Optimal tuning of PI Controllers for Doubly-Fed Induction Generator using Grey Wolf Optimizer

Al-Said Abd ElAziz Osman[1], Amged S. El-Wakeel[2], A.kamel[3] and Hatem M. Seoudy[3]

[1] Electrical engineering Department, Faculty of Engineering - AlAzhar University, Cairo, Egypt
[2] Electrical power and energy Department, Military Technical College, Cairo, Egypt
[3] Electrical power and Machine Department High Institute of Engineering, El Shorouk Academy, Cairo, Egypt
hatemseoudy@gmail.com

Abstract: An intelligent control of Doubly Fed Induction Generator (DFIG) system using Proportional-Integral (PI)controller tuned by optimization techniques is proposed in this paper.System identification technique was presented in this work to estimate the transfer function of the reactive power loop and speed loop of the proposed system.An implemented laboratory prototype consists of 0.37kW, 220 V, 50Hz Brushless DC Motor (BLDC) and its drive circuit controlled by voltage source inverter for various wind speed.A 0.27 kW wound rotor induction machine, working as the DFIG, coupled with turbine machine by a coupler and driven through a back-to-back converter. This system can be applied as a stand-alone power supply system or as the emergency power system when the electricity grid fails. The rotor side converter is controlled using the field-oriented control to control the reactive power at different rotor speeds.Grey Wolf Optimizer (GWO) proposed in this study to tune the (PI) controller. Moreover, Particle Swarm Optimization (PSO) is also used to tune the PI controller for comparison. For studying the performance of each algorithm, different case studies are performed, such as step changes in the rotating speed andelectrical load. Experimentalresults showed that the proposed techniqueis adequate and sufficient to be used with off-grid stand-alone DFIG systems. It alsoshowed the improved performance of GWO over the PSOin tuning the PI controller.

[Al-Said Abd ElAziz Osman, Amged S. El-Wakeel, A.kamel and Hatem M. Seoudy. **Optimal tuning of PI Controllers for Doubly-Fed Induction Generator using Grey Wolf Optimizer.** *J Am Sci* 2015;11(11):154-164]. (ISSN: 1545-1003). http://www.jofamericanscience.org. 16. doi:10.7537/marsjas111115.16.

Keywords: Brushless DC motor (BLDC), Doubly Fed Induction Generator (DFIG),Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO),PI Controller, Wind Turbine Emulator and Optimal control.

1. Introduction

The DFIG systems have been widely used on large variable-speed fixed-frequency wind power generation systems, hydropower systems and turbine engine power generation systems. There are two types of applications of a DFIG system, grid connected type and stand-alone type.

The grid-connected DFIG system is developed and is widely used today and many control schemes have been reported. However, only little attention has been paid toward the development of the control schemes of the stand-alone type DFIG system[1, 2]. On the other hand, satisfied transient performance can't be achieved using the traditional PI control for the DFIG system due to time-varying and nonlinear controlcharacteristics of the DFIG system. Therefore, this study addresses the issue to develop an intelligent control scheme for the DFIG system for stand-alone application.

Numerous control methods such as adaptive control, neural control, and fuzzy control have been applied. Among them, the best known is the PI controller. Conventional PI controller is widely used in wind energy conversion systems due to its simplicity in design and implementation. However it is difficult to achieve the desired control performance without controller tuning.A parallel PI controller is shown in Figure 1. The controller parameters or gains (Kp,and Ki)are chosen to meet prescribed performance criteria, classically specified in terms of rise and settling times, overshoot, and steady state error, following a step change in the demand[3].

Over the years, numerous heuristic methods have been proposed for tuning PI controllers. The first method used the classical tuning rules proposed by Ziegler and Nichols[4]. In general, it is often hard to determine optimal or near optimal PI parameters with the Ziegler-Nichols formula in many plants. Later on, a lot of researches were devoted to the intelligent PI controllers such as the fuzzy, and variable gain algorithms[5, 6], but the fuzzy and variable gain rules still need to be optimized. Thus, the biological optimization algorithms such as the evolutionary computing, swarm intelligence, and so on, were introduced to improve the optimization of PI parameters[3, 7, 8]. Swarm intelligence has purposed Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO) that have opened paths to a new generation of advanced process control. These advanced techniques to design control systems are, in general, dependent on achieving optimum performance with various types of disturbance that are unknown in most practical applications[8-10].

In this paper, the system modeling is described in section2. A comparison between the simulation results obtained by the proposed GWO method, and the other stochastic techniques such as PSO, is presented in section 3.

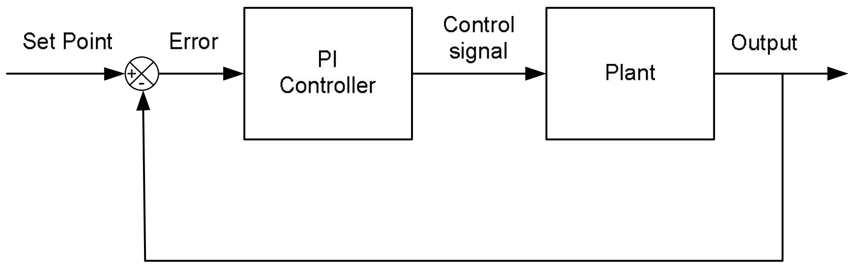


Figure 1. PI controller

2. System modeling

2.1. PI controller and fitness function modeling

The tuning of PI parameters is related to the characters of the system. Thus, the properly tuned PI parameters are needed to approach the required performance. The transfer function of a PI controller is usually given by Alrashidi at[11]:

$$G_c(s) = K_p + \frac{K_i}{S} \quad (1)$$

where, K_p,and$K_i$denotes the proportional gain, and integral gain respectively.

