithm optimized PI controller to measure the effectiveness of the proposed control system. The developed system is tested for different operating conditions such as static wind speed settings, static pitch angle conditions, step-change in wind speed settings, and step-change in pitch angle settings. Finally, the proposed system is realized in real-time by hardware experimentations. The results of the experimentation are compared with simulation results as well. The presented simulation and hardware result shows good agreement, which confirms the effectiveness of the proposed method. Thereby, the proposed optimization-based PI-controlled wind emulator can be recommended for emulating the characteristics of any type of WT with a low-cost system.

**Keywords:** BAT algorithm; black widow optimization algorithm; genetic algorithm; particle swarm optimization; proportional-integral (PI) controller; wind turbine emulator

## 1. Introduction

Wind power generation, integration, and control have gained more attention among researchers globally to achieve global energy sustainability [1]. Typically, wind power generation is the second-largest power energy source in the world [2]. More research is ongoing in maximum power extraction of wind power generation and grid integration [3]. In various applications, such as wind-powered battery charging, wind-powered electric vehicle charging stations, and in similar domains, researchers face an uphill task in testing maximum power tracking algorithms in the matching design of the battery charging controller and control of electric vehicle charging in real-time. Due to the exorbitant cost of wind turbines (WT), the stochastic nature of wind, and numerous

uncountable associated problems in this research arena have made the researchers unable to test their developed algorithms under all specified possible conditions [4]. To overcome this problem, a WT emulator has been developed for testing the developed algorithms. Further, the emulator's necessity and its wide range of applications were clearly described in [5,6]. Any WT emulator should emulate the three regions of the WT, namely, pitch angle control region, maximum power point tracking region, and parking region [7]. The WT emulator was developed and tested with two modes, namely simulation model testing and real-time model testing.

The literature reviews on such a WT emulator are as follows: the WT characteristics were realized using a separately excited direct current motor [8]. The digital signal processor was used to implement the separately excited direct current motor-based WT emulator's proportional-integral control logic. In [9], the WT characteristic was emulated via an economical direct current motor drive. In this, the dSPACE hardware board was used to implement the WT reference model with PI control. The PIC-controlled WT emulator was presented in [10]. In this, a simple gain PI controller was implemented to mimic the characteristics of the real-time WT. The WT characteristics were emulated with the help of the DC motor by authors in [11]. The wind reference model output depends on the wind velocity and DC motor speed. The personal computer with supervisory control was used to implement the control logic of the emulator. The emulation efficiency of the system was around 88% only. In [12], the DC motor was controlled by armature voltage control and field control to realize the WT characteristics. The PI control algorithm was implemented in armature voltage control, and proportional control action was used in the field flux control. The WT reference model generates the reference command based on wind velocity and speed of the motor. Moreover, emulation efficiency was about 90%.

In [13], real-time WT characteristics were emulated via a separately excited DC motor (SEDCM). The analog circuit concept was used to build the WT model mathematically. The PI control of the WT emulator control logic was optimized using constrained optimization. WT emulator was realized using SEDCM in [14]. The WT reference model was generating the reference command based on the following parameters: the speed of the motor, wind velocity, and pitch angle. The torque of the motor was controlled using PI control. The emulation efficiency system was around 91%. In [15], WT characteristics were emulated using PI controlled DC motor. The WT reference torque was generated based on the DC motor's rotor speed, wind velocity, and pitch angle. The DS1104 hardware control board was used to implement the WT reference model and PI control.

In [16], an open-loop DC motor control was developed to resemble the features of the WT. Open-loop armature voltage control and field flux control was adapted to mimic the behavior of the WT. The DC motor's voltage was controlled by power resistors, and it produced more power losses in the power resistor. It degrades the overall efficiency of the system. In [17], DC motor-based WT emulator and its associate control were implemented using MATLAB simulation. The WT reference model generates the command current signal based on wind, DC motor speed, and pitch angle. The PI control has been used to control the direct current motor's torque to emulate the WT characteristics. In [18], a fuzzy-based proportional derivative controlled WT emulator was introduced. DC motor speed and torque were controlled by using fuzzy logic control. The fuzzy logic controller was compared with conventional PI control. The emulation efficiency of the system with fuzzy logic control was around 92%. The permanent magnet direct current motor (PMDC) based WT emulator was implanted in [19]. The simulation and hardware realization of the WT emulator was studied. PI torque control scheme was presented to control the characteristics of the PMDC motor. The WT characteristics were emulated using a four-quadrant chopper-controlled direct current motor in [20]. LabVIEW software was used to implement the WT emulator control logic in the microcontroller hardware control board.

Speed and current control were implemented using the PI control algorithm. The emulation efficiency of the system is around 91% only. In [21], an open-loop field-programmable gate array (FPGA) control DC motor was developed to resemble WT characteristics. The PWM pulse for the DC-DC converter fed DC motor was generated based on wind velocity and WT reference model

program of the FPGA control board. The WT emulator for the stand-alone application was developed in [22]. The WT characteristics were emulated using closed-loop control of SEDCM. The simulation study has been implemented for a WT emulator using MATLAB. PI current control logic of the WT emulator was implemented in the FPGA hardware control board, and it has been employed for the DC motor control.

The controller was used for the DC motor control from the above-reported literature, and mostly PI current control has been employed. Moreover, accurate tuning of gain parameters such as the PI control's proportional and integral value is not reported in the literature for useful emulation of WT characteristics in all regions. Invariably the WT emulator control algorithm was implemented using dSPACE, FPGA, and digital signal processor (DSP). These hardware controllers are costlier and also present complexity in the implementation. It is identified that scope is abundant in developing WT emulator considering the following objectives: to emulate the WT emulator in three operating regions and achieve effective emulation of a WT in all regions by using a black widow optimization algorithm optimized PI controller. The black widow optimization algorithm optimized PI controller is compared with the genetic algorithm (GA), BAT algorithm, and particle swarm optimization (PSO) to optimize the PI current controller and realize WT emulator control algorithm using a low-cost microcontroller board.

The main contributions of the article can be given as follows:

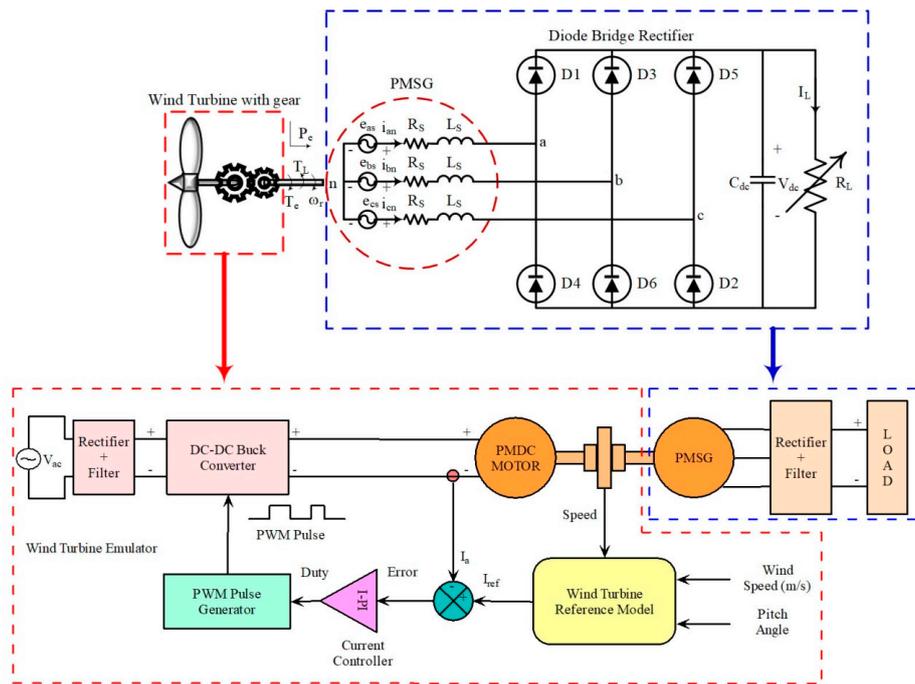
- Proposed an effective control method to design and analysis of WT emulator.
- The gains of the PI controller were effectively optimized using a black widow optimization algorithm.
- Optimized PI controlled DC–DC buck converter fed permanent magnet direct current motor with a low-cost microcontroller is proposed.
- The prototype of the proposed WT emulator is developed, and obtained results are compared with the results acquired via simulation.
- The proposed emulator's efficiency has been tested for three different cases to assess the developed WT emulator's superiority.

The paper's organization as follows: design and analysis of WT emulator using PI controlled DC–DC converter fed PMDC motor is presented in Section 2. Section 3 presents the black widow optimization algorithm (BWOA), GA, PSO, and BAT algorithm optimization for PI controlled WT emulator. Simulation results and discussions are outlined in Section 4, and in Section 5, explanations about hardware implementation are presented. Concluding remarks are provided in Section 6.

## 2. Modeling of WT Emulator

The mathematical modeling of WT, DC–DC buck converter, PMDC motor, and PI controller is discussed in the present section. Figure 1 depicts the arrangement of the typical WT emulator. The WT emulator comprises of DC supply, DC–DC buck converter, PMDC motor, WT reference model, permanent magnet synchronous generator (PMSG), current controller, PWM pulse generator, and electrical load.

The WT reference model receives three inputs, i.e., wind speed, pitch angle, and speed of the PMDC motor. Based on these inputs, the WT model generates the reference current for the next stage. The reference current from the WT reference model and the armature current of the PMDC motor are compared, and it is processed through a proportional-integral current controller. The PI current controller generates the duty cycle for the next stage based on the error current. The duty cycle is processed through a PWM pulse generator to generate the PWM pulse for the DC–DC buck converter switch. The converter voltage and current vary based on the duty cycle, and it is used to control the PMDC motor characteristics. The PMDC motor emulates the WT reference model characteristics. In this WT emulator control structure, the PI controller parameters, such as proportional ( $K_P$ ) and integral ( $K_I$ ) gains, are optimized using a black widow optimization algorithm.



**Figure 1.** Wind turbine (WT) Emulator using proportional-integral (PI) controlled DC–DC converter fed permanent magnet direct current motor.

2.1. Modeling of WT Reference Model

The mathematical model of WT is the heart of the WT emulator. The WT’s mathematical model provides the reference torque or current for the current controller to mimic the characteristics of the WT using a DC–DC buck converter fed PMDC motor. The WT torque is related to the speed of the PMDC motor, pitch angle, and wind velocity, and it is expressed in Equation (1):

$$T_{WT} = \frac{C_p(\lambda, \beta) \times \rho \times \pi \times R^3 \times V_W^2}{2 \times \lambda} \tag{1}$$

where  $C_p(\lambda, \beta)$  is WT power coefficient,  $V_W$  is the wind velocity or speed (m/s),  $R$  is the turbine radius (m),  $\rho$  is the air density ( $\text{kg/m}^3$ ),  $\beta$  is the pitch angle (deg), and  $\lambda$  is the tip speed ratio. The WT power coefficient is related to pitch angle and tip speed ratio, and it is expressed in Equation (2):

$$C_p(\lambda, \beta) = C_1 \left( \frac{C_2}{\lambda_i} - C_3 \times \beta - C_4 \right) \exp \left( \frac{-C_5}{\lambda_i} \right) + C_6 \times \lambda \tag{2}$$

with,

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08 \times \beta} - \frac{0.035}{\beta^3 + 1}$$

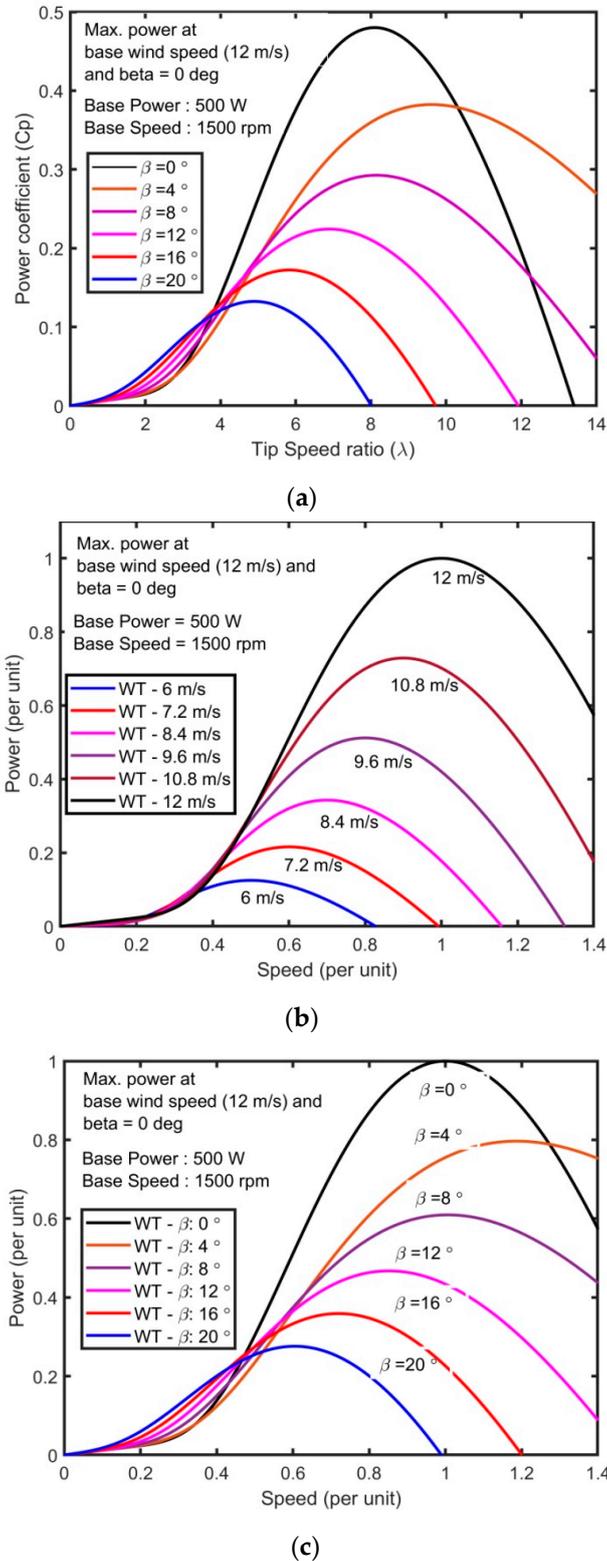
where  $C_1, C_2, C_3, C_4, C_5,$  and  $C_6$  are the empirical power coefficients, and the values for these coefficients are 0.5176, 116, 0.4, 5, 21, and 0.0068, respectively.  $\lambda_i$  is the tip speed ratio at  $i^{\text{th}}$  step time. Figure 2a shows the WT power coefficient’s response concerning tip speed with different pitch angles. From this figure, the maximum power coefficient  $C_{pmax}$  is 0.48 for pitch angle equal to  $0^\circ$  and normalized tip speed ratio ( $\lambda_{nom}$ ) equal to 8.1 [23].

The tip speed ratio and power of the WT model being expressed in Equations (3) and (4), respectively,

$$\lambda = \frac{N \times R}{V_W} \tag{3}$$

$$P_{WT} = C_p(\lambda, \beta) \times \frac{\rho \times A}{2} \times V_W^3 \tag{4}$$

where  $A$  is the turbine swept area ( $m^2$ ), and  $N$  is the speed of the generator (rad/s). In this work, the rated power of the WT is fixed at 500 W; base wind speed is set at 12 m/s. Figure 2b shows the response of WT power concerning the speed of the WT at different wind speed profiles, and Figure 2c shows the response of WT power concerning the speed of the WT at different pitch angles.



**Figure 2.** Characteristics of WT: (a)  $C_p$ - $\lambda$  characteristics for different pitch angle, (b) power-speed characteristics at different wind speed profile, and (c) power-speed characteristics at different pitch angle.

## 2.2. Modeling of the DC–DC Buck Converter

The buck converter can convert the fixed DC voltage into a variable DC output voltage based on the buck converter's duty cycle. The relation between duty cycle and an output voltage of the buck converter can be expressed as [12]:

$$\frac{V(S)}{D(S)} = \frac{V_S}{1 + S^2 \cdot LC + S \cdot \frac{L}{R_L}} \quad (5)$$

where  $V(s)$  is the converters output voltage,  $V_S$  is the input voltage,  $D(s)$  is the duty cycle, and  $R_L$  is a resistive load of the converter in ohms. The specification of parameters used for the DC–DC buck converter is: inductance ( $L$ ) is 10 mH, capacitance ( $C$ ) is 4800  $\mu$ F, switching frequency is 3 kHz, the voltage rating is 240 V, and the power rating is 750 W.