In thispaper, the strategies of PSO, and GWO are implemented for the optimum search of the controller parameters for reactive power control of DFIG and speed control of BLDC motor (as prime mover) according to the criteria of performance index. These criteria include WGAM1 (Weighted Goal Attainment Method 1), and WGAM2 (Weighted Goal Attainment Method 2) described by equations 2, and 3 respectively[4, 12].

$$WGAM1 = \frac{1}{[c_1(t_r - t_{rd})^2 + c_2(M_p - M_{pd})^2 + c_3(t_s - t_{sd})^2 + c_4(e_{ss} - e_{ssd})^2]} \quad (2)$$

$$WGAM\ 2 = \frac{1}{(1 - e^{-\beta}) \cdot (M_p + e_{ss}) + (e^{-\beta}) \cdot (t_s - t_r)} \quad (3)$$

Where, *r(t)* is the desired output, *y(t)* is the plant output, *e(t)* is the error signal, β weighting factor, c_1 : c_4 are positive constants (weighting factors), their values are chosen according to prioritizing their importance, t_{rd} is the desired rise time, M_{pd} is the desired maximum overshoot, t_{sd} is the desired settling time, and e_{ssd} is the desired steady state error.

In WGAM1, the actual closed-loop specification of the system with controller, t_r, M_p, t_s, *and* $e_{ss}$are used to evaluate the fitness function. This is done by summing the squares of the errors between actualanddesired specifications, t_{rd}, M_{pd}, t_{sd}, ande_{ssd}, as given in equation3.

The WGAM2 can satisfy the designer requirements using the weighting factor β value as given in equation 7. The factor β is set larger than 0.7 to reduce the overshoot and steady state error. On the other hand β is set smaller than 0.7 to reduce the rise time and settling time[3].

2.2. Laboratory Wind Turbine Test Bed Modeling

The system identification toolbox in MATLAB used to find the transfer function of both the DFIG with its converter for reactive power control and the BLDC motor and its drive circuit for speed control[4, 10].

The transfer function models for reactive power control of DFIG are estimated for orders of rang from 2 to 4 as poles and 1 to 3 as zeroes using step input voltage data (0-0.6 V) and its related output as a test data. Figure2shows the measured output and the percentage fit of each simulated model to the measured output. It is clear from Figure2that the best fitting for the validation data set is obtained for a transfer function model of 4 poles, with 2 and 3 zeroes which will lead to choose 4 poles and 2 zeroes as less order with the same fitting (91.5%). The transfer function of the reactive power loop of DFIG is:

$$G_Q(S) = \frac{8134S^2+2.101e04S+411.6}{S^4+26.4S^3+215.5S^2+508.4S+7.229} \tag{4}$$

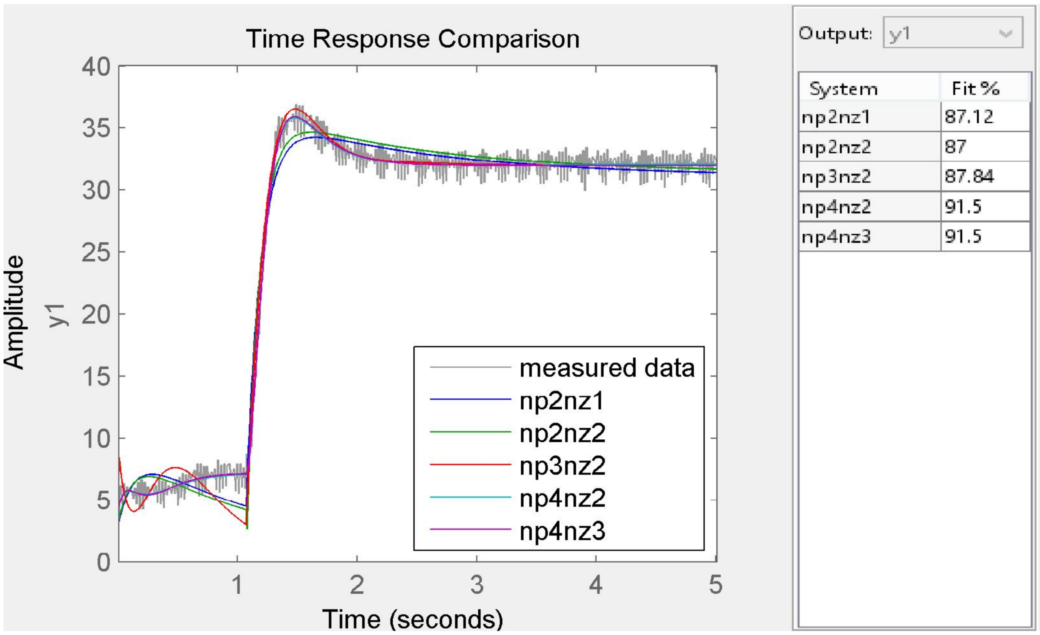


Figure2.Measured and simulated transfer function model output of reactive power loop

In the meantime, the transfer function models for speed control of BLDC motor are estimated for orders of rang from 1 to 3 as poles and 1 to 3 as zeroes using descending step input voltage data (2.5-1 V) and its related output as a test data. Figure3shows the measured output and the percentage fit of each simulated model to the measured output.Figure3 has clarify the best fitting for the validation data set which is obtained for a transfer function model of 3 poles 2 zeroes with fitting (95.13%). The transfer function of the Speed loop of BLDC and its drive circuit is:

$$G_{Speed}(S) = \frac{1279S^2+5.867e05S-1.053e04}{S^3+50.21S^2+915S+2.506e-12} \tag{5}$$