## 2.3. Modeling of the PMDC Motor

The PMDC motor is used to convert electrical energy into useful mechanical energy based on its input voltage. The transfer functions related to the mechanical and electrical parameters of the motor, such as speed, input voltage, and armature current can be expressed as [12]:

$$\frac{N(s)}{V(s)} = \frac{K_t}{JL_a S^2 + (JR_a + BL_a)S + (BR_a + K_b K_t)} \quad (6)$$

$$\frac{I_a(s)}{V(s)} = \frac{(JS^2 + BS)}{(R_a + L_a S)(JS^2 + BS) + K_b K_t S} \quad (7)$$

$$\frac{N(S)}{I_a(S)} = \frac{SK_t}{(JS^2 + BS)} \quad (8)$$

where  $V(s)$  is the input voltage of the motor,  $I_a(s)$  is the armature current of the motor,  $N(s)$  is the speed of the motor,  $B$  is the frictional coefficient of the motor,  $J$  is the inertia of the motor,  $L_a$  is the armature inductance of the motor,  $R_a$  is the armature resistance of the motor,  $K_t$  is the torque constant of the motor and  $K_b$  is the back emf constant of the motor. The specification used for this motor is:  $J$  is 0.02215 kg-m<sup>2</sup>,  $B$  is 0.002953 N-m/(rad/s),  $R_a$  is 2.581 ohms,  $L_a$  is 0.028 H,  $K_b$  is 0.08 V/rpm,  $K_t$  is 1 N-m/A, the power rating is 700 W, and voltage rating is 220 V.

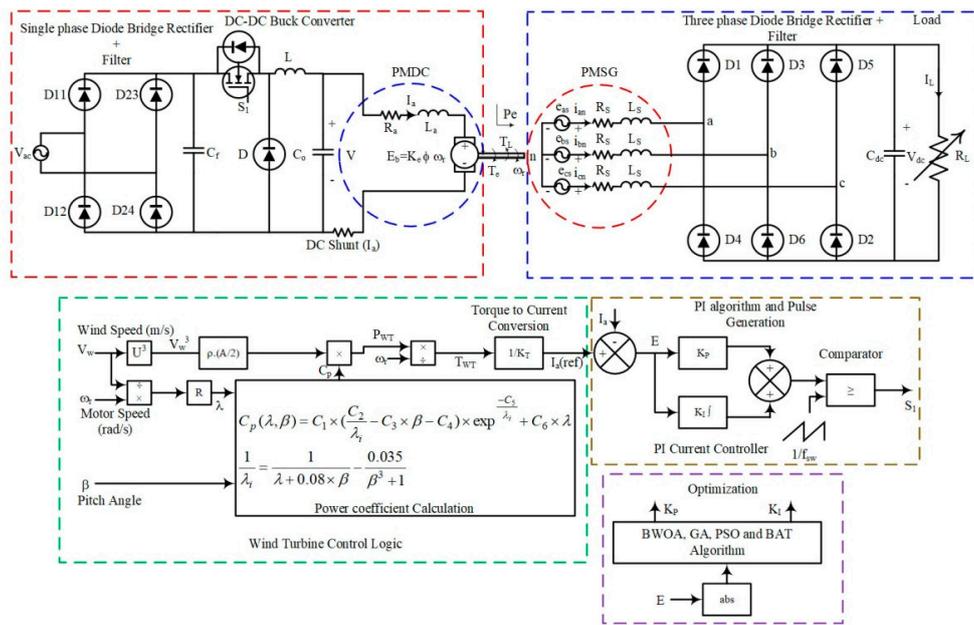
## 2.4. Modeling of the PI Controller

The transfer function model of the proportional-integral controller used for the WT emulator can be expressed as:

$$\frac{D(S)}{\Delta I_a(S)} = K_P + \frac{K_I}{S} \quad (9)$$

$$\Delta I_a(S) = I_{ref}(S) - I_a(S) \quad (10)$$

where  $K_P$ ,  $K_I$  is proportional gain and integral gain of the PI controller, respectively. The error current is denoted by  $\Delta I_a(s)$ . Generally, the PI controller's gain parameter is tuned by the trial and error method and the ZN method. However, these methods had some disadvantages, such as the trial and error method, which takes more time to find the PI controller's optimal value; the ZN method only provides the initial guess for the PI controller's gain parameter. To overcome this problem, optimization techniques may be utilized to adjust the parameter of the PI controller. In this work, a new optimization algorithm, i.e., a black widow optimization algorithm, is used to optimize the parameter of the PI current controller of the emulator to emulate the exact characteristics of the reference WT mathematical model. Figure 3 shows the overall block diagram of the PI controlled DC–DC converter fed PMDC motor-based WT emulator.



**Figure 3.** Block diagram of the PI controlled DC–DC buck converter fed permanent magnet direct current (PMDC) motor-based WT emulator.

### 3. Tuning of PI Current Controller of WT Emulator Using BWOA

The proposed near accurate WT emulator using a black widow optimization algorithm optimized PI controlled DC–DC converter fed permanent magnet direct current (PMDC) motor is presented in this section. The PI current controller parameters such as proportional gain ( $K_p$ ) and integral gain ( $K_i$ ) of the WT emulator is optimized using the black widow optimization algorithm (BWOA). The black widow optimization algorithm is imitating the lifestyle of the evolution of the black widow spider. Generally, female black widow spiders make the net during the night and leave some pheromone in some place of her net to attract male black spiders to have mating. Male black widow spiders get attracted by this pheromone and join in the net. The female black widow spider eats the male black widow spider after or during mating. After mating, the female black widow lays egg socks on the net. After 11 days, young spiders come out from the eggs, and these spiders will participate in sibling cannibalism. The young spiders stay in the mother’s net for a short period, and interestingly, the mother even eats some young spiders during this brief period sometimes. Other young spiders from the net are considered the fittest young spiders based on this concept; this black widow optimization algorithm is developed [24].

The black widow optimization proceeds with a random initial black widow spider population. This population has male and female black widow spiders for generating offspring for the next generation. The initial population of black widow spiders can be expressed as

$$X_{N,d} = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,d} \\ & & \vdots & & \\ x_{N,1} & x_{N,2} & x_{N,3} & \cdots & x_{N,d} \end{bmatrix} \quad (11)$$

$lb \leq X_i \leq ub$

where  $X_{N,d}$  is the population of black widow spiders,  $d$  is the number of decision variables,  $N$  is the number of population,  $lb$  is the lower bound of the population, and  $ub$  is the upper bound of the population. The potential solution populations ( $X_{N,d}$ ) are used to minimize or maximize the following objective function represented in Equation (12):

$$Objective\ function = f(X_{N,d}) \quad (12)$$

The next process in the black widow optimization algorithm is reproducing the young spiders from male and female spiders' mating. During or after mating, male spiders may be eaten by female spiders. The random selection process is used to select the pair of spiders for mating to reproduce young spiders. The reproduction process of the black widow optimization is expressed by the equation given in Equation (13):

$$\begin{aligned} Y_{i,d} &= \beta \times X_{i,d} + (1 - \beta) \times X_{j,d} \\ Y_{j,d} &= \beta \times X_{j,d} + (1 - \beta) \times X_{i,d} \end{aligned} \quad (13)$$

where  $Y_{i,d}$ , and  $Y_{j,d}$  are the young spiders from reproduction,  $i$  and  $j$  are a random number between 1 to  $N$  and  $\beta$  is the random number between 0 to 1. To avoid random duplication selection of pairs, the reproduction process is carried out for  $d/2$  times.

After reproduction, the mother spider population and young spider population are sorted with their fitness function value and cannibalism rate. During the optimization process, three cannibalism procedures are considered. The first one is sexual cannibalism, where the female spider population eats the male spider population during or after mating. This concept can be applied to the fitness value of the female and male spider populations. The second one is sibling cannibalism, where the healthy young spiders eat the weaker young spiders. This concept is applied using cannibalism rate, and with this rate only, fittest young spiders remain in the population, and others are discarded from the population. In the third type of cannibalism, the young spiders eat their mother. This concept is applied based on the fitness value of the mother spider and young spiders.

The next process in the black widow optimization is mutation. The young spiders are selected based on the mutation rate, and a small random value is added with selected young spiders for mutation; and this process is expressed in Equation (14):

$$Z_{k,d} = Y_{k,d} + \alpha \quad (14)$$

where  $Z_{k,d}$  is the mutated spider population,  $Y_{k,d}$  is the randomly selected young spider,  $k$  is the random number, and  $\alpha$  is the random mutate value.