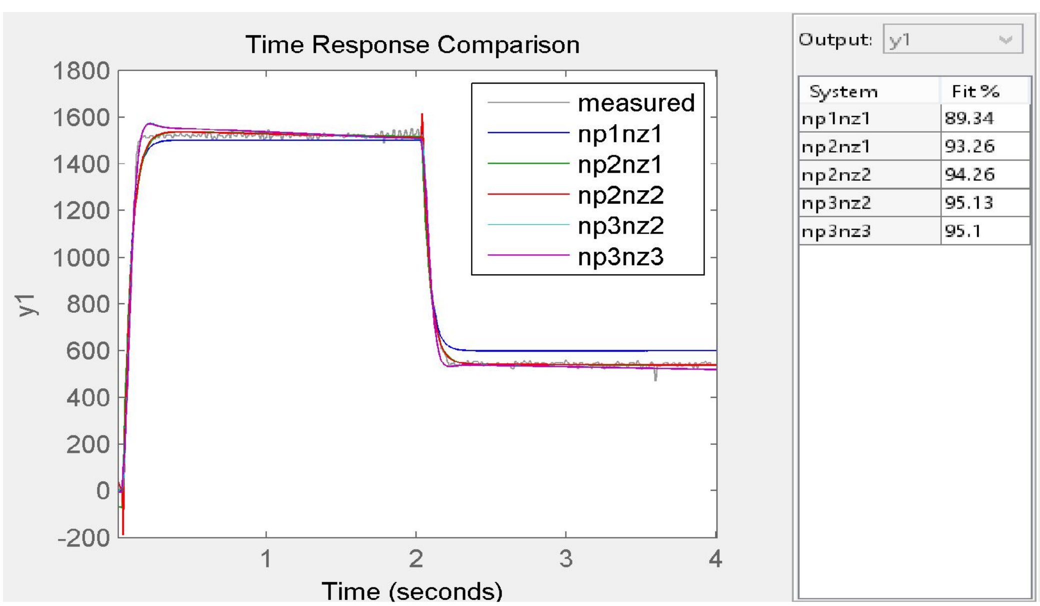


Figure3. Measured and simulated transfer function model output of speed loop

3. Experimental Models Validation

The transfer function of the models is explained in equation (4) and (5); Figure4shows the response of both identified models and practical system for step input.

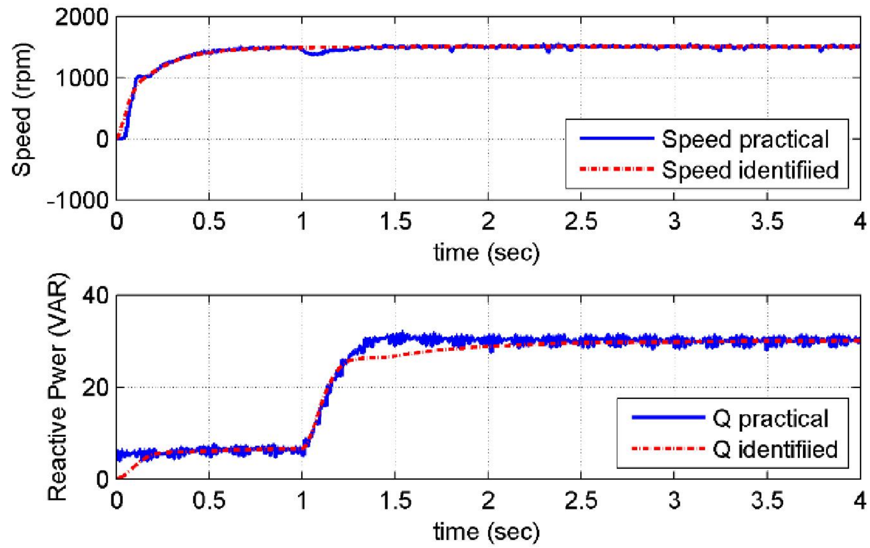


Figure4.Reactive Power and Speed response of practical and identified model

It is clear from Figure4 that the response of the identified model is approximately coincide with the response of the actual system which prove that the model obtained by equation (4) and (5) can be used in the simulation study.

The closed loop system which explains the structure of PI tuning system is shown in FIGURE 5.

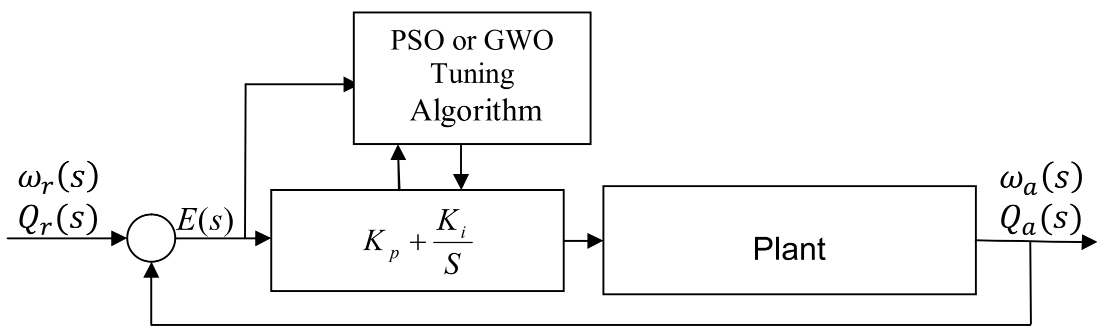


FIGURE **5**.Structure of closed loop system with PID tuning algorithms.

2.3. Modeling of the PSO-PI controller

In this subsection, the PSO-PI controller is proposed. The method of tuning the parameters of PI controller by the Particle Swarm Optimization (PSO) is briefly reviewed. In the PSO algorithm, a population of particles is put into the d-dimensional search space with randomly chosen velocities and positions knowing their best values so far (*pbest*) and the position in the d-dimensional space. The velocity of each particle is adjusted according to its own flying experience and the other particles flying experience as the following[4, 10, 13-15]:

$$\mathbf{v_i^{k+1} = \omega v_i^k + c_1 rand_{1i}(pbest_i - s_i^k) + c_2 rand_{2i}(gpest - s_i^k)} \quad (6)$$

Where, $v_i^k$is the current velocity of particle i at iteration k,v_i^{k+1} is the updated velocity of particle i, ω is the inertia weight. c_1, c_2 are two acceleration positive PSO constants,s_i^k is the current position of particle i at iteration k,$rand_{1i}$, $rand_{2i}$ are random numbers between 0 and 1, $pbest_i$ is the best position of particle i, and gbest is the global best position of the group so far.

The new position s_i^{k+1} can be modified using the present positions_i^k and updated velocityv_i^{k+1}.

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (7)$$

The positive constants c_1 and c_2 are usually set between 0.5 to 2 [8].The inertia weight w is set as a decreasing linear function with the iteration number from 0.9 to 0.4 [8, 10, 16, 17].This large value of inertia weight at the beginning enhances the PSO global searching ability, while, the small inertia weight near the end of the run improves its local search ability.

The fitness function value is calculated for each particle. If the value is better than the current *pbest* of the particle, the *pbest*value is replaced by the current value. If the best value of *pbest* is better than the current *gbest*, the *gbest* is replaced by the best value and the particle number with the best value is stored. The operation is continued until the current iteration number reaches the predetermined maximum iteration number.