This black widow optimization algorithm depends on three parameters, namely the reproduction rate (RP), cannibalism rate (CP), and mutation rate (MR). The reproduction rate controls the generation of young spiders and provides opportunities to explore the search space for finding a better solution. The cannibalism rate controls the weaker fittest population in the generation, and only the stronger fittest populations are allowed for the next generation. The mutation rate controls the diversity in the current generation to the next generation. The flowchart for the BWOA is shown in Figure 4.

The optimization of the PI controller using BWOA is shown in Figure 5. The BWOA receives the absolute current error as a fitness value from the WT emulator. BWOA optimizes the proportional gain and integral gain to minimize the absolute error in the system. To verify the black widow optimization method's effectiveness, it is compared with the BAT algorithm, particle swarm optimization, and genetic algorithm. The parameter used for the GA, PSO, and BWOA is shown in Table 1.

The GA, PSO, BAT, and BWOA optimization is implemented in the R2017b MATLAB software and tested in an i3-4005U CPU, 1.7 GHz personal computer. These algorithms have been executed for 100 runs and 4040 fitness value evaluated in a single run, i.e., 40 population with 100 generation, which equals 4000 fitness value evaluation plus fitness value evaluation for an initial 40 random population. Optimization algorithm results such as optimal gain parameter, best fitness, worst fitness, standard deviation, mean fitness value, and average computation time are taken for 100 trials with GA, PSO, BAT, and BWOA are presented in Table 2. The convergence plots for these algorithms are provided in Figure 6.

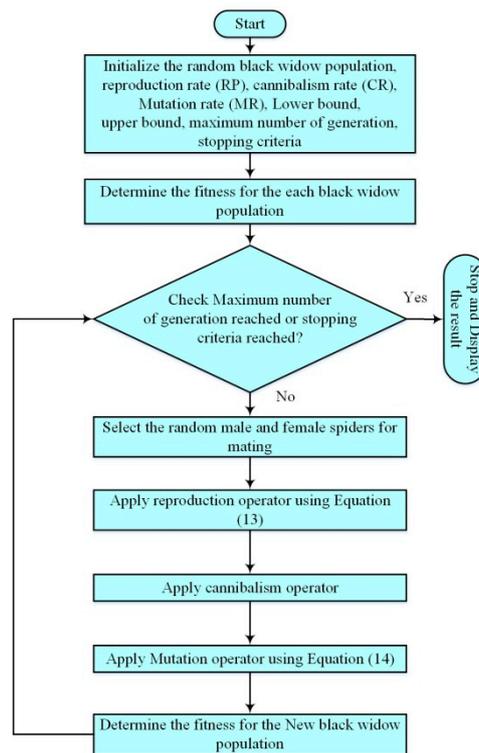


Figure 4. Flowchart for black widow optimization algorithm (BWOA).

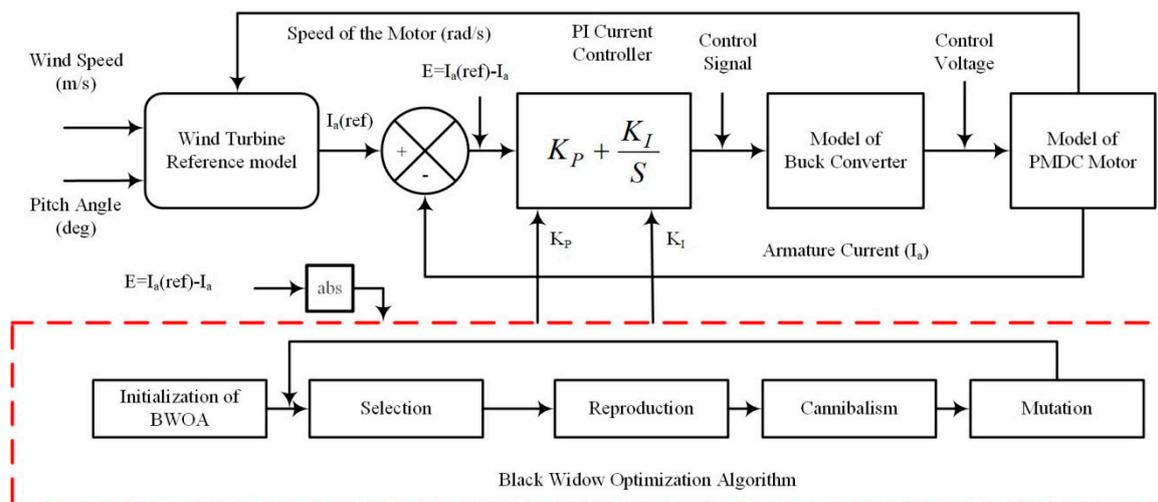


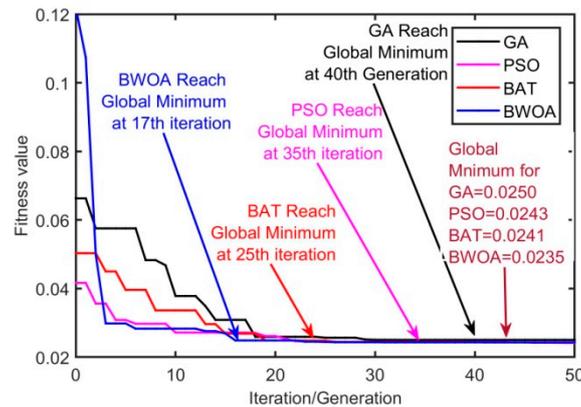
Figure 5. Optimization of PI controller using BWOA.

Table 1. Specification of BWOA, genetic algorithm (GA), particle swarm optimization (PSO) and BAT algorithm.

Black Widow Optimization Algorithm		Genetic Algorithm		Particle Swarm Optimization		BAT Optimization	
Spider Size	40	Population Size	40	Swarm Size	40	BAT Size	40
Generations	100	Generations	100	Iterations	100	Iterations	100
Reproduction rate	0.6	Crossover rate	0.7	Cognitive factor (C1)	1.2	Lower limit of the frequency (fmin)	0
Cannibalism rate	0.44	Mutation Rate	0.3	Social factor (C2)	1.2	Upper limit of the frequency (fmax)	100
Mutation rate	0.4			Lower limit of the inertia weight (Wmin)	0.1	Loudness factor at starting (R0)	0.9
				Upper limit of the inertia weight (Wmax)	0.9	Pulse rate at starting (P0)	0.9

**Table 2.** Results of the BWOA, GA, PSO and BAT algorithm.

Algorithm	$K_P$	$K_I$	Best Fitness	Worst Fitness	Standard Deviation	Mean	Mean Computation Time (s)
GA	0.0230	0.0663	0.0250	0.47356	0.0160	0.0299	339
PSO	0.0243	0.0510	0.0243	0.45799	0.0036	0.0259	371
BAT	0.0271	0.0525	0.0241	0.43471	0.0059	0.0266	439
BWOA	0.0271	0.0522	0.0235	0.34280	0.0086	0.0245	332

**Figure 6.** Convergence plot of GA, PSO, BAT, BWOA.

From the test results, GA has a fitness value of 0.025 and computation time of 339 s, PSO has a fitness value of 0.0243 and computation time of 371 s, BAT algorithm has a fitness value of 0.0241 and computation time of 439 s, but BWOA has best less fitness value of 0.0235 and less computation time of 332 s when compared with algorithms mentioned above. From the convergence plot, it is clear that GA takes 40 generations to reach the global minimum value of 0.025, PSO takes 35 generations to reach a global minimum value of 0.0243, BAT algorithm takes 25 generations to reach a global minimum value of 0.0241, but BWOA takes only 17 generations to reach the global minimum value of 0.0235. Thus black widow optimization algorithm has outperformed the other considered GA, PSO, and BAT algorithms and exhibited better performance.

#### 4. Simulation Results and Discussions

In this section, the simulation result of GA, PSO, BAT, and BWOA algorithm optimized PI controlled DC–DC buck converter fed PMDC motor based WT emulator is tested with different operating conditions such as WT emulator power-speed characteristics for static wind profile settings, WT emulator power-speed characteristics for static pitch angle settings, step-change in wind profile settings and step-change pitch angle settings.

Wind speed profile set at 12 m/s, 10.8 m/s, 9.6 m/s, 8.4 m/s, 7.2 m/s, and 6 m/s in wind reference models with a pitch angle maintained at 0°. The corresponding WT emulator power is measured for different speeds of the WT emulator. Figure 7 shows the WT emulator power response with respect speed of the emulator at different wind speed profile settings with GA-PI, PSO-PI, BAT-PI, and BWOA-PI controlled WT emulator.