The PSO algorithm has been run for twenty independent trials with different settings until the solutions are very close to each other[10]. According to the trials, the PSO parameters are summarizedinTable 1.

Table 1. PSO parameters

Population size	10
Number of generations	10
Acceleration Constant c_1	0.5
Acceleration Constantc_2	1.5
Initial inertia weight w_{max}	0.9
Final inertia weight w_{min}	0.2

2.4. Modeling of the GWO-PI controller

This section reviews the main steps of gray wolf optimizer (GWO) to tune the PI controller. GWO is a new population based algorithm which is introduced by Gaing in[6]. GWO algorithm inspired by grey wolves. The method mimicked the social hierarchy and hunting behavior of grey wolves. For simulating the leadership hierarchy in GWO algorithm, four groups are defined: alpha, beta, delta, and omega.

The three main steps of hunting, searching for prey, encircling prey, and attacking prey, are simulated.

This algorithm requires a number of parameters to be set, which is:

- Initialize alpha, beta, and delta,
- Number of search agents,
- Maximum number of iterations,
- Number of sites selected for neighborhood search (out of n visited sites) and the stopping criterion.

The main steps of grey wolf hunting are as follows:

- Tracking, chasing, and approaching the prey.
- Pursuing, encircling, and harassing the prey until it stops moving.
- Attack towards the prey.

For modeling the social hierarchy of wolves until designing GWO, the fittest solution is considered as the alpha. Accordingly, the second and third best solutions are beta and delta respectively. The rest of the candidate solutions are considered to be omega. The x wolves follow these three wolves.

For modeling encircling behavior, some equations are considered:

$$\vec{D} = \left|\vec{C} \cdot \vec{X_p}(t) - \vec{X}(t)\right| \quad (8)$$

$$\vec{X}(t+1) = \vec{X_p}(t) + \vec{A} \cdot \vec{D} \quad (9)$$

where t indicates the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X_p}$ is the position vector of the prey, and $\vec{X}$ indicates the position vector of a grey wolf. The vectors $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (10)$$

$$\vec{C} = 2 \cdot \vec{r_2} \quad (11)$$

where components of $\vec{a}$ are linearly decreased from 2 to 0 over the course of iterations and r_1, r_2 are random vectors in [0, 1].

The first three best solutions obtained so far and oblige the other search agents (including the omegas) to update their positions according to the position of the best search agents. The following formulas are proposed in this regard.

$$\vec{D_\alpha} = \left|\vec{C_1} \cdot \vec{X_\alpha} - \vec{X}\right|, \vec{D_\beta} = \left|\vec{C_2} \cdot \vec{X_\beta} - \vec{X}\right|, \vec{D_\delta} = \left|\vec{C_2} \cdot \vec{X_\delta} - \vec{X}\right| \quad (12)$$

$$\vec{X_1} = \vec{X_\alpha} + \vec{A_1} \cdot (\vec{D_\alpha}), \vec{X_2} = \vec{X_\beta} + \vec{A_2} \cdot (\vec{D_\beta}), \vec{X_3} = \vec{X_\delta} + \vec{A_3} \cdot (\vec{D_\delta}) \quad (13)$$

$$\vec{X}(t+1) = \vec{X_1} + \vec{X_2} + \vec{X_3} \big/ 3 \quad (14)$$

It can be observed that the final position would be in a random place within a circle which is defined by the positions of alpha, beta, and delta in the search space. In other words alpha, beta, and delta estimate the position of the prey, and other wolves updates their positions randomly around the prey [18-21].

The flow chart of the GWO algorithm is presented in Figure6.

4. Wind Turbine Emulator and DFIG System Test Bed

As part of an implementing of the Wind Power Generator System, the turbine must be emulated by a wind speed dependent torque source. A Brushless DC Motor (BLDC) and its drive circuit is used to emulate the wind turbine. Due to the unconventional torque and power generation from wind, a speed control system has been designed to control the BLDC motor speed.The controller takes the generator speed feedback through an encoder, which is connected to the shaft of the DFIG. The encoder output ranges between 0 and 2 volts for shaft speed ranges between 0 and 2000 r.p.m.The drive circuit of the BLDC motor is supplied from 220 volts AC mains, rectified to a DC output and passed through an IGBT inverter, which drives the BLDC motor. The control input voltage ranges from 0 to 5 volts, which determines the output voltage delivered to the motor.Besides, of wind turbine, a 0.27 kW wound rotor induction machine, working as the DFIG, is mechanically coupled with turbine machine by a coupler and driven through a converter.

A computer (i7-4700MQ CPU, 2.40 GHz, 6 GB RAM), running MATLAB/SIMULINK, is connected to the system setup through a data acquisition card (NI 6009). Data acquisition card has been interfaced to MATLAB/SIMULINK using Data Acquisition Toolbox.This computer is used to design and implement the PI controllers tuned by optimization techniques presented earlier in order to control the system test bench.The control program has been applied practically in real time for PI controllers. It performs the measurement of the output signals from tachometer and the smart power transducer. It also computes the control signal based on the control strategy and apply it in analog form (0 to 5V) to the controller to drive circuit of the BLDC motor and the inverter of the DFIG with voltage (0 to 220 V).