The emulation efficiency of the WT emulator is related to the power of the wind turbine emulator ( $P_{WTE}$ ) and power of the wind turbine reference model ( $P_{WTR}$ ), and it is expressed in Equation (15):

$$\eta_{emul} = \frac{100 \times P_{WTE}}{P_{WTR}} \quad (15)$$

Table 3 shows the emulation efficiency of the WT emulator for different wind speed profiles. The WT emulator's emulation efficiency range with the GA-PI controller is from 81% to 95.42%, from 88.31% to 97.34% with PSO-PI controller, from 94.85% to 98.21% with BAT-PI controller, and for

BWOA-PI controller, it is from 97.63% to 99.32%. It is evident that BWOA-PI controlled WT emulator has provided maximum efficiency than GA, PSO, and BAT PI-controlled WT emulator.

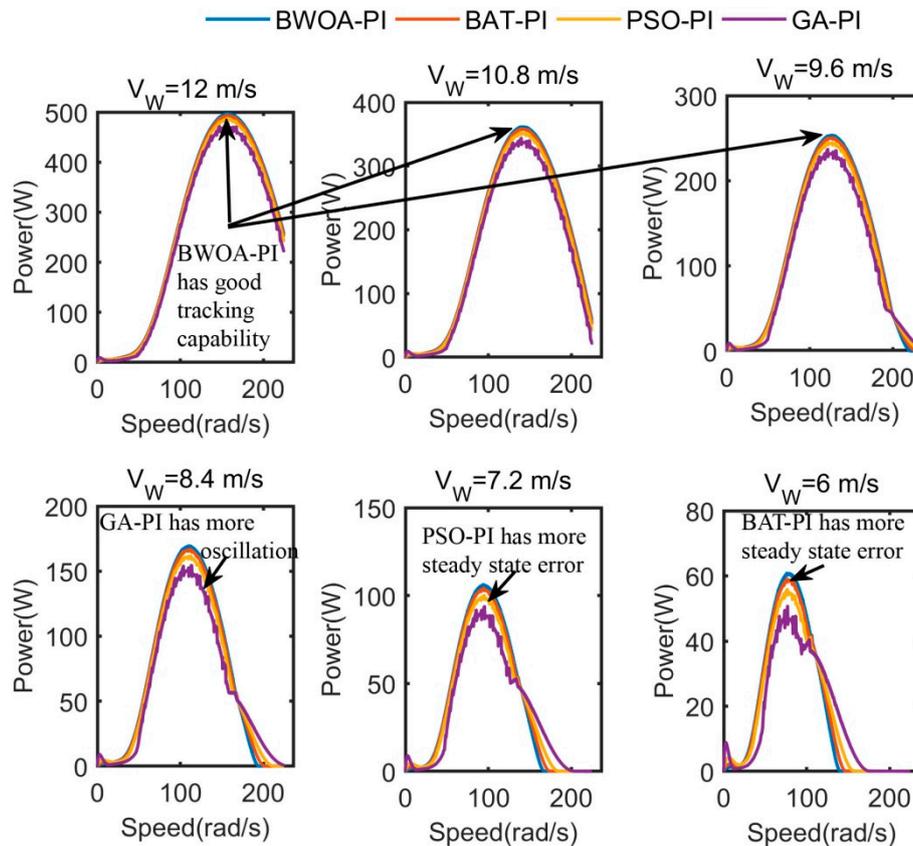


Figure 7. WT emulator power versus speed at different wind profile settings.

Table 3. Emulation efficiency of WT emulator with different wind speed profile.

Algorithm	Wind Speed Profile					
	12 m/s	10.8 m/s	9.6 m/s	8.4 m/s	7.2 m/s	6 m/s
GA-PI	95.42	94.24	92.5	89.07	86.21	81.15
PSO-PI	97.34	96.24	95.25	93.37	92.15	88.31
BAT-PI	98.21	98.16	97.50	96.40	96.20	94.85
BWOA-PI	99.32	99.07	98.95	98.37	98.06	97.63

Pitch angle is set at 0°, 4°, 8°, 12°, 16° and 20° in wind reference model (wind speed fixed at 12 m/s), and corresponding WT emulator power is measured for different speeds of the WT emulator. Figure 8 shows the WT emulator power-speed characteristics for different pitch angle settings with GA-PI, PSO-PI, BAT-PI, and BWOA-PI controlled WT emulator. The emulation efficiency of the WT emulator is calculated using Equation (15), and the corresponding details are provided in Table 4.

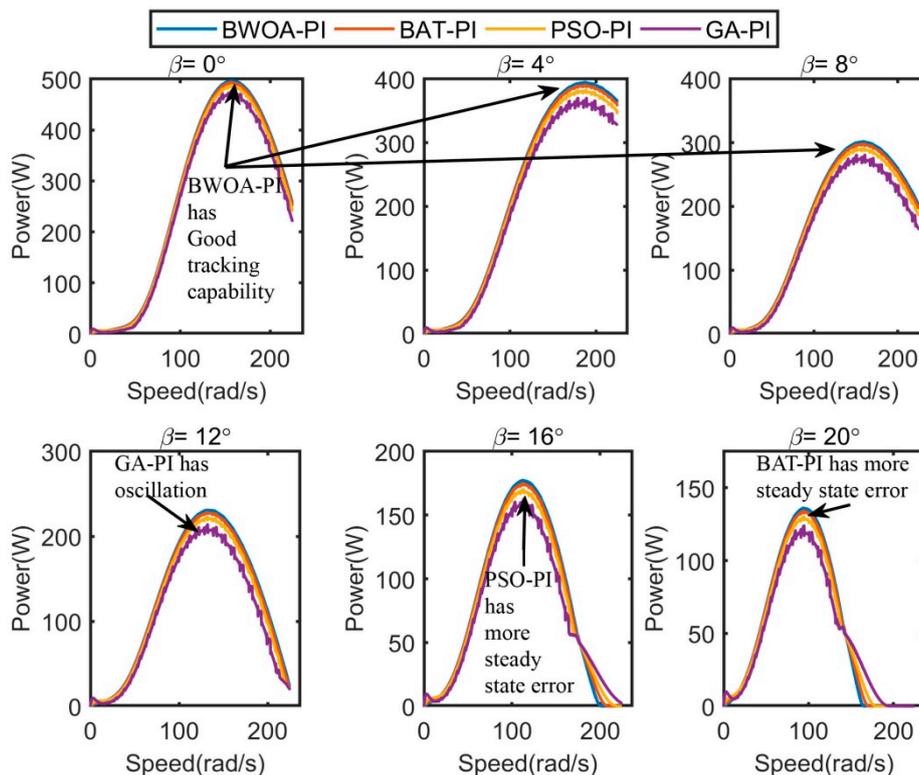


Figure 8. WT emulator power–speed characteristics for different pitch angle settings.

Table 4. Emulation efficiency of WT emulator with different pitch angle.

Algorithm	Pitch Angle					
	0°	4°	8°	12°	16°	20°
GA-PI	95.40	92.62	92.07	91.07	90.06	89.20
PSO-PI	97.08	95.66	95.25	94.83	94.00	93.70
BAT-PI	98.44	97.92	97.67	97.05	96.94	96.74
BWOA-PI	99.36	99.05	99.05	98.50	98.22	98.48

The emulation efficiency range of the WT emulator with the GA-PI controller is from 89.2% to 95.4%, from 93.7% to 97.08% with the PSO-PI controller, from 96.74% to 98.44% with BAT-PI controller, and for BWOA-PI controller, it is from 98.48% to 99.36%. It is ascertained that BWOA-PI controlled WT emulator has provided maximum efficiency than GA, PSO, and BAT PI-controlled WT emulator.

WT emulator is tested for a step-change in wind speed profile, i.e., wind speed is varied from 12 m/s to 10.8 m/s, 10.8 m/s to 9.6 m/s for every 2 s, and corresponding results are measured. The WT emulator’s current and power response with GA, PSO, BAT, and BWOA optimized PI controller is shown in Figure 9, and the corresponding performance results are presented in Table 5. From this result, the BWOA PI-controlled WT emulator has a less steady-state error, quick settling time, and less rise time. These parameters are not in favor of the other considered algorithms such as GA, PSO, and BAT algorithm optimized PI controlled WT emulator. WT emulator is tested for a step-change in pitch angle, i.e., the pitch angle is varied from 0° to 4°, 4° to 8° for every 2 s, and corresponding results are measured. The WT emulator’s current and power response with GA, PSO, BAT, and BWOA optimized PI controller is shown in Figure 10, and corresponding performance results are presented in Table 6.

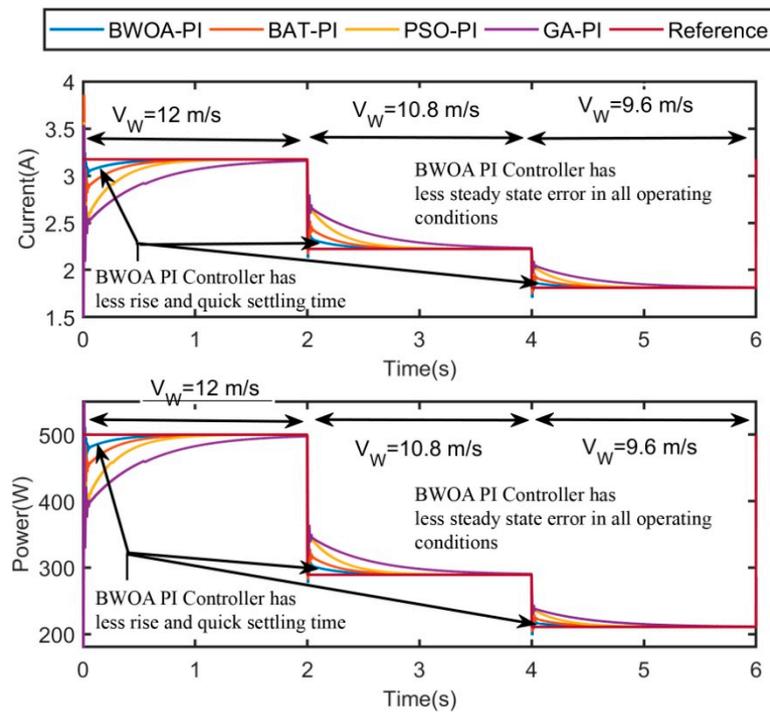


Figure 9. Current and power response of WT emulator for step change in wind speed.

Table 5. Performance analysis of developed emulator for step change in wind velocity.

Time (s)	0–2			2–4			4–6		
Algorithm	tr (s)	ts (s)	Ess (A)	tr (s)	ts (s)	Ess (A)	tr (s)	ts (s)	Ess (A)
GA-PI	0.663	1.383	0.0230	2.677	3.185	0.0274	4.294	5.112	0.0264
PSO-PI	0.317	0.520	0.0258	2.215	2.467	0.0224	4.214	4.396	0.0244
BAT-PI	0.204	0.343	0.0121	2.111	2.386	0.0181	4.101	4.324	0.0151
BWOA-PI	0.058	0.178	0.005	2.054	2.273	0.0065	4.056	4.258	0.0058

tr—rise time, ts—settling time and Ess—steady state error.

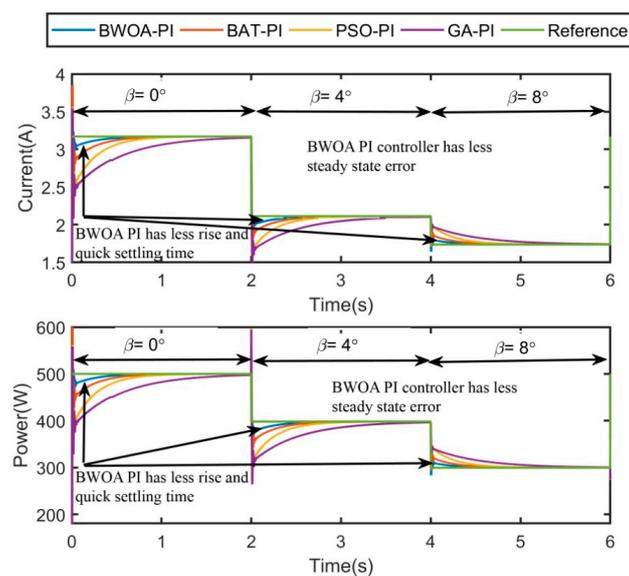


Figure 10. Current and power response of WT emulator for step change in pitch angle.

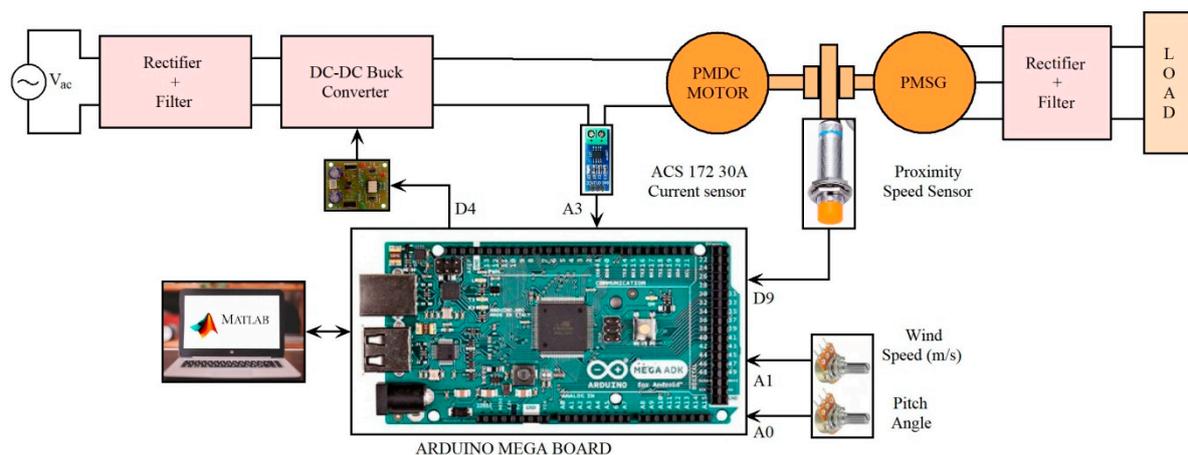
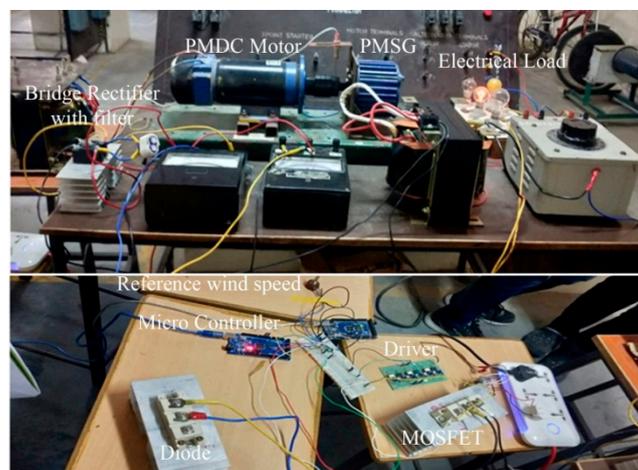
**Table 6.** Performance results for wind turbine emulator for a step change in pitch angle.

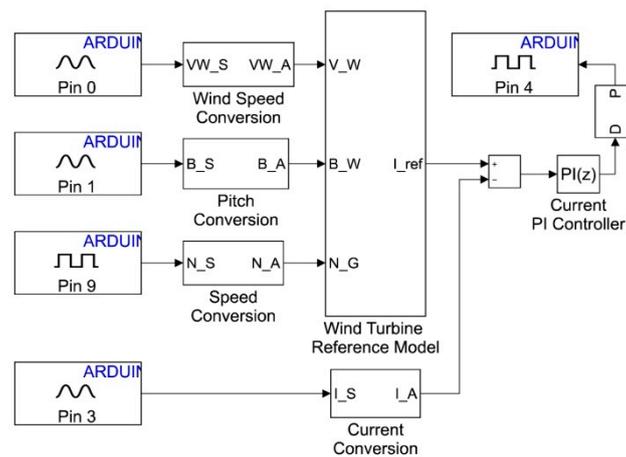
Time (s)	0–2			2–4			4–6		
Algorithm	tr (s)	ts (s)	Ess (A)	tr (s)	ts (s)	Ess (A)	tr (s)	ts (s)	Ess (A)
GA-PI	0.695	1.54	0.0240	2.677	3.415	0.0290	5.454	5.112	0.0275
PSO-PI	0.262	1.021	0.0158	2.215	3.107	0.0178	5.102	4.396	0.0185
BAT-PI	0.165	0.854	0.0142	2.111	2.564	0.0162	4.524	4.324	0.0151
BWOA-PI	0.067	0.231	0.0061	2.054	2.247	0.0084	4.187	4.258	0.0072

From these results, it is observed that the BWOA PI-controlled WT emulator has a less steady-state error, quick settling time, and less rise time. However, these parameters are not in favor of the other considered algorithms such as GA, PSO, and BAT algorithm optimized PI controlled WT emulator. A black widow optimization algorithm optimized PI controlled WT emulator has shown better performance in all operating conditions from these results analyses. In the next section, BWOA optimized PI controlled WT emulator is tested experimentally.