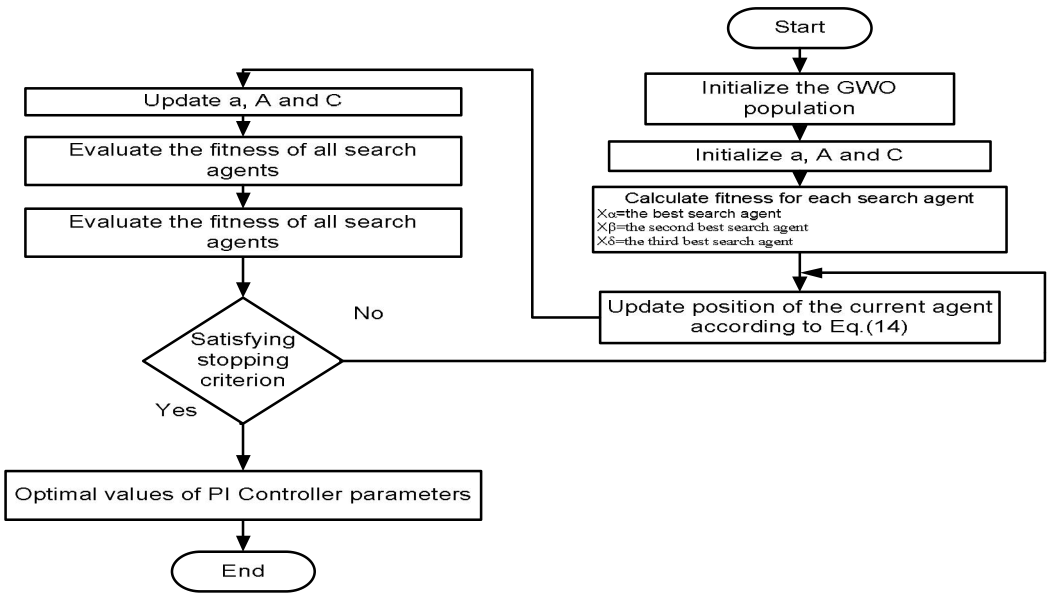


Figure6. Flow chart of the proposed GWO algorithm.

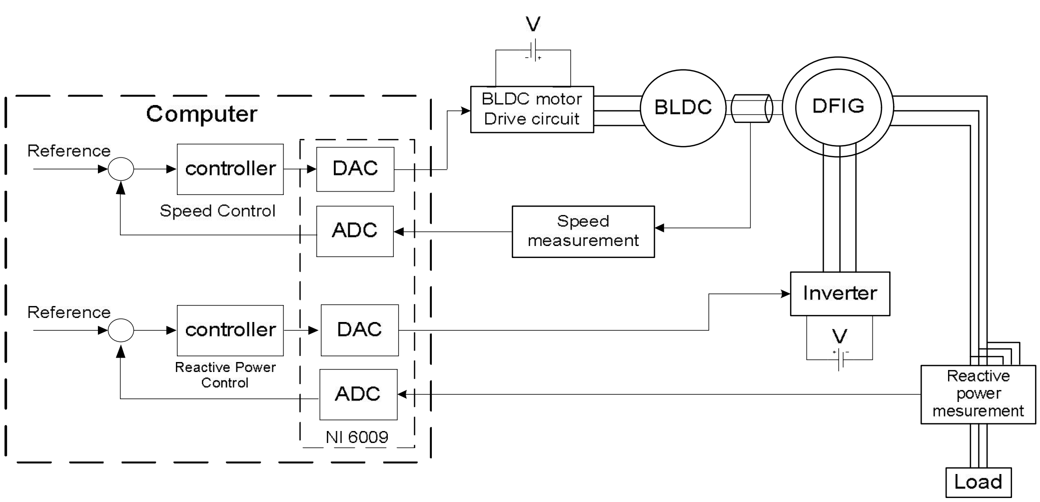


Figure7.Block Diagram of DFIG system test bench

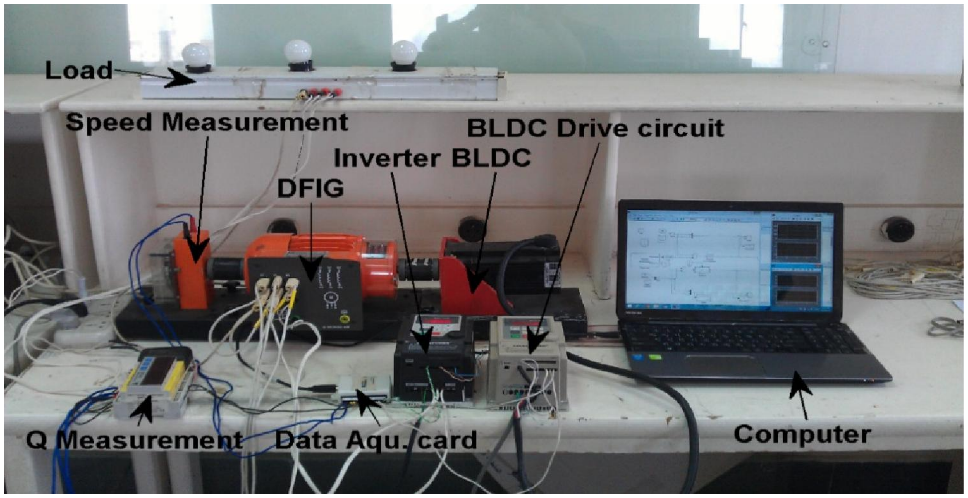


Figure8. General view of the system setup

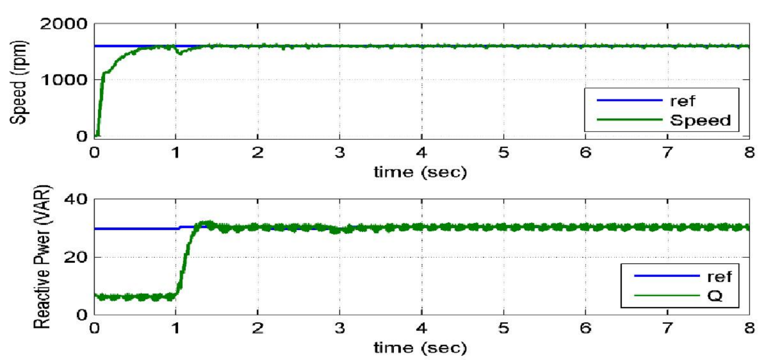
The whole experimental system control scheme is illustrated in Figure8. The block diagram of the whole system is shown in Figure7.