### 5. Experimental Verification of BWOA Optimized PI Controlled WT Emulator

In this section, experimental verification of the BWOA algorithm optimized PI controlled WT emulator is presented. Figure 11 shows the hardware block diagram of the proposed WT emulator. Figure 12 shows the snapshot of the hardware setup of the proposed WT emulator. Figure 13 shows the control logic implemented in the ARDUNIO mega board using the MATLAB/Simulink environment.

**Figure 11.** Hardware block diagram of the BWOA optimized PI controlled WT emulator.**Figure 12.** Snapshot of experimental setup of the BWOA optimized PI controlled WT emulator.

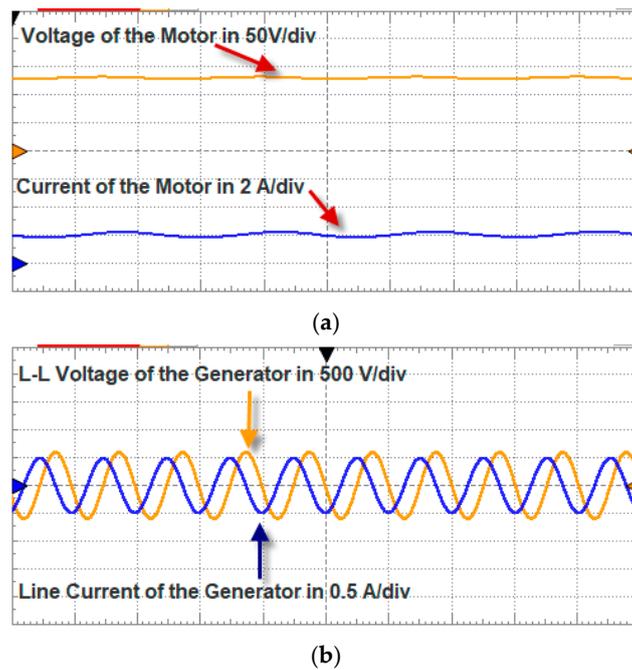


**Figure 13.** BWOA optimized PI controlled WT emulator control logic implementation using MATLAB.

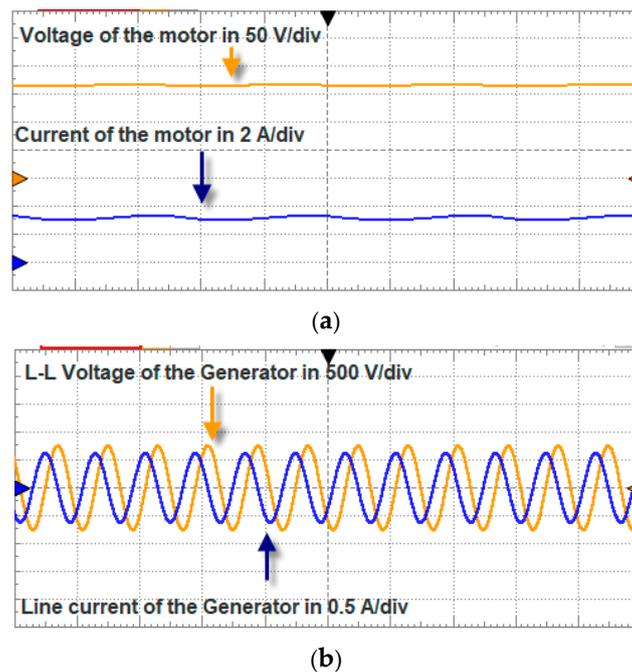
A total of two potentiometers are used for providing wind speed and pitch angle command for the WT reference model. The proximity sensor used for sensing the PMDC motor's speed is the third input for the WT reference model. ACS 172-30A is used for measuring the current of the PMDC motor. The wind speed command, pitch angle command, and current sensor data being received in the analog channel ( $A_0$ ,  $A_1$ , and  $A_3$  pins) of the ARDUINO mega board. The proximity sensor data are received in the ARDUINO mega board in the digital pin ( $D_9$  pin). The wind speed command, pitch angle command, current sensor data, and speed sensor data are converted into a suitable form using conversion blocks. After conversion, these data are processed via the WT reference model and PI controller to generate the corresponding duty cycle. This duty cycle is processed via the PWM generating unit to generate the DC–DC buck converter's MOSFET pulse. The generated pulse is taken out from the digital pin ( $D_4$  pin) of the ARDUINO mega board and given through the driver board to control the buck converter for emulating the characteristics of the WT reference model.

Three operating conditions are considered for testing the effectiveness of the BWOA optimized PI controlled WT emulator in real-time. In case 1, Wind speed is set at 9.6 m/s. Pitch angle is set at  $0^\circ$  using potentiometer of the hardware setup and corresponding results such as the voltage of the motor, current of the motor, the line to line voltage of the generator and line current of the generator are measured and shown in Figure 14. The motor's voltage is around 135 V, the current of the motor is around 2 A, and the motor's power is 270 W. The emulation efficiency obtained from the hardware is around 97.5%, and for simulation, it is around 98.95%. In case 2, Wind speed is set at 12 m/s. Pitch angle is set at  $0^\circ$  using potentiometer of the hardware setup and corresponding results such as the voltage of the motor, current of the motor, the line to line voltage of the generator and line current of the generator are measured and shown in Figure 15. The motor's voltage is around 163 V, the current of the motor is around 3 A, and the power consumed by the motor is 480 W. The emulation efficiency obtained from hardware is around 97.8%, and it is 99.36% for the simulation analysis.

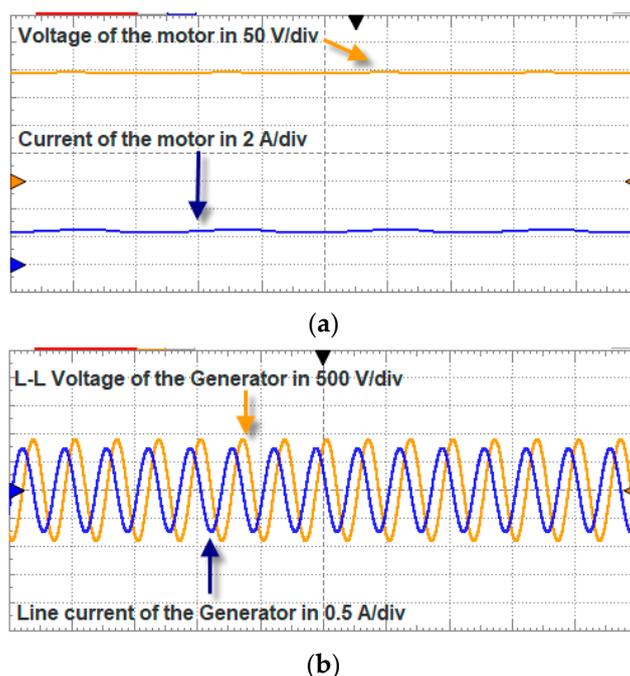
In case 3, Wind speed is set at 12 m/s, and the pitch angle is set at  $4^\circ$  using potentiometer of the hardware setup. The corresponding results, such as voltage and current of the motor, line-to-line voltage of the generator, and line current of the generator, are measured and shown in Figure 16. The motor's voltage is around 204 V, the current of the motor is around 2.4 A, and the motor's power is 489.6 W. The emulation efficiency obtained from the hardware is around 97.92%, and for simulation, it is around 99.05%. Comparisons of the emulation efficiency from simulation and experimentation for the three cases are shown in Table 7. From the results, it is observed that experimentation results deviate minimally from the simulation results.



**Figure 14.** Experimental results of case 1 conditions: (a) voltage and current of the motor, and (b) L-L voltage and line current of the generator.



**Figure 15.** Experimental results of case 2 conditions: (a) voltage and current of the motor, and (b) L-L voltage and line current of the generator.



**Figure 16.** Experimental results of case3 conditions: (a) voltage and current of the motor, and (b) L-L voltage and line current of the generator.

**Table 7.** Comparisons of emulation efficiency in simulation and experimentation.

Conditions	Emulation Efficiency in Simulation (%)	Emulation Efficiency in Hardware (%)	Deviation of the Efficiency (%)
Case 1	98.95	97.50	1.45
Case 2	99.36	97.80	1.56
Case 3	99.05	97.92	1.13

## 6. Conclusions

This paper presents the black widow optimization algorithm based on an optimized PI controlled WT emulator. The overall Simulink model of the proposed system was designed and tested in a MATLAB/Simulink environment. BWOA optimized PI controlled WT emulator performances were compared with GA, PSO, and BAT algorithm optimized PI controlled WT emulator. The BWOA optimized PI controlled WT emulator's emulation efficiency range was from 97.63 to 99.36% in simulation. The tracking of the proposed BWOA optimized PI controller's reference command had comparatively less rise time, quick settling time, and less steady-state error than BAT, GA, and PSO algorithm optimized PI controllers. The experimental verification of the proposed BWOA optimized PI controlled WT emulator was carried out in real-time to check the proposed controller's suitability. The emulation efficiency range obtained from experimentation was from 97.5 to 97.92%. Therefore, the proposed black widow optimization algorithm optimized PI controller may be considered suitable for the WT emulator system.

**Author Contributions:** K.P.: conceptualization, methodology, formal analysis, writing—reviewing and editing. M.V.: software, investigation, analysis, writing—reviewing and editing. T.S.B.: analysis, writing—reviewing and editing, project administration. B.V.M.: supervision, writing—reviewing and editing. T.T.: supervision, writing—reviewing and editing. A.N.A.: supervision, writing—reviewing and editing. M.R.I.: reviewing and editing, project administration, facilities. A.Z.K.: reviewing and editing, project administration. M.A.P.M.: reviewing and editing, project administration. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Karthik, B.; Thanikanti, S.B.; Subramaniam, U.; Sudhakar, N.; Sichilalu, S. A novel review on Optimization Techniques used in Wind Farm Modelling. *Renew. Energy Focus* **2020**, *35*, 84–96.
2. Campos, R.A.; Nascimento, L.R.d.; Rütther, R. The complementary nature between wind and photovoltaic generation in Brazil and the role of energy storage in utility-scale hybrid power plants. *Energy Convers. Manag.* **2020**, *221*, 113160. [[CrossRef](#)]
3. Kathiresan, C.A.; PandiaRajan, J.; Sivaprakash, A.; Babu, T.S.; Islam, M. An Adaptive Feed-Forward Phase Locked Loop for Grid Synchronization of Renewable Energy Systems under Wide Frequency Deviations. *Sustainability* **2020**, *12*, 7048. [[CrossRef](#)]
4. Wollz, D.H.; da Silva, S.A.O.; Sampaio, L.P. Real-time monitoring of an electronic wind turbine emulator based on the dynamic PMSG model using a graphical interface. *Renew. Energy* **2020**, *155*, 296–308. [[CrossRef](#)]
5. Azharuddin, M.; Babu, T.S.; Bilakanti, N.; Rajasekar, N. A nearly accurate solar photovoltaic emulator using a dSPACE controller for real-time control. *Electr. Power Compon. Syst.* **2016**, *44*, 774–782. [[CrossRef](#)]
6. Sudhakar Babu, T.; Mohammed Azharuddin, S.; Nishant, B.; Rajasekar, N. A dynamic photo voltaic emulator using dSPACE controller with high accuracy solar photo voltaic characteristics. *J. Renew. Sustain. Energy* **2016**, *8*, 15503. [[CrossRef](#)]
7. Zhang, J.; Zhang, M.; Li, Y.; Qin, J.; Wei, K.; Song, L. Analysis of wind characteristics and wind energy potential in complex mountainous region in southwest China. *J. Clean. Prod.* **2020**, *274*, 123036. [[CrossRef](#)]
8. Battaiotto, P.E.; Mantz, R.J.; Puleston, P.F. A wind turbine emulator based on a dual DSP processor system. *Control Eng. Pract.* **1996**, *4*, 1261–1266. [[CrossRef](#)]
9. Chinchilla, M.; Arnaltes, S.; Rodriguez-Amenedo, J.L. Laboratory setup for wind turbine emulation. In Proceedings of the IEEEICIT'04, Hammamet, Tunisia, 8–10 December 2004; pp. 553–557.
10. Arifujjaman, M.; Iqbal, M.T.; Quaicoe, J.E. An Isolated Small wind turbine Emulator. In Proceedings of the Canadian Conference on Electrical and Computer Engineering, Ottawa, ON, Canada, 7–10 May 2006; pp. 1854–1857.
11. Li, W.; Xu, D.; Zhang, W.; Ma, H. Research on wind turbine Emulation based on DC Motor. In Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications, Harbin, China, 23–25 May 2007; pp. 2589–2593.
12. Patel, R.; Patki, C.V.; Agarwal, V. Armature and Field Controlled DC Motor Based Wind Turbine Emulation for Wind Energy Conversion Systems Operating over a Wide Range of Wind Velocity. In *Sustainability in Energy and Buildings*; Howlett, R.J., Jain, L.C., Lee, S.H., Eds.; Springer: Berlin/Heidelberg, Germany, 2009. [[CrossRef](#)]
13. Mahdy, A.; El-Hakim, S.M.; Hanafy, H.H. Small wind turbine emulator with separately excited DC motor using analog electronic circuit. In Proceedings of the IET Conference RPG 2011, Edinburgh, UK, 6–8 September 2011; pp. 1–6.
14. Satpathy, A.S.; Sahoo, N.C. Development of control scheme for a stand-alone wind system: Wind turbine emulated using DC motor. In Proceedings of the IEEE Students' Conference on Electrical, Electronics and Computer Science, Bhopal, India, 1–2 March 2012; pp. 1–4.
15. Kouadria, S.; Belfedhal, S.; Berkouk, E.M.; Meslem, Y. Development of real time wind turbine emulator based on DC motor controlled by PI regulator. In Proceedings of the Eighth International Conference and Exhibition on Ecological Vehicles and Renewable Energies, Monte Carlo, Monaco, 27–30 March 2013; pp. 1–5.
16. Martinez, F.; Herrero, L.C.; de Pablo, S. Open loop wind turbine emulator. *Renew. Energy* **2014**, *63*, 212–221. [[CrossRef](#)]
17. Yadav, A.K.; Singh, M.; Meena, D.C. Modelling and simulation of wind turbine emulator using DC motor. In Proceedings of the IEEE 7th Power India International Conference, Bikaner, India, 25–27 November 2016; pp. 1–5.
18. Bailapudi, M.P.K.; Sinha, N. Fuzzy logic controlled wind turbine emulator (WTE). In Proceedings of the International Conference on Information Communication and Embedded Systems, Chennai, India, 25–26 February 2016; pp. 1–8.
19. Benaouinate, L.; Khafallah, M.; Mesbahi, A.; Martinez, A. Development of a useful wind turbine emulator based on permanent magnet DC motor. In Proceedings of the 14th International Multi-Conference on Systems, Signals & Devices, Marrakech, Morocco, 28–31 March 2017; pp. 44–48.

20. Sirouni, Y.; Hani, S.E.; Naseri, N.; Aghmadi, A.; Harouri, K.E. Design and Control of a Small Scale wind turbine emulator with a DC Motor. In Proceedings of the 6th International Renewable and Sustainable Energy Conference, Rabat, Morocco, 5–8 December 2018; pp. 1–6.
21. Moussa, I.; Bouallegue, A.; Khedher, A. A New wind turbine emulator based on DC machine: Hardware Implementation using FPGA board for an Open Loop Operation. *IET Circ. Device Syst.* **2019**, *13*, 896–902. [[CrossRef](#)]
22. Satish Kumar, P.; Chandrasenan, R.P.S.; Victor Sam Moses Babu, K. Design and Implementation of wind turbine emulator using FPGA for Stand Alone Applications. *Int. J. Ambient Energy* **2020**. [[CrossRef](#)]
23. Kan, C.; Devrim, Y.; Eryilmaz, S. On the theoretical distribution of the wind farm power when there is a correlation between wind speed and wind turbine availability. *Reliab. Eng. Syst. Saf.* **2020**, *203*, 107115. [[CrossRef](#)]
24. Hayyolalam, V.; Kazem, A.A.P. Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Eng. Appl. Artif. Intel.* **2020**, *87*, 103249. [[CrossRef](#)]

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).