5. Experimental results

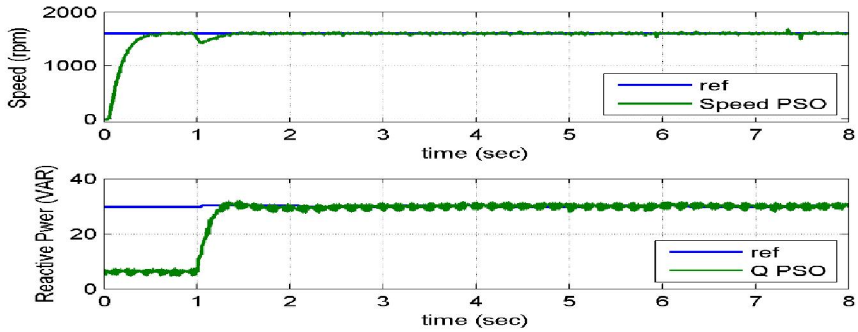
In order to verify the control performance of the proposed techniques controlling the DFIG system, four cases are presented. Step commands of 1600 rpm and 30 VAR are given to show the regulating response with three-phase Resistive-Inductive(RL) load (Case 1) and then change the load to three-phase induction motor (Case 2) to demonstrate the capability for the stand-alone power application. Moreover, the rotor speed is changed to emulate the wind speed variation: The rotor speed is changed from 1600 to 1400 rpm to 1600 rpmwith three-phase RL load (Case 3).Then The rotor speed is changed from 1600 to 1400 rpm to 1600 rpmwith three-phase induction motor (Case 4).

In the experimentation, first, some experimental resultsusing traditional PI control for both the speed regulation and reactive power regulation. Since theDFIG system is a nonlinear time-varying system withcomplex dynamic models. Thus, in this study, the gains of thePI controller for both controllers are obtained by trial anderror in order to achieve good transient and steady state control performance at different operating conditions. The resulted gains are Kp = 0.0007 and Ki = 0.01 for the speed regulation and Kp = 0.02 and Ki = 0.1 for the reactive power regulation. To further emulate the variation of the wind speed, someexperimental results using PI, PSO-PI, and GWO-PI controllers areprovided to verify the control performance of the controlled DFIG system.

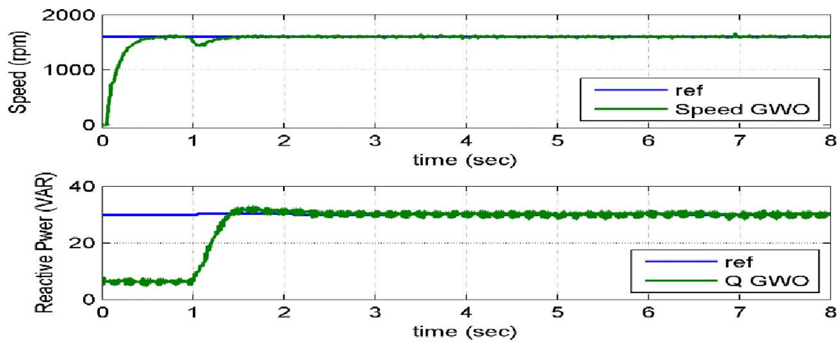
The response of the implemented system under the influence of PI, PSO-PI, and GWO-PI controllers are shown in Figures 11 to 14.



(a)

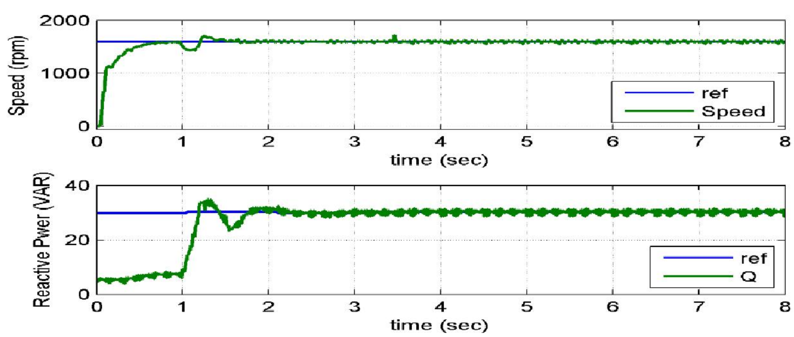


(b)

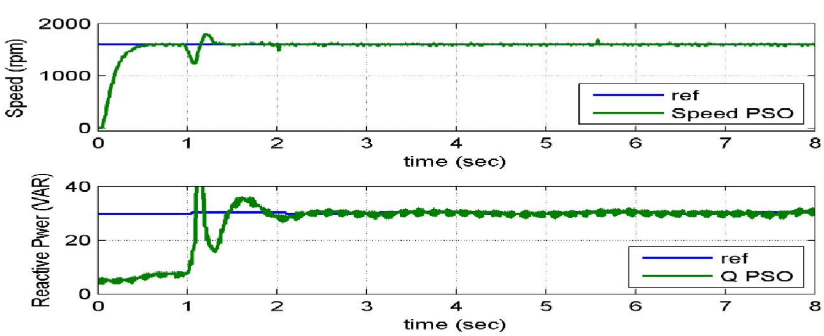


(c)

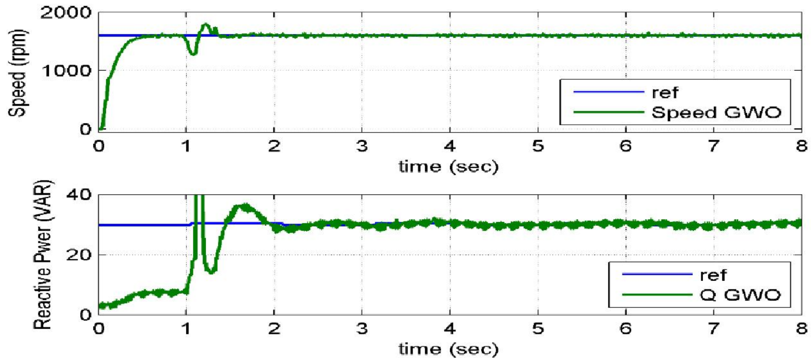
Figure 9. Experimental results of Speed and Reactive Power Case 1 at 1600 rpm. (a) PI controller.(b)PSO-PI controller. (c)GWO-PI controller.



(a)

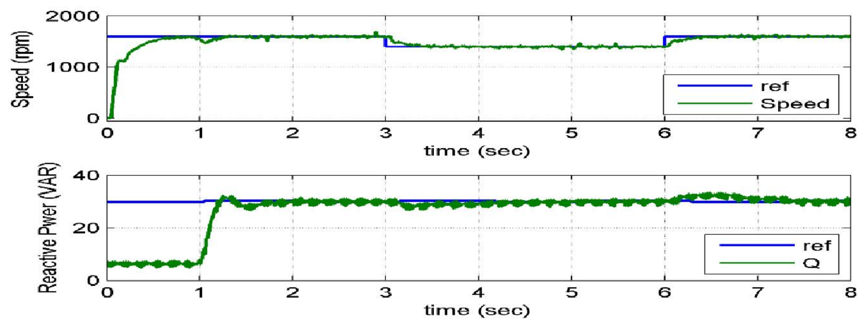


(b)

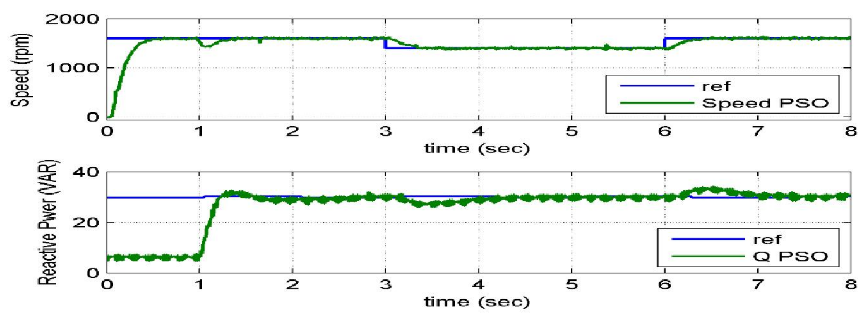


(c)

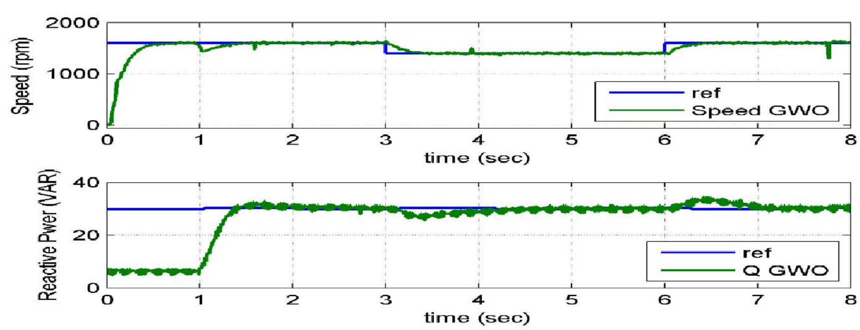
Figure 10. Experimental results of Speed and Reactive Power Case 2 at 1600 rpm. (a) PI controller.(b)PSO-PI controller. (c)GWO-PI controller.



(a)



(b)



(c)

Figure 11. Experimental results of Speed and Reactive Power Case 3 at 1600 rpm to 1400 rpm to 1600 rpm. (a) PI controller.(b)PSO-PI controller. (c)GWO-PI controller.

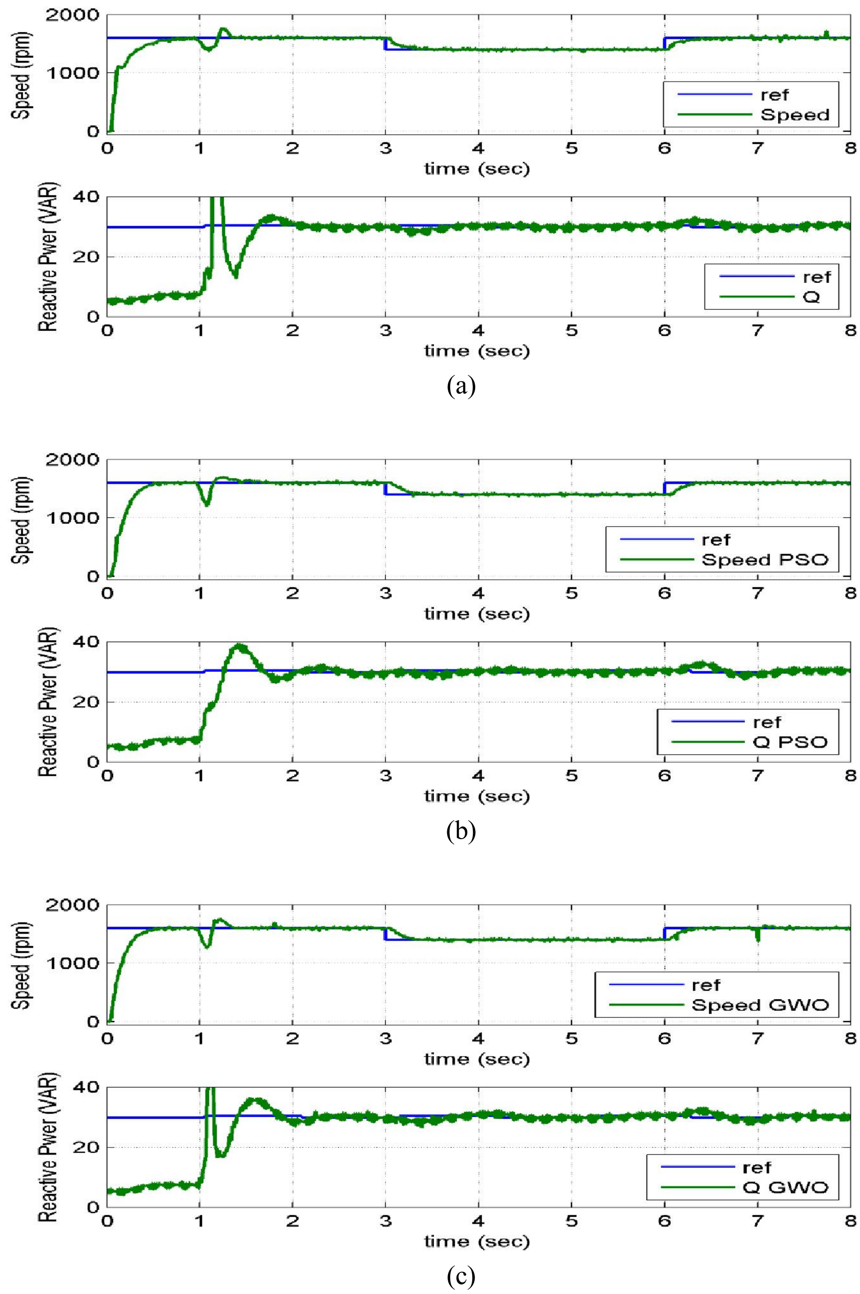


Figure 12. Experimental results of Speed and Reactive Power Case 4 at 1600 rpm to 1400 rpm to 1600 rpm. (a) PI controller.(b)PSO-PI controller. (c)GWO-PI controller.

From the above Figures, it is concluded that the GWO algorithm has been proven to be more efficient than the PSO algorithm in finding the global optimum PI parameters. Thus, the system performs better time response with the optimum PI controller. Also, the experimental results has a better time response under the PI controller tuned by the GWO.

6. Conclusions

In this paper, application of different stochastic optimization techniques such as PSO and GWO to tune the PI controllers of reactive power and speed control of a DFIG driven by Brushless DC Motor has been introduced. System identification technique was presentedin this work to estimate the transfer function of the reactive power loop and speed loop of the proposed system from the input-output test data. The comparison between the practical system and its identified model is performed and it conclude that a very small error which can be neglected.

Experimental results proved that the GWO is more efficient than PSO in seeking for the global optimum PI parameters with respect to the desired performance indices.Therefore the GWO algorithm offers a new optimization tool for tuning PI controller.

References

1. F.-J. Lin, J.-C. Hwang, K.-H. Tan, Z.-H. Lu, and Y.-R. Chang, "Control of doubly-fed induction generator system using PIDNNs," in *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on*, 2010, pp. 675-680.
2. L. Faa-Jeng, T. Kuang-Hsiung, L. Zong-Han, and C. Yung-Ruei, "Control of doubly-fed induction generator system using PFNN," in *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, 2011, pp. 2614-2621.
3. M. E.-S. El-Telbany, "Employing Particle Swarm Optimizer and Genetic Algorithms for Optimal Tuning of PID Controllers: A Comparative Study," *ICGST-ACSE Journal,* vol. Volume 7, November 2007.
4. S. El-Wakeel, F.Hassan, A.Kamel, and A. Abdel-hamed, "OPTIMUM TUNING OF PID CONTROLLER FOR A PERMANENT MAGNET BRUSHLESS DC MOTOR," *International journal of electrical engineering &Technology (IJEET),* vol. 4, pp. 53-64, 2013.
5. D. Nangru, D. K. Bairwa, K. Singh, S. Nema, and P. K. Padhy, " Modified PSO based PID Controller for Stable Processes," presented at the PDPM Indian Institute of Information Technology Design and Manufacturing, Jabalpur-482005 (M.P.), India,,IEEE, 2013. .
6. Z.-L. Gaing, "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System," IEEE Tran. Energy Conversion, Vol.19(2)Jun. 2004.
7. G. Huang, D. Wu, W. Yang, and Y. Xue, " Self-tuning of PID Parameters Based on the Modified Particle Swarm Optimization," presented at the Proceedings of the 8th World Congress on Intelligent Control and Automation, Jinan, China, IEEE,, 2010.
8. J.Kennedy and R.Eberhart, "Particle Swarm Optimization," presented at the IEEE International Conf. in Neural Networks, 1995.
9. S.-F. Chen, "Particle Swarm Optimization for PID Controllers with Robust Testing," presented at the International Conference on Machine Learning and Cybernetics, 19-22 Aug. 2007.
10. S. El-Wakeel, A. kamel, and A. Abdel-hamed, "A Hybrid Bacterial Foraging-Particle Swarm Optimization Technique for Optimal Tuning of Proportional-Integral-Derivative Controller of a Permanent Magnet Brushless DC Motor," *Electric Power Components and Systems,* vol. 43, pp. 309-309, 2-2015 2015.
11. M. R. Alrashidi; and M. E. El-Hawary, "A Survey of Particle Swarm Optimization Applications in Power System Operations," *Electric Power Components and Systems,* vol. 34, pp. 1349–1357, 2006.
12. k. Amged S. El-Wakeel, .A. Abdel-hamed, "“A Hybrid Bacterial Foraging-Particle Swarm Optimization Technique for Optimal Tuning of Proportional-Integral-Derivative Controller of a Permanent Magnet Brushless DC Motor”," *Electric Power Components and Systems,* vol. 43, pp. 309-309, 2-2015 2015.
13. L. Ljung, "“System Identification Toolbox 7.3: User’s Guide”," 2009.
14. S.El-Wakeel, "Design optimization of PM couplings using hybrid Particle Swarm Optimization-Simplex Method (PSO-SM) Algorithm," *Electric Power Systems Research,* vol. 116, pp. 29-35, Nov. 2014.
15. S. El-Wakeel, "Design optimization of PM couplings using hybrid Particle Swarm Optimization-Simplex Method (PSO-SM) Algorithm," *Electric Power Systems Research,* vol. 116, pp. 29-35, 2014.
16. M. Belsam, J. Ananth, and C. S. RaviChandran, " Soft Computing Based Design of PID Controller for a Linear Brushless DC Motor " *International Journal of Soft Computing,* vol. 7, pp. 58-62, 2012.
17. H. E. A. Ibrahim, F. N. Hassan, and A. O. Shomer, "Optimal PID control of a brushless DC motor using PSO and BF techniques " *Ain Shams Engineering Journal,* vol. 4, 2013.
18. Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer," in *IEEE int. Conf. Evolutionary Computation, Indiana Univ., Indianapolis*, 1998, pp. 69-73.
19. S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software,* vol. 69, pp. 46-61, 2014.
20. A. El-Fergany and H. M. Hasanien, "Single and Multi-objective Optimal Power Flow Using Grey Wolf Optimizer and Differential Evolution Algorithms," *Electric Power Components and Systems,* vol. 43, pp. 1548-1559, 2015.
21. Y. Sharma and L. C. Saikia, "Automatic generation control of a multi-area ST–Thermal power system using Grey Wolf Optimizer algorithm based classical controllers," *International Journal of Electrical Power & Energy Systems,* vol. 73, pp. 853-862, 2015.

11/10/2